



Handbücher/Manuals



**VIPA**  
Gesellschaft für Visualisierung  
und Prozessautomatisierung mbH

Ohmstraße 4  
D-91074 Herzogenaurach  
Tel.: +49-9132-744-0  
Fax: +49-9132-744-144  
Internet: [www.vipa.de](http://www.vipa.de)  
E-Mail: [Info@vipa.de](mailto:Info@vipa.de)

# Handbuch

## VIPA System 200V

Best.-Nr.: VIPA HB97D

Rev. 03/51



Die Angaben in diesem Handbuch erfolgen ohne Gewähr. Änderungen des Inhalts können jederzeit ohne Vorankündigung erfolgen.

© Copyright 2003 VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH  
Ohmstraße 4, D-91074 Herzogenaurach,  
Tel.: +49 (91 32) 744 -0  
Fax.: +49 (91 32) 744-144  
EMail: info@vipa.de  
<http://www.vipa.de>

**Hotline: +49 (91 32) 744-114**

Alle Rechte vorbehalten

#### **Haftungsausschluss**

Der Inhalt dieses Handbuchs wurde auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft.

Dennoch können Abweichungen nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden regelmäßig überprüft und erforderliche Korrekturen sind in den nachfolgenden Auflagen enthalten.

Für Verbesserungsvorschläge sind wir dankbar.

#### **Warenzeichen**

VIPA, System 100V, System 200V, System 300V und System 500V sind eingetragene Warenzeichen der VIPA Gesellschaft für Visualisierung und Prozessautomatisierung mbH.

STEP und S7-300

sind eingetragenes Warenzeichen der Siemens AG.

Alle ansonsten im Text genannten Warenzeichen sind Warenzeichen der jeweiligen Inhaber und werden als geschützt anerkannt.

## Über dieses Handbuch

Das Handbuch beschreibt die bei VIPA erhältlichen System 200V Komponenten mit Ausnahme der CPUs. Hier finden Sie neben einer Produktübersicht eine detaillierte Beschreibungen der einzelnen Module. Sie erhalten Informationen für den Anschluss und die Handhabung der System 200V Komponenten. Am Ende eines Kapitels befinden sich immer die Technischen Daten der jeweiligen Module.

Für die CPUs sind gesonderte Handbücher erhältlich.

### Überblick

#### **Teil 1: Einleitung**

Im Rahmen dieser Einleitung erfolgt die Vorstellung des System 200V von VIPA als zentrales bzw. dezentrales Automatisierungssystem.

Des Weiteren finden Sie hier allgemeine Hinweise zum System 200V wie Maße, Montage und Betriebsbedingungen.

#### **Teil 2: Profibus-DP**

Inhalt dieses Kapitels ist der Einsatz des System 200V unter Profibus. Hier wird die Projektierung und Parametrierung der Profibus-Master- und -Slave-Module von VIPA beschrieben und auf verschiedene Kommunikationsbeispiele näher eingegangen.

#### **Teil 3: Interbus**

In diesem Kapitel befinden sich alle Informationen, die zur Anbindung einer System 200V Peripherie an Interbus erforderlich sind. Beschrieben sind Aufbau, Inbetriebnahme und Parametrierung des Interbus-Kopplers.

#### **Teil 4: CAN-Bus CANopen**

Dieser Teil befasst sich mit den CANopen-Slaves von VIPA und deren Einsatz im CAN-Bus. Anhand von Beispielen werden Programmaufbau und Parametrierung des CAN-Slaves gezeigt.

#### **Teil 5: DeviceNet**

In diesem Kapitel befindet sich die Beschreibung des DeviceNet-Kopplers von VIPA. Nach Vorstellung und Beschreibung des Moduls wird anhand von Beispielen die Projektierung des DeviceNet-Kopplers und die Parametrierung der System 200V Module im *DeviceNet-Manager* der Firma Allen - Bradley beschrieben. Am Ende des Kapitels finden Sie eine Übersicht der Diagnosemeldungen und die Anbindungsmöglichkeiten an Profibus.

#### **Teil 6: SERCOS**

Inhalt dieses Kapitels ist die Beschreibung des SERCOS-Kopplers von VIPA. Ein Bestandteil ist die Projektierung, Parametrierung und die Adressierung. Über ein Beispiel wird die ID-Zuweisung erklärt.

**Teil 7: Ethernet-Koppler**

Inhalt dieses Kapitels ist die Beschreibung des Ethernet-Kopplers IM 253NET von VIPA. Sie bekommen hier alle Informationen, die für Aufbau und Inbetriebnahme des Ethernet-Kopplers erforderlich sind.

**Teil 8: PC 288 - CPU**

Im Rahmen dieses Kapitels wird die PC-CPU PC 288 und deren Einsatz im System 200V beschrieben. Es wird ausführlich auf die Projektierung eines PC-basierenden Systems eingegangen.

Eine Übersicht des BIOS-Setup und der Register runden das Kapitel ab.

**Teil 9: Kommunikationsprozessor CP 240**

In diesem Kapitel finden Sie Informationen über den Aufbau, die Anschlussbelegung sowie die Übertragungsprotokolle der Kommunikationsprozessoren CP 240. Außerdem sind die mitgelieferten Standardhantierungsbausteine für die CPU21x und CPU24x von VIPA erklärt.

**Teil 10: Zähler-Module**

Dieser Teil befasst sich mit den Zähler-Modulen von VIPA. Hier finden Sie auch Informationen zum SSI-Modul. Aufbau, Parametrierung und die verschiedenen Zählermodi mit deren Anschaltungen sind Bestandteil dieses Kapitels.

**Teil 11: MotionControl-Module FM 253 und FM 254**

Das Kapitel beschreibt das MotionControl-Stepper- und das MotionControl Servo-Modul von VIPA. Sie erhalten Informationen zu Aufbau, Betriebsarten, Datenübergabe und zum Einsatz in Verbindung mit einem Schritt- oder Servomotor.

**Teil 12: Spannungsversorgungen**

Dieser Teil befasst sich mit den externen Stromversorgungen für das System 200V. Neben Aufbau, Montage und Inbetriebnahme finden Sie hier alle Sicherheitshinweise, die beim Einsatz zu beachten sind.

**Teil 13-15: Digitale Ein-/Ausgabe-Module**

In diesen Kapiteln werden die digitalen Peripherie-Module von VIPA beschrieben. Sie erhalten hier alle Informationen, die für den Einsatz dieser Module erforderlich sind. Sie finden in Kapitel 13 alle Eingabe-Module, in Kapitel 14 alle Ausgabe-Module und in Kapitel 15 alle Ein-/Ausgabe-Module.

**Teil 16-18: Analoge Ein-/Ausgabe-Module**

Inhalt dieses Kapitels ist die Beschreibung der analogen Peripherie-Module. Auch hier erhalten Sie für jedes Modul alle Informationen, die für den Einsatz erforderlich sind. Teil 16 beschreibt alle Eingabe-, Teil 17 alle Ausgabe- und Teil 18 alle analogen Ein-/Ausgabe-Module von VIPA.

**Teil 19: Systemerweiterungen**

Dieses Kapitel befasst sich mit den Systemerweiterungs-Modulen, die für das System 200V erhältlich sind. Dies sind z.B. die Buserweiterungen IM26x zur Erweiterung von einer Buszeile auf mehrere, der 4-fach Mini-Switch CM 240 und die Klemmen-Module zur Erweiterung der Anschlussmöglichkeiten.

**Teil 20: Montage und Aufbaurichtlinien**

Alle Informationen, die für den Aufbau und die Verdrahtung einer Steuerung aus den Komponenten des Systems 200V erforderlich sind, finden Sie in diesem Kapitel.

# Inhaltsverzeichnis

<b>Benutzerhinweise .....</b>	<b>1</b>
<b>Sicherheitshinweise .....</b>	<b>2</b>
<b>Teil 1 Einleitung .....</b>	<b>1-1</b>
Sicherheitshinweise für den Benutzer .....	1-2
Hinweise zum Green Cable von VIPA.....	1-3
Übersicht .....	1-4
Komponenten .....	1-5
Übersicht der GSD-Dateien von VIPA.....	1-6
Allgemeine Beschreibung System 200V .....	1-7
ISO/OSI-Schichtenmodell.....	1-8
Kommunikationsebenen in der Automatisierung.....	1-11
<b>Teil 2 Profibus-DP .....</b>	<b>2-1</b>
Systemübersicht .....	2-2
Grundlagen.....	2-5
IM 208DP - Master mit RS485 - Aufbau.....	2-10
IM 208DP - Master mit RS485 - Einsatz an einer CPU 21x.....	2-13
IM 208DP - Master mit RS485 - Projektierung.....	2-14
IM 208DP - Master mit RS485 - Urlöschen .....	2-22
IM 208DPO - Master mit LWL - Aufbau .....	2-23
IM 208DPO - Master mit LWL - Einsatz an einer CPU 21x .....	2-26
IM 208DPO - Master mit LWL - Projektierung.....	2-27
IM 253DP - Slave (Standard) - Aufbau.....	2-33
IM 253 DPR - Slave (redundant) - Aufbau .....	2-36
IM 253DP, DO 24xDC 24V - Aufbau.....	2-39
IM 253DP - Slave - Blockschaltbild .....	2-43
IM 253DP - Slave - Projektierung.....	2-44
IM 253DP - Slave - Parameter .....	2-46
IM 253DP - Slave - Diagnosefunktionen .....	2-47
Aufbau Richtlinien .....	2-54
Inbetriebnahme.....	2-64
Einsatz der Diagnose-LEDs.....	2-65
Beispiele zur Profibus-Kommunikation.....	2-66
Technische Daten.....	2-74
<b>Teil 3 Interbus .....</b>	<b>3-1</b>
Systemübersicht .....	3-2
Grundlagen.....	3-3
IM 253IBS - Interbus-Koppler - Aufbau .....	3-7
Anschluss an Interbus .....	3-10
Einsatz im Interbus .....	3-11
Inbetriebnahme.....	3-15
Technische Daten.....	3-18

<b>Teil 4 CANopen.....</b>	<b>4-1</b>
Systemübersicht .....	4-2
Grundlagen.....	4-3
IM 253CAN - CANopen Slave - Aufbau .....	4-5
IM 253CAN, DO 24xDC 24V - Aufbau .....	4-9
CANopen Schnelleinstieg .....	4-13
Einstellung von Baudrate und Modul-ID .....	4-17
Telegrammaufbau .....	4-18
PDO - Prozessdaten-Objekt .....	4-20
SDO - Servicedaten-Objekt .....	4-24
Objekt-Verzeichnis.....	4-26
Emergency Object .....	4-67
NMT - Netzwerk Management .....	4-68
Technische Daten.....	4-70
<b>Teil 5 DeviceNet .....</b>	<b>5-1</b>
Systemübersicht .....	5-2
Grundlagen.....	5-3
IM 253DN - DeviceNet-Koppler - Aufbau .....	5-5
Projektierung unter Einsatz des DeviceNet-Managers .....	5-8
Einstellung von Baudrate und DeviceNet-Adresse.....	5-9
Test am DeviceNet-Bus.....	5-10
Module im DeviceNet-Manager parametrieren.....	5-11
I/O-Adressierung des DeviceNet-Scanners .....	5-16
Diagnose .....	5-17
Schnittstelle zum Profibus.....	5-22
Technische Daten.....	5-23
<b>Teil 6 SERCOS.....</b>	<b>6-1</b>
Systemübersicht .....	6-2
Grundlagen.....	6-3
IM 253Sercos - SERCOS-Koppler - Aufbau.....	6-5
Grundparametrierung über Adresseinsteller .....	6-8
SERCOS Identifier .....	6-10
Beispiel zur automatischen ID-Zuweisung .....	6-13
Technische Daten.....	6-22
<b>Teil 7 Ethernet-Koppler.....</b>	<b>7-1</b>
Systemübersicht .....	7-2
Grundlagen Ethernet .....	7-3
Planung eines Netzwerks .....	7-7
IM 253NET - Ethernet-Koppler - Aufbau .....	7-9
Zugriffsmöglichkeiten auf den Ethernet-Koppler .....	7-11
Prinzip der automatischen Adressierung.....	7-14
Projektierung unter WinNCS.....	7-15
Diagnose und Test mittels Internet Browser .....	7-16
ModbusTCP.....	7-20
Modbus-Funktionscodes.....	7-21



Siemens S5 Header Protokoll .....	7-25
Programmierbeispiel .....	7-27
Technische Daten .....	7-28
<b>Teil 8 PC 288 - CPU .....</b>	<b>8-1</b>
Systemübersicht .....	8-2
Grundlagen .....	8-3
Eigenschaften .....	8-4
PC 288 - CPU - Aufbau .....	8-4
Komponenten .....	8-5
Einsatz der Speichermedien .....	8-9
Einsatz im System 200V .....	8-10
BIOS-Setup Bedienung .....	8-13
Registerbeschreibung .....	8-21
Technische Daten .....	8-23
<b>Teil 9 Kommunikationsprozessor CP 240 .....</b>	<b>9-1</b>
Systemübersicht .....	9-2
Grundlagen ASCII, STX/ETX, 3964(R), RK512 .....	9-3
Grundlagen Modbus .....	9-10
CP 240 mit 20mA/RS232C-Schnittstelle - Aufbau .....	9-11
CP 240 mit RS422/RS485-Schnittstelle - Aufbau .....	9-16
Parametrierung .....	9-22
Zugriff auf das CP 240 Interface unter ASCII, STX/ETX, 3964(R) .....	9-30
Einsatz unter Modbus .....	9-32
Modbus-Funktionscodes .....	9-36
Beispiel zum Einsatz unter Modbus .....	9-39
Kommunikation über Standardhantierungsbausteine .....	9-45
Standardhantierungsbausteine für CPU 24x .....	9-46
Standardhantierungsbausteine für CPU 21x .....	9-61
Technische Daten .....	9-78
<b>Teil 10 Zähler-Module .....</b>	<b>10-1</b>
Systemübersicht .....	10-2
FM 250S - SSI-Interface - Aufbau .....	10-3
FM 250 - Zähler-Modul - Aufbau .....	10-9
Übersicht Zählermodi und Anschaltung .....	10-12
Zählermodi .....	10-14
Technische Daten .....	10-58
<b>Teil 11 MotionControl-Module .....</b>	<b>11-1</b>
Systemübersicht .....	11-2
FM 253 - MotionControl Stepper .....	11-3
FM 253 - MotionControl Stepper - Aufbau .....	11-4
FM 253 - Anschluss eines Antriebs .....	11-6
FM 253 - Datenübergabe >> FM 253 .....	11-8
FM 253 - Parametrierung .....	11-9
FM 253 - Betriebsarten .....	11-11
FM 253 - Datenübergabe > CPU .....	11-15

FM 253 - Hantierungsbausteine .....	11-17
FM 254 - MotionControl Servo .....	11-23
FM 254 - MotionControl Servo - Aufbau.....	11-24
FM 254 - Anschluss eines Antriebs mit Drehgeber .....	11-26
FM 254 - Übersicht der Parameter und Übergabewerte.....	11-28
FM 254 - Parametrierung.....	11-29
FM 254 - Datenübergabe >> FM 254.....	11-30
FM 254 - Betriebsarten .....	11-31
FM 254 - Datenübergabe >> CPU .....	11-37
Technische Daten.....	11-38
<b>Teil 12 Spannungsversorgungen.....</b>	<b>12-1</b>
Sicherheitshinweise .....	12-2
Systemübersicht .....	12-3
PS 207/2 - Spannungsversorgung - Aufbau.....	12-4
PS 207/2CM - Spannungsversorgung mit Klemmen - Aufbau .....	12-6
Montage.....	12-8
Verdrahtung.....	12-9
Technische Daten.....	12-10
<b>Teil 13 Digitale Eingabe-Module .....</b>	<b>13-1</b>
Systemübersicht .....	13-2
DI 8xDC 24V.....	13-4
DI 8xDC 24V 0,2ms .....	13-6
DIa 8xDC 24V.....	13-8
DI 8xDC 24V NPN .....	13-10
DI 4xAC/DC 90...230V .....	13-12
DI 8xAC/DC 60...230V .....	13-14
DI 8xAC/DC 24...48V .....	13-16
DI 8xAC 240V .....	13-18
DI 8xAC/DC 180...265V .....	13-20
DI 16xDC 24V mit UB4x .....	13-22
DI 16xDC 24V.....	13-24
DI 16xDC 24V/1C .....	13-26
DI 16xDC 24V NPN .....	13-36
DI 32xDC 24V.....	13-38
<b>Teil 14 Digitale Ausgabe-Module .....</b>	<b>14-1</b>
Systemübersicht .....	14-2
DO 8xDC 24V 1A.....	14-4
DO 8xDC 24V 2A.....	14-6
DO 8xDC 24V 2A potenzialgetrennt 4 á 2.....	14-8
DO 16xDC 24V 0,5A mit UB4x .....	14-10
DO 16xDC 24V 1A.....	14-12
DO 16xDC 24V 2A.....	14-14
DO 16xDC 24V 0,5A NPN .....	14-16
DO 32xDC 24V 1A.....	14-18
DO 8xRelais COM .....	14-20

DO 4xRelais.....	14-22
DO 4xRelais bistabil.....	14-24
DO 8xSolid State COM.....	14-26
DO 4xSolid State .....	14-28
<b>Teil 15 Digitale Ein-/Ausgabe-Module.....</b>	<b>15-1</b>
Systemübersicht .....	15-2
Sicherheitshinweise zum Einsatz der DIO-Module.....	15-2
DIO 8xDC 24V 1A.....	15-3
DI 16xDC 24V, DO 16xDC 24V 1A .....	15-5
<b>Teil 16 Analoge Eingabe-Module .....</b>	<b>16-1</b>
Systemübersicht .....	16-2
Allgemeines .....	16-4
AI 4x16Bit, Multi-Input.....	16-5
AI 4x12Bit, 4 ... 20mA, potenzialgetrennt.....	16-16
AI 4x12Bit, $\pm 10V$ , potenzialgetrennt.....	16-19
AI 4x16Bit f .....	16-22
AI 8x16Bit .....	16-32
<b>Teil 17 Analoge Ausgabe-Module .....</b>	<b>17-1</b>
Systemübersicht .....	17-2
Allgemeines .....	17-3
AO 4x12Bit, Multi-Output .....	17-4
AO 4x12Bit f, Multi-Output .....	17-12
<b>Teil 18 Analoge Ein-/Ausgabe-Module.....</b>	<b>18-1</b>
Systemübersicht .....	18-2
Sicherheitshinweis zur Bereichseinstellung.....	18-2
Allgemeines .....	18-3
AI 2/AO 2x12Bit, Multi-In-/Output.....	18-4
<b>Teil 19 Systemerweiterungen.....</b>	<b>19-1</b>
Systemübersicht .....	19-2
Buserweiterung IM 260, IM 261 .....	19-4
4fach Mini-Switch CM 240 .....	19-7
Klemmen-Modul CM 201 .....	19-10
<b>Teil 20 Montage und Aufbaurichtlinien .....</b>	<b>20-1</b>
Übersicht .....	20-2
Montage.....	20-4
Verdrahtung.....	20-9
Einbaumaße .....	20-11
Maschinelle Beschriftung .....	20-12
Aufbaurichtlinien .....	20-13
<b>Anhang .....</b>	<b>A-1</b>
Index.....	A-1



## Benutzerhinweise

**Zielsetzung und Inhalt** Dieses Handbuch beschreibt die Module, die im System 200V eingesetzt werden können. Beschrieben werden Aufbau, Projektierung und Technische Daten.

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

**Aufbau des Handbuchs** Das Handbuch ist zur Zeit gegliedert in 20 Kapitel. 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
- Stichwortverzeichnis (Index) am Ende des Handbuchs

**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 Tips

## Sicherheitshinweise

### Bestimmungsgemäße Verwendung

Das System 200V 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:

- Änderung am Automatisierungssystem nur im spannungslosen Zustand vornehmen!
- Anschluss und 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 Einleitung

### Überblick

Kernthema dieses Kapitels ist die Vorstellung des System 200V von VIPA. In einer Übersicht werden die Möglichkeiten zum Aufbau von zentralen und dezentralen Systemen aufgezeigt.

Auch finden Sie hier allgemeine Angaben zum System 200V wie Maße, Hinweise zur Montage und zu den Umgebungsbedingungen.

Mit dem 7-Schichten-Modell und einer Auflistung der Kommunikationsebenen in der Automatisierungstechnik endet das Kapitel.

Nachfolgend sind beschrieben:

- Vorstellung des System 200V
- Allgemeine Beschreibung wie Maße, Montage, Betriebssicherheit und Umgebungsbedingungen
- 7-Schichten-Modell und Kommunikationsebenen

### Inhalt

Thema	Seite
<b>Teil 1 Einleitung</b> .....	<b>1-1</b>
Sicherheitshinweise für den Benutzer .....	1-2
Hinweise zum Green Cable von VIPA.....	1-3
Übersicht .....	1-4
Komponenten .....	1-5
Übersicht der GSD-Dateien von VIPA.....	1-6
Allgemeine Beschreibung System 200V .....	1-7
ISO/OSI-Schichtenmodell .....	1-8
Kommunikationsebenen in der Automatisierung .....	1-11

## Sicherheitshinweise für den Benutzer

### Handhabung elektrostatisch gefährdeter Baugruppen

VIPA-Module und 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 Komponenten wird nachfolgendes Symbol verwendet:



Das Symbol befindet sich auf Modulen, Baugruppen, Baugruppenträgern oder auf Verpackungen und weist so auf elektrostatisch gefährdete Komponenten 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 diese 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 Baugruppen

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

- Potenzialfreie 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.



## Hinweise zum Green Cable von VIPA

### Was ist das Green Cable



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

Das Green Cable ist ein Programmier- und Downloadkabel für VIPA CPUs 11x, 21x, 31x, 51x sowie VIPA Feldbus-Master. Sie erhalten das Green Cable von VIPA unter der Best.-Nr.: VIPA 950-0KB00.

Mit dem Green Cable können Sie:

- *Projekte seriell übertragen*  
Unter Umgehung aufwändiger Hardware (MPI-Adapter, etc.) können Sie über das Green Cable eine serielle Punkt-zu-Punkt-Verbindung über die MP<sup>2</sup>I-Schnittstelle realisieren. Somit können Sie Komponenten an Ihre VIPA-CPU anschließen, die seriell über einen MPI-Adapter kommunizieren können wie beispielsweise ein Visualisierungssystem.
- *Firmware-Updates der CPUs und Feldbus-Master durchführen*  
Über das Green Cable können Sie unter Einsatz eines Upload-Programms die Firmware aller aktuellen CPUs 11x, 21x, 31x, 51x sowie bestimmte Feldbus-Master (s. Hinweis) aktualisieren.



### 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).

Zurzeit unterstützen folgende Komponenten das Green Cable:

CPUs 11x, 21x, 31x, 51x sowie die Feldbus-Master 208-1xx01 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!

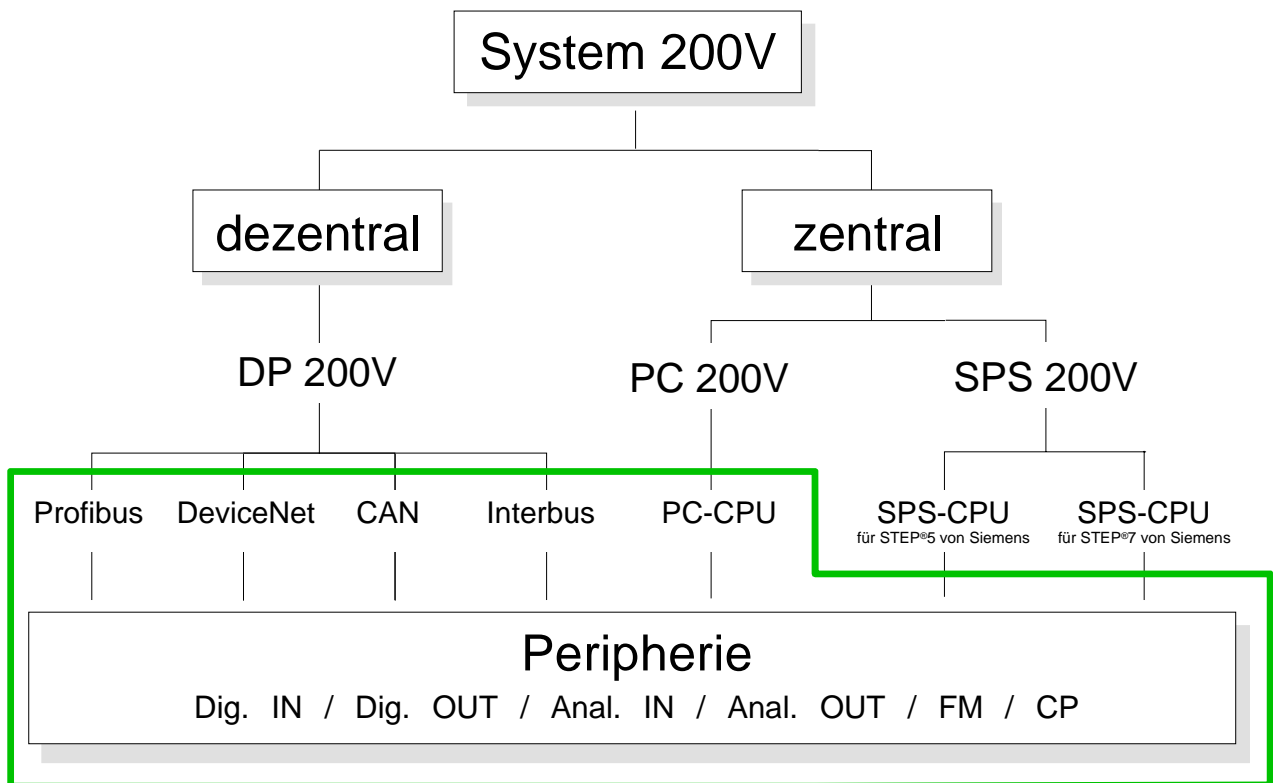
Das Green Cable darf nur mit einem 1:1 Kabel (alle 9 Pin 1:1 verbunden) verlängert werden.

# Übersicht

## Das System 200V

Das System 200V ist ein modulares, zentral wie dezentral einsetzbares Automatisierungssystem für Anwendungen im unteren und mittleren Leistungsbereich. Die einzelnen Module werden direkt auf eine 35mm-Normprofilschiene montiert und über Busverbinder, die vorher in die Profilschiene eingelegt werden, gekoppelt.

Die nachfolgende Abbildung soll Ihnen den Leistungsumfang des System 200V verdeutlichen:



## Komponenten

- Zentrales System** Im System 200V stehen verschiedene SPS-CPU's zur Verfügung. Programmiert wird in STEP<sup>®</sup>5 oder STEP<sup>®</sup>7 von Siemens.
- CPU's mit integrierter Ethernetanschaltung oder mit zusätzlichen seriellen Schnittstellen garantieren eine komfortable Integration der SPS in ein Netzwerk oder den Anschluss von zusätzlichen Endgeräten.
- Das Anwenderprogramm wird im Flash oder einem zusätzlich steckbaren Speichermodul gespeichert.
- Bedienen/Beobachten, Steuerungsaufgaben oder andere Dateiverarbeitungsaufgaben können mit der PC-basierenden CPU 288 realisiert werden.
- Programmiert wird in C++ oder Pascal.
- Die PC 288-CPU ermöglicht einen aktiven Zugriff auf den Rückwandbus und ist so mit allen Peripherie- und Funktionsmodulen des VIPA System 200V als zentrale Steuerung einsetzbar.
- Mit einer Zeilenanschaltung ist ein Aufbau des System 200V in bis zu 4 Zeilen möglich.
- Dezentrales System** Die SPS-CPU's oder die PC-CPU bilden in Kombination mit einem Profibus DP-Master die Basis für ein Profibus-DP-Netzwerk nach DIN 19245-3. Das DP-Netzwerk kann mit den gängigen Konfigurationstools parametrierbar werden.
- Eine direkte Parametrierung über das Profibus-Netzwerk ist mit der VIPA-Software WinNCS unter Verwendung einer Profibus-Master-PC-Steckkarte der Firma Softing möglich. Optional stehen Profibus-Module auch mit Kunststoff-LWL-Anschluss zur Verfügung.
- Die Anbindung an weitere Feldbussysteme ermöglichen Slaves für Interbus, CANopen und DeviceNet.
- Peripheriemodule** Von VIPA erhalten Sie eine Vielzahl an Peripheriemodulen wie z.B. für digitale bzw. analoge Ein-/Ausgabe, Zählerfunktionen, Wegmessung, Positionierung und serielle Kommunikation.
- Die Peripheriemodule können zentral und dezentral betrieben werden.

## Übersicht der GSD-Dateien von VIPA

### Allgemeines

Die Funktionalität aller Systemkomponenten von VIPA sind in Form von verschiedenen GSD-Dateien verfügbar.

Da die Profibus-Schnittstelle auch softwareseitig standardisiert ist, können wir auf diesem Weg gewährleisten, dass über die Einbindung einer GSD-Datei die Funktionalität in Verbindung mit dem STEP<sup>®</sup>7 Manager von Siemens jederzeit gegeben ist.

Für jede Systemfamilie erhalten Sie eine GSD-Datei. Eine Systemzuordnung und den jeweiligen symbolischen Namen im Hardwarekatalog finden Sie in der nachfolgenden Tabelle:

GSD		System			Kommentar
Dateiname	Symbolname	100V	200V	300V	
DP2V0550.GSD	VIPA_DP200V	✓	✓		Alte GSD-Datei wird nicht weiter gepflegt
<b>System 100V</b>					
VIPA04D4.GSD	VIPA_DP100V	✓			Zur Einbindung dezentrale Peripherie 15x in DP-Master-System
VIPA_11x.GSD	VIPA_CPU11x	✓			Zur Projektierung der direkten E/A-Peripherie in der CPU als virtuelles Profibus-System
VIPA04Dx.GSD	VIPA_CPU11xDP	✓			Zur Einbindung CPU 11x als intelligenter Slave in DP-Master-System
<b>System 200V</b>					
VIPA0550.GSD	VIPA_DP200V_2		✓		Zur Einbindung IM 253 Slave in DP-Master-System
VIPA_21x.GSD	VIPA_CPU21x		✓		Zur Projektierung der direkten E/A-Peripherie in der CPU als virtuelles Profibus-System
VIPA04D5.GSD	VIPA_CPU2xxDP		✓		Zur Einbindung CPU 21x als intelligenter Slave in DP-Master-System
VIPA2ETH.GSD	VIPA_ETH200V		✓		Zur Projektierung IM 253NET unter WinNCS
<b>System 300V</b>					
VIPA056B.GSD	VIPA_DP300V			✓	Zur Einbindung IM 353 Slave in DP-Master-System
VIPA802F.GSD	VIPA_CPU31xDP			✓	Zur Einbindung CPU 31x als intelligenter Slave in DP-Master

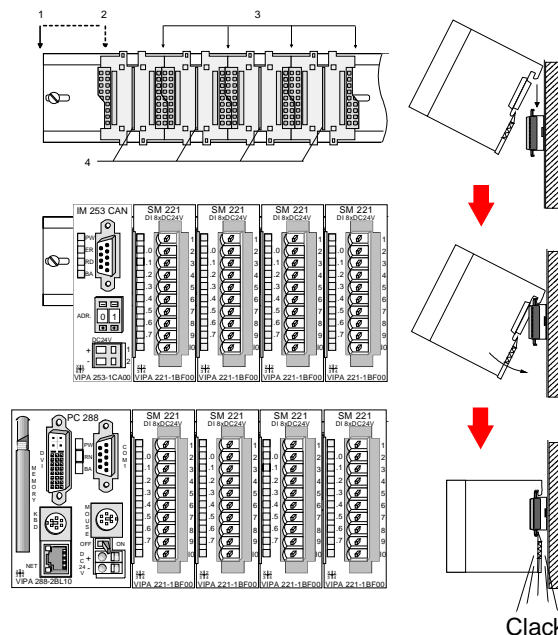
## Allgemeine Beschreibung System 200V

### Aufbau/Maße

- Normprofil-Hutschiene 35mm
- Peripherie-Module mit seitlich versenkbaren Beschriftungsstreifen
- Maße Grundgehäuse:
  - 1fach breit: (HxBxT) in mm: 76x25,4x76 in Zoll: 3x1x3
  - 2fach breit: (HxBxT) in mm: 76x50,8x76 in Zoll: 3x2x3

### Montage

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



- [1] Kopfmodul wie PC, CPU, Buskoppler wenn doppelt breit
- [2] Kopfmodul, wenn einfach breit
- [3] Peripheriemodule
- [4] Führungsleisten

### Hinweis

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

Im Kapitel "Montage und Aufbau Richtlinien" finden Sie eine Übersicht der Stromaufnahmen.

### 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)

### Umgebungsbedingungen

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

## ISO/OSI-Schichtenmodell

### Übersicht

Das ISO/OSI-Schichtenmodell basiert auf einem Vorschlag, der von der International Standards Organization (ISO) entwickelt wurde. Es stellt den ersten Schritt zur internationalen Standardisierung der verschiedenen Protokolle dar. Das Modell trägt den Namen ISO-OSI-Schichtenmodell. OSI steht für **O**pen **S**ystem **I**nterconnection, die Kommunikation offener Systeme. Das ISO/OSI-Schichtenmodell ist keine Netzwerkarchitektur, da die genauen Dienste und Protokolle, die in jeder Schicht verwendet werden, nicht festgelegt sind. Sie finden in diesem Modell lediglich Informationen über die Aufgaben, die die jeweilige Schicht zu erfüllen hat.

Jedes offene Kommunikationssystem basiert heutzutage auf dem durch die Norm ISO 7498 beschriebenen ISO/OSI Referenzmodell. Das Referenzmodell strukturiert Kommunikationssysteme in insgesamt 7 Schichten, denen jeweils Teilaufgaben in der Kommunikation zugeordnet sind. Dadurch wird die Komplexität der Kommunikation auf verschiedene Ebenen verteilt und somit eine größere Übersichtlichkeit erreicht.

Folgende Schichten sind definiert:

Schicht	Funktion
Schicht 7	Application Layer (Anwendung)
Schicht 6	Presentation Layer (Darstellung)
Schicht 5	Session Layer (Sitzung)
Schicht 4	Transport Layer (Transport)
Schicht 3	Network Layer (Netzwerk)
Schicht 2	Data Link Layer (Sicherung)
Schicht 1	Physical Layer (Bitübertragung)

Je nach Komplexität der geforderten Übertragungsmechanismen kann sich ein Kommunikationssystem auf bestimmte Teilschichten beschränken.

Im Falle von INTERBUS und Profibus sind beispielsweise nur die Schichten 1 und 2 implementiert.

Auf der Folgeseite finden Sie eine nähere Beschreibung der Schichten.

**Schichten****Schicht 1** Bitübertragungsschicht (physical layer)

Die Bitübertragungsschicht beschäftigt sich mit der Übertragung von Bits über einen Kommunikationskanal. Allgemein befasst sich diese Schicht mit den mechanischen, elektrischen und prozeduralen Schnittstellen und mit dem physikalischen Übertragungsmedium, das sich unterhalb der Bitübertragungsschicht befindet:

- Wieviel Volt entsprechen einer logischen 0 bzw. 1.
- Wie lange muss die Spannung für ein Bit anliegen.
- Pinbelegung der verwendeten Schnittstelle.

**Schicht 2** Sicherungsschicht (data link layer)

Diese Schicht hat die Aufgabe, die Übertragung von Bitstrings zwischen zwei Teilnehmern sicherzustellen. Dazu gehören die Erkennung und Behebung bzw. Weitermeldung von Übertragungsfehlern, sowie die Flusskontrolle.

Die Sicherungsschicht verwandelt die zu übertragenden Rohdaten in eine Datenreihe. Hier werden Rahmengrenzen beim Sender eingefügt und beim Empfänger erkannt. Dies wird dadurch erreicht, dass am Anfang und am Ende eines Rahmens spezielle Bitmuster gesetzt werden. In der Sicherungsschicht wird häufig noch eine Flussregelung und eine Fehlererkennung integriert.

Die Datensicherungsschicht ist in zwei Unterschichten geteilt, die LLC- und die MAC-Schicht.

Die MAC (**Media Access Control**) ist die untere Schicht und steuert die Art, wie Sender einen einzigen Übertragungskanal gemeinsam nutzen

Die LLC (**Logical Link Control**) ist die obere Schicht und stellt die Verbindung für die Übertragung der Datenrahmen von einem Gerät zum anderen her.

**Schicht 3** Netzwerkschicht (network layer)

Die Netzwerkschicht wird auch Vermittlungsschicht genannt.

Die Aufgabe dieser Schicht besteht darin, den Austausch von Binärdaten zwischen nicht direkt miteinander verbundenen Stationen zu steuern. Sie ist für den Ablauf der logischen Verknüpfungen von Schicht 2-Verbindungen zuständig. Dabei unterstützt diese Schicht die Identifizierung der einzelnen Netzwerkadressen und den Auf- bzw. Abbau von logischen Verbindungskanälen.

Eine weitere Aufgabe der Schicht 3 besteht in der priorisierten Übertragung von Daten und die Fehlerbehandlung von Datenpaketen.

**Schicht 4** Transportschicht (transport layer)

Die Aufgabe der Transportschicht besteht darin, Netzwerkstrukturen mit den Strukturen der höheren Schichten zu verbinden, indem sie Nachrichten der höheren Schichten in Segmente unterteilt und an die Netzwerkschicht weiterleitet. Hierbei wandelt die Transportschicht die Transportadressen in Netzwerkadressen um.

Gebräuchliche Transportprotokolle sind: TCP, SPX, NWLink und NetBEUI.

**Schichten**  
**Fortsetzung ...****Schicht 5** Sitzungsschicht (session layer)

Die Sitzungsschicht wird auch Kommunikationssteuerungsschicht genannt. Sie erleichtert die Kommunikation zwischen Service-Anbieter und -Requestor durch Aufbau und Erhaltung der Verbindung, wenn das Transportsystem kurzzeitig ausgefallen ist.

Auf dieser Ebene können logische Benutzer über mehrere Verbindungen gleichzeitig kommunizieren. Fällt das Transportsystem aus, so ist es die Aufgabe, gegebenenfalls eine neue Verbindung aufzubauen.

Darüber hinaus werden in dieser Schicht Methoden zur Steuerung und Synchronisation bereitgestellt.

**Schicht 6** Darstellungsschicht (presentation layer)

Auf dieser Ebene werden die Darstellungsformen der Nachrichten behandelt, da bei verschiedenen Netzsystemen unterschiedliche Darstellungsformen benutzt werden.

Die Aufgabe dieser Schicht besteht in der Konvertierung von Daten in ein beiderseitig akzeptiertes Format, damit diese auf den verschiedenen Systemen lesbar sind.

Hier werden auch Kompressions-/Dekompressions- und Verschlüsselungs-/Entschlüsselungsverfahren durchgeführt.

Man bezeichnet diese Schicht auch als Dolmetscherdienst. Eine typische Anwendung dieser Schicht ist die Terminalemulation.

**Schicht 7** Anwendungsschicht (application layer)

Die Anwendungsschicht stellt sich als Bindeglied zwischen der eigentlichen Benutzeranwendung und dem Netzwerk dar. Sowohl die Netzwerk-Services wie Datei-, Druck, Nachrichten-, Datenbank- und Anwendungs-Service als auch die zugehörigen Regeln gehören in den Aufgabenbereich dieser Schicht.

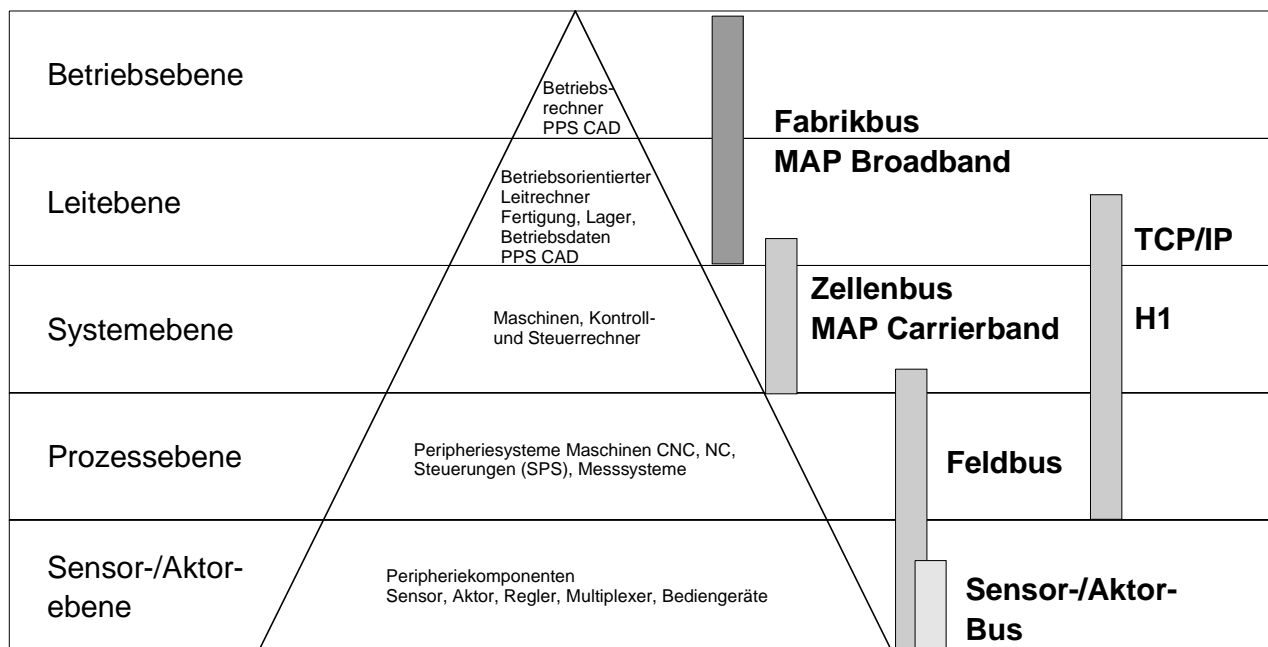
Diese Schicht setzt sich aus einer Reihe von Protokollen zusammen, die entsprechend den wachsenden Anforderungen der Benutzer ständig erweitert werden.



## Kommunikationsebenen in der Automatisierung

Der Informationsfluss in einem Unternehmen stellt sehr unterschiedliche Anforderungen an die eingesetzten Kommunikationssysteme. Je nach Unternehmensbereich hat ein Bussystem unterschiedlich viele Teilnehmer, es sind unterschiedlich große Datenmengen zu übertragen, die Übertragungsintervalle variieren, usw.

Aus diesem Grund greift man je nach Aufgabenstellung auf unterschiedliche Bussysteme zurück, die sich wiederum in verschiedene Klassen einteilen lassen. Eine Zuordnung verschiedener Bussysteme zu den Hierarchieebenen eines Unternehmens zeigt das folgende Modell:



Während innerhalb der Betriebsebene häufig sehr große, zeitunkritische Datenmengen übertragen werden, ist vor allem in der untersten Ebene, dem Sensor- / Aktor-Bereich, eine effiziente Übertragung von sehr kleinen Datenmengen gefordert. Zudem muss ein Bussystem auf Sensor- / Aktor-ebene häufig Echtzeitanforderungen erfüllen.



## Teil 2 Profibus-DP

### Überblick

Inhalt dieses Kapitels ist der Einsatz des System 200V unter Profibus-DP. Nach einer kurzen Einführung und Systemvorstellung wird die Projektierung und Parametrierung der Profibus-Master- und -Slave-Module von VIPA gezeigt. Verschiedene Kommunikationsbeispiele und die technischen Daten runden das Kapitel ab.

Nachfolgend sind beschrieben:

- Systemübersicht der Profibus-Module von VIPA
- Grundlagen zum Profibus-DP
- Aufbau und Projektierung der Profibus-Master IM 208DP
- Aufbau und Projektierung der Profibus-Slaves IM 253DP
- Projektierbeispiele
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 2 Profibus-DP</b> .....	<b>2-1</b>
Systemübersicht .....	2-2
Grundlagen.....	2-5
IM 208DP - Master mit RS485 - Aufbau.....	2-10
IM 208DP - Master mit RS485 - Einsatz an einer CPU 21x.....	2-13
IM 208DP - Master mit RS485 - Projektierung .....	2-14
IM 208DP - Master mit RS485 - Urlöschen .....	2-22
IM 208DPO - Master mit LWL - Aufbau .....	2-23
IM 208DPO - Master mit LWL - Einsatz an einer CPU 21x .....	2-26
IM 208DPO - Master mit LWL - Projektierung.....	2-27
IM 253DP - Slave (Standard) - Aufbau.....	2-33
IM 253 DPR - Slave (redundant) - Aufbau .....	2-36
IM 253DP, DO 24xDC 24V - Aufbau .....	2-39
IM 253DP - Slave - Blockschaltbild .....	2-43
IM 253DP - Slave - Projektierung.....	2-44
IM 253DP - Slave - Parameter .....	2-46
IM 253DP - Slave - Diagnosefunktionen .....	2-47
Aufbaurichtlinien .....	2-54
Inbetriebnahme.....	2-64
Einsatz der Diagnose-LEDs.....	2-65
Beispiele zur Profibus-Kommunikation.....	2-66
Technische Daten.....	2-74

## Systemübersicht

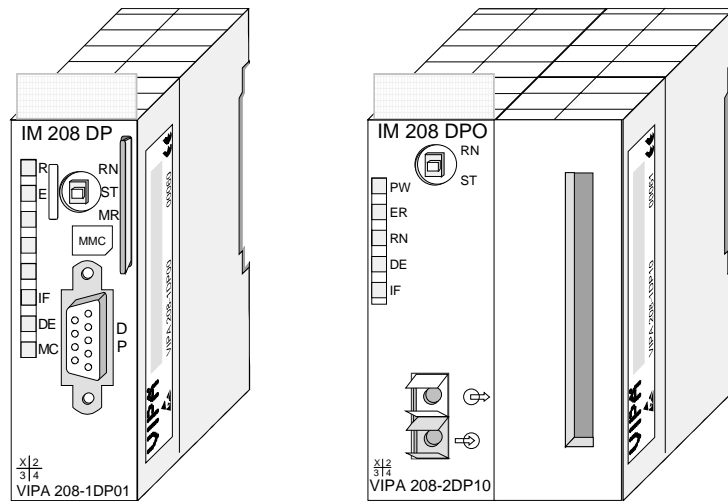
### System 200V Profibus-DP Module

Die meisten System 200V-Profibus-Module sind sowohl mit RS485- als auch mit LWL-Anschluss verfügbar. Folgende Profibus-Modul-Gruppen sind zur Zeit erhältlich:

- Profibus-DP-Master
- Profibus-DP-Slave
- Profibus-DP-Slave Kombimodule
- CPU 21x DP - CPU 21x für S7 von Siemens mit integriertem Profibus-DP-Slave (siehe Handbuch HB103)
- CPU 24x DP - CPU 24x für S5 von Siemens mit integriertem Profibus-DP-Slave (siehe Handbuch HB99)

### Profibus- DP-Master

- Profibus-DP-Master der Klasse 1
- Projektierung unter WinNCS von VIPA
- Projektierung unter COM Profibus von Siemens möglich
- Projektierdaten werden im internen Flash-ROM abgelegt oder auf Flash-Memory-Karte gespeichert.

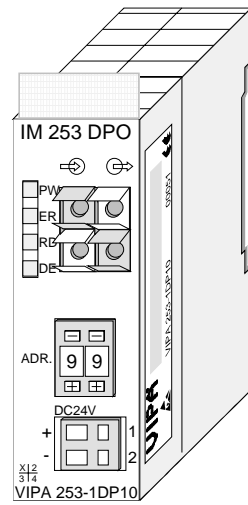
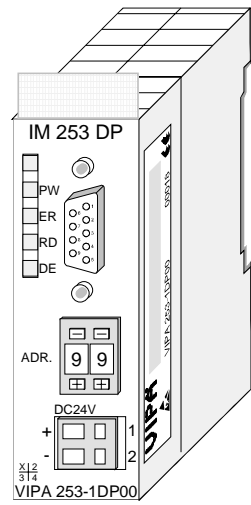


### Bestelldaten DP-Master

Typ	Bestellnummer	Beschreibung
IM 208DP	VIPA 208-1DP01	Profibus-DP-Master mit RS485
IM 208DPO	VIPA 208-2DP10	Profibus-DP-Master mit LWL

**Profibus-DP-Slaves (Standard)**

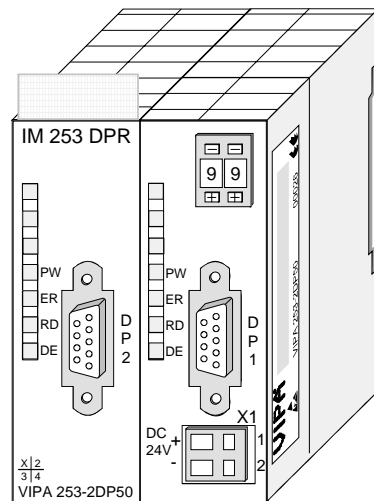
- Ausführung mit RS485-Schnittstelle oder Lichtwellenleiter-Anschluss
- Online Diagnoseprotokoll mit Zeitstempel



**Bestelldaten**

Typ	Bestellnummer	Beschreibung
IM 253DP	VIPA 253-1DP00	Profibus-DP -Slave
IM 253DPO	VIPA 253-1DP10	Profibus-DP -Slave mit LWL-Interface

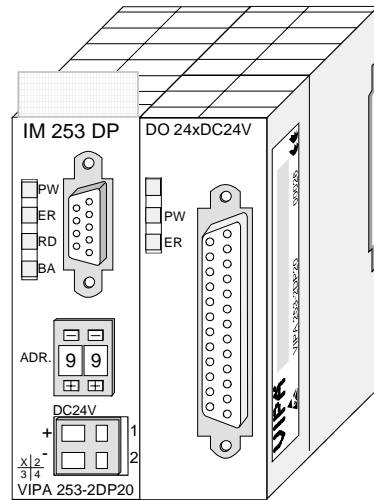
**Profibus-DPR-Slave (redundant)**



**Bestelldaten**

Typ	Bestellnummer	Beschreibung
IM 253DPR	VIPA 253-2DP50	Profibus-DP-Slave 2 Kanal redundant

**Profibus-DP-Slave  
(Kombimodule)**



**Bestelldaten**

Typ	Bestellnummer	Beschreibung
IM 253DP DO 24xDC24V	VIPA 253-2DP20	Profibus-DP-Slave mit 24fach DO

## Grundlagen

### Allgemein

Profibus ist ein internationaler offener Feldbus-Standard für Gebäude-, Fertigungs- und Prozessautomatisierung. Profibus legt die technischen und funktionellen Merkmale eines seriellen Feldbus-Systems fest, mit dem verteilte digitale Feldautomatisierungsgeräte im unteren (Sensor-/Aktor-Ebene) bis mittleren Leistungsbereich (Prozessebene) vernetzt werden können.

Profibus besteht aus einem Sortiment kompatibler Varianten. Die hier angeführten Angaben beziehen sich auf den Profibus-DP.

### Profibus-DP

Profibus-DP ist besonders geeignet für die Fertigungsautomatisierung. DP ist sehr schnell, bietet "Plug and Play" und ist eine kostengünstige Alternative zur Parallelverkabelung zwischen SPS und dezentraler Peripherie. Profibus-DP ist für den schnellen Datenaustausch auf der Sensor-Aktor-Ebene konzipiert.

Der Datenaustausch "DataExchange" erfolgt zyklisch. Während eines Buszyklus liest der Master die Eingangswerte der Slaves und schreibt neue Ausgangsinformationen an die Slaves.

### Master und Slaves

Profibus unterscheidet zwischen aktiven Stationen (Master) und passiven Stationen (Slave).

#### *Master-Geräte*

Master-Geräte bestimmen den Datenverkehr auf dem Bus. Es dürfen auch mehrere Master an einem Profibus eingesetzt werden. Man spricht dann von Multi-Master-Betrieb. Durch das Busprotokoll wird ein logischer Tokenring zwischen den intelligenten Geräten aufgebaut. Nur der Master, der in Besitz des Tokens ist, kommuniziert mit seinen Slaves gerade.

Ein Master (IM 208DP bzw. IM 208DPO) darf Nachrichten ohne externe Aufforderung aussenden, wenn er im Besitz der Buszugriffsberechtigung (Token) ist. Master werden im Profibus-Protokoll auch als aktive Teilnehmer bezeichnet.

#### *Slave-Geräte*

Ein Profibus-Slave stellt Daten von Peripheriegeräten, Sensoren, Aktoren und Messumformern zur Verfügung. Die VIPA Profibus-Koppler (IM 253DP, IM 253DPO und die CPU 24xDP, CPU 21xDP) sind modulare Slave-Geräte, die Daten zwischen der System 200V Peripherie und dem übergeordneten Master transferieren.

Diese Geräte haben gemäß der Profibus-Norm keine Buszugriffsberechtigung. Sie dürfen nur Nachrichten quittieren oder auf Anfrage eines Masters Nachrichten an diesen übermitteln. Slaves werden auch als passive Teilnehmer bezeichnet.

---

**Kommunikation**

Das Busübertragungsprotokoll bietet zwei Verfahren für den Buszugriff:

**Master mit Master**

Die Master-Kommunikation wird auch als Token-Passing-Verfahren bezeichnet. Das Token-Passing-Verfahren garantiert die Zuteilung der Buszugriffsberechtigung. Das Zugriffsrecht auf den Bus wird zwischen den Geräten in Form eines "Token" weitergegeben. Der Token ist ein spezielles Telegramm, das über den Bus übertragen wird.

Wenn ein Master den Token besitzt, hat er das Buszugriffsrecht auf den Bus und kann mit allen anderen aktiven und passiven Geräten kommunizieren. Die Tokenhaltezeit wird bei der Systemkonfiguration bestimmt. Nachdem die Tokenhaltezeit abgelaufen ist, wird der Token zum nächsten Master weitergegeben, der dann den Buszugriff hat und mit allen anderen Geräten kommunizieren kann.

**Master-Slave-Verfahren**

Der Datenverkehr zwischen dem Master und den ihm zugeordneten Slaves wird in einer festgelegten, immer wiederkehrenden Reihenfolge automatisch durch den Master durchgeführt. Bei der Projektierung bestimmen Sie die Zugehörigkeit des Slaves zu einem bestimmten Master. Weiter können Sie definieren, welche DP-Slaves für den zyklischen Nutzdatenverkehr aufgenommen oder ausgenommen werden.

Der Datentransfer zwischen Master und Slave gliedert sich in Parametrierungs-, Konfigurations- und Datentransfer-Phasen. Bevor ein DP-Slave in die Datentransfer-Phase aufgenommen wird, prüft der Master in der Parametrierungs- und Konfigurationsphase, ob die projektierte Konfiguration mit der Ist-Konfiguration übereinstimmt. Überprüft werden Gerätetyp, Format- und Längeninformatoren und die Anzahl der Ein- und Ausgänge. Sie erhalten so einen zuverlässigen Schutz gegen Parametrierfehler.

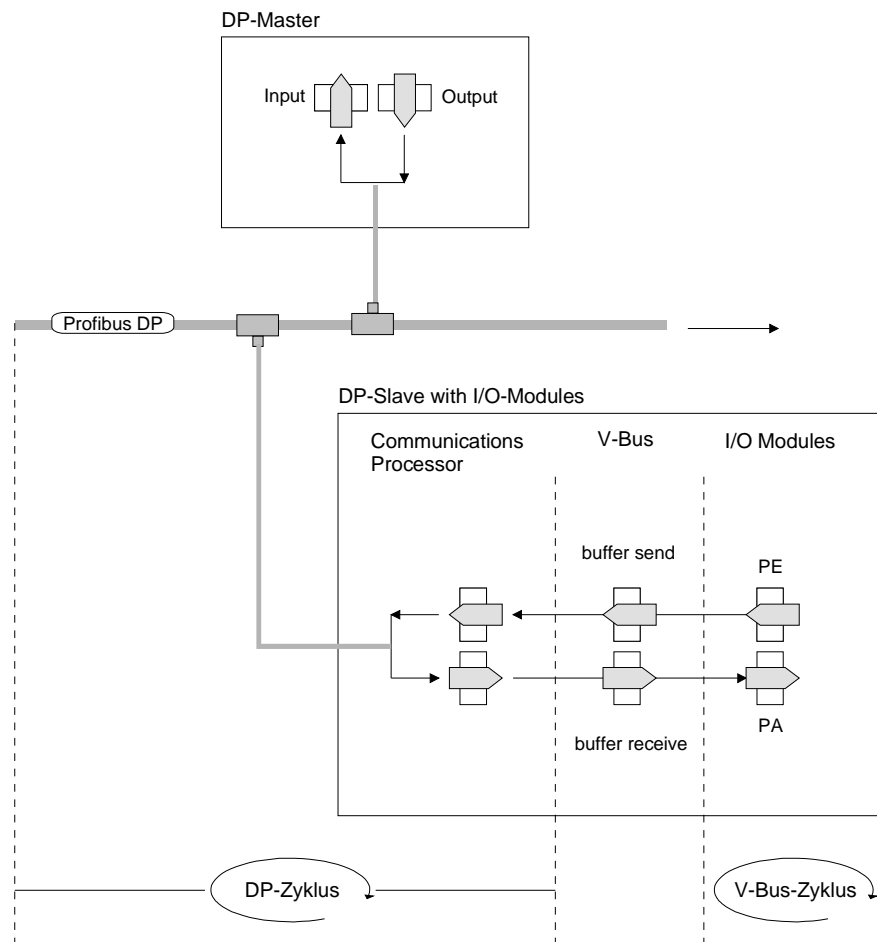
Zusätzlich zum Nutzdatentransfer den der Master selbständig durchführt, können Sie neue Parametrierdaten an einen Bus-Koppler schicken.

Im Zustand DE "DataExchange" sendet der Master neue Ausgangsdaten an den Slave und im Antworttelegramm des Slaves werden die aktuellen Eingangsdaten an den Master übermittelt.



## Funktionsweise der Datenübertragung

Der Datenaustausch zwischen DP-Master und DP-Slave erfolgt zyklisch über Sende- und Empfangspuffer.



PE: Prozessabbild der Eingänge

PA: Prozessabbild der Ausgänge

## V-Bus-Zyklus

In einem V-Bus-Zyklus (V-Bus=VIPA-Rückwandbus) werden alle Eingangsdaten der Module im PE gesammelt und alle Ausgangsdaten des PA an die Ausgabe-Module geschrieben. Nach erfolgtem Datenaustausch wird das PE in den Sendepuffer (buffer send) übertragen und die Inhalte des Empfangspuffers (buffer receive) nach PA transferiert.

## DP-Zyklus

In einem Profibus-Zyklus spricht der Master alle seine Slaves der Reihe nach mit einem DataExchange an. Beim DataExchange werden die dem Profibus zugeordneten Speicherbereiche geschrieben bzw. gelesen.

Danach wird der Inhalt des Profibus-Eingangsbereichs in den Empfangspuffer (buffer receive) geschrieben und die Daten des Sendepuffers (buffer send) in den Profibus-Ausgangsbereich übertragen.

Der Datenaustausch zwischen DP-Master und DP-Slave über den Bus erfolgt zyklisch, unabhängig vom V-Bus-Zyklus

**V-Bus-Zyklus ≤ DP-Zyklus**

Zur Gewährleistung einer zeitgleichen Datenübertragung sollte die V-Bus-Zykluszeit immer kleiner oder gleich der DP-Zykluszeit sein.

In der mitgelieferten GSD-Datei befindet sich der Parameter **min\_slave\_interval = 3ms**.

Für einen durchschnittlichen Aufbau wird garantiert, dass spätestens nach 3ms die Profibus-Daten am V-Bus aktualisiert wurden. Sie dürfen also alle 3ms einen DataExchange mit dem Slave ausführen.

**Hinweis!**

Sobald der V-Bus-Zyklus länger dauert als der DP-Zyklus, erlischt die RUN-LED am VIPA-Profibus-Slave.

Diese Funktion wird ab dem Hardwareausgabestand 6 unterstützt.

**Datenkonsistenz**

Daten bezeichnet man als konsistent, wenn sie inhaltlich zusammengehören. Inhaltlich gehören zusammen: das High- und Low-Byte eines Analogwerts (wortkonsistent) und das Kontroll- und Status-Byte mit zugehörigem Parameterwort für den Zugriff auf die Register.

Die Datenkonsistenz ist im Zusammenspiel von Peripherie und Steuerung grundsätzlich nur für 1 Byte sichergestellt. Das heißt, die Bits eines Bytes werden zusammen eingelesen bzw. ausgegeben. Für die Verarbeitung digitaler Signale ist eine byteweise Konsistenz ausreichend.

Für Daten, deren Länge ein Byte überschreiten, wie z.B. bei Analogwerten muss die Datenkonsistenz erweitert werden.

VIPA Profibus-DP-Master garantieren ab Firmware-Version V3.00 eine Konsistenz über die erforderliche Länge.

**Einschränkungen**

- Max. 125 DP-Slaves an einem DP-Master - max. 32 Slaves/Segment
- Max. 16 DPO-Slaves an einem DPO-Master bei 1,5Mbaud
- Peripherie-Module dürfen nur nach Power-Off gesteckt oder gezogen werden!
- Max. Leitungslänge unter RS485 zwischen zwei Stationen 1200m (baudratenabhängig)
- Max. Leitungslänge unter LWL zwischen zwei Stationen 300m (bei HCS-LWL) und 50m (bei POF-LWL)
- Die maximale Baudrate liegt bei 12Mbaud
- Die Profibus-Adresse darf während des Betriebs nicht verstellt werden

**Diagnose**

Die umfangreichen Diagnosefunktionen unter Profibus-DP ermöglichen eine schnelle Fehlerlokalisierung. Die Diagnosedaten werden über den Bus übertragen und beim Master zusammengefasst.

---

**Übertragungs-  
medium**

Profibus verwendet als Übertragungsmedium eine geschirmte, verdrehte Zweidrahtleitung auf Basis der RS485-Schnittstelle oder eine Duplex-Lichtwellenleitung (LWL). Die Übertragungsrate liegt bei beiden Systemen bei maximal 12MBaud.

Nähere Angaben hierzu finden Sie in den "Aufbaurichtlinien".

**Elektrisches  
System über  
RS485**

Die RS485-Schnittstelle arbeitet mit Spannungsdifferenzen. Sie ist daher unempfindlicher gegenüber Störeinflüssen als eine Spannungs- oder Stromschnittstelle. Sie können das Netz sowohl als Linien-, als auch als Baumstruktur konfigurieren. Auf Ihrem VIPA Profibus-Koppler befindet sich eine 9-polige Buchse. Über diese Buchse koppeln Sie den Profibus-Koppler als Slave direkt in Ihr Profibus-Netz ein.

Die Busstruktur unter RS485 erlaubt das rückwirkungsfreie Ein- und Auskoppeln von Stationen oder die schrittweise Inbetriebnahme des Systems. Spätere Erweiterungen haben keinen Einfluss auf Stationen, die bereits in Betrieb sind. Es wird automatisch erkannt, ob ein Teilnehmer ausgefallen oder neu am Netz ist.

**Optisches  
System über  
Lichtwellenleiter**

Das Lichtwellenleitersystem arbeitet mit Lichtimpulsen von monochromatischem Licht. Der Lichtwellenleiter ist völlig unempfindlich gegenüber Störspannungen von außen. Ein Lichtwellenleitersystem wird in Linienstruktur aufgebaut. Jedes Gerät ist mit einem Hin- und Rückleiter zu verbinden. Ein Abschluss am letzten Gerät ist nicht erforderlich.

Das rückwirkungsfreie Ein- und Auskoppeln von Stationen ist aufgrund der Linienstruktur nicht möglich.

---

**Adressierung**

Jeder Teilnehmer am Profibus identifiziert sich mit einer Adresse. Diese Adresse darf nur einmal in diesem Bussystem vergeben sein und kann zwischen 0 und 125 liegen. An den VIPA Profibus-Kopplern stellen Sie die Adresse mit dem Adressierungsschalter an der Front ein.

Dem VIPA Profibus-Master müssen Sie die Adresse bei der Projektierung zuteilen.

**GSD-Datei**

Zur Konfiguration einer Slave-Anschaltung in Ihrem eigenen Projektierool bekommen Sie die Leistungsmerkmale der VIPA-Komponenten in Form einer GSD-Datei mitgeliefert.

Aufbau und Inhalt der GSD-Datei sind durch die Profibus Nutzerorganisation (PNO) genormt und können dort jederzeit abgerufen werden.

**Die GSD-Datei für die VIPA Profibus-DP-Slaves lautet: VIPA0550.GSD**

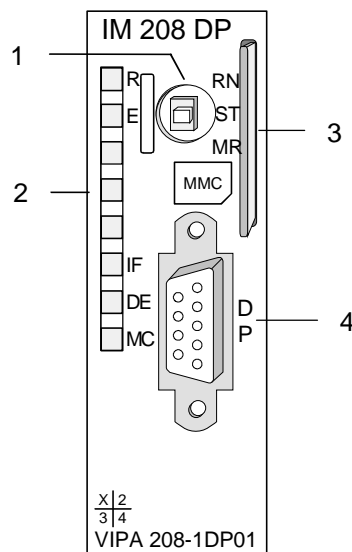
Installieren Sie diese GSD-Datei in Ihrem Projektierool. Nähere Hinweise zur Installation der GSD-Datei finden Sie im Handbuch zu Ihrem Projektierool.

## IM 208DP - Master mit RS485 - Aufbau

### Eigenschaften

- Profibus-DP-Master der Klasse 1
- 125 DP-Slaves an einen DP Master ankoppelbar
- Blendet Datenbereich der Slaves über den V-Bus im Adressbereich der CPU 2xx ein
- Projektierung unter WinNCS von VIPA bzw. STEP<sup>®</sup>7 Manager und ComProfibus von Siemens
- Diagnosefähig

### Frontansicht IM 208DP



- [1] Betriebsarten-Schalter  
RUN/STOP/MR
- [2] LED Statusanzeigen
- [3] Steckplatz für MMC
- [4] RS485-Schnittstelle

### Komponenten

#### LEDs

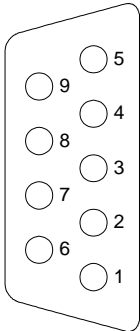
Die Master-Module besitzen verschiedene LEDs, die der Busdiagnose dienen und den eigenen Betriebszustand anzeigen. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bez.	Farbe	Bedeutung
R	Grün	Leuchtet nur R befindet sich der Master im RUN. Die Slaves werden angesprochen und die Ausgänge sind 0 ("clear"-Zustand). Leuchten R+DE befindet sich der Master im "operate"-Zustand. Er tauscht Daten mit den Slaves aus. 3x Blinken: Transfer von MMC in Flash-ROM war fehlerfrei.
E	Rot	Leuchtet bei Slave-Ausfall (ERROR). 3x Blinken: Fehler bei Transfer von MMC in Flash-ROM.
IF	Rot	Initialisierungsfehler für fehlerhafte Parametrierung.
DE	Gelb	DE (DataExchange) zeigt an, dass eine Kommunikation über Profibus stattfindet.
MC	Gelb	Zeigt Bereitschaft für Datentransfer von einer MMC an.

**RS485-Schnittstelle**

Der Profibus-Master von VIPA wird über eine 9polige Buchse in das Profibus-System eingebunden.

Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung



Pin	Belegung
1	Schirm
2	n.c.
3	RxD/TxD-P
4	CNTR-P
5	GND
6	5V (max. 70mA)
7	n.c.
8	RxD/TxD-N
9	n.c.

**Spannungsversorgung**

Der Profibus-Master bezieht seine Spannungsversorgung über den Rückwandbus.

**Betriebsart-Schalter**

Mit dem Betriebsart-Schalter können Sie zwischen den Betriebsarten STOP (ST), RUN (RN) und Master Reset (MR) wählen.

Bei Betriebsartschalter auf RN und gültigen Parametern, geht der Master in den RUN-Zustand über.

Wird der Betriebsart-Schalter auf ST gestellt, geht der Master in den STOP-Zustand über. Er beendet die Kommunikation, worauf alle Ausgänge auf 0 gelegt werden und sendet einen Alarm an das übergeordnete System.

Eine ausführliche Erklärung zu den Übergängen zwischen RUN und STOP finden Sie in diesem Kapitel unter "Betriebszustände".

Mit der Tasterstellung MR können Sie folgendes auslösen:

- Datentransfer von MMC in das Flash-ROM
- Serieller Modus zum Einsatz des Green Cable von VIPA
- Urlösches des DP-Masters

Näheres zu diesen Möglichkeiten finden Sie in diesem Kapitel weiter unten.

**MMC als externes Speichermedium**

Als externes Speichermedium kommt die MMC-Speicherkarte von VIPA unter der Best.-Nr.: VIPA 953-0KX00 zum Einsatz. Von VIPA erhalten Sie ein externes MMC-Lesegerät (Best.-Nr: VIPA 950-0AD00). Hiermit können Sie Ihre MMCs am PC beschreiben bzw. lesen.

Die Übertragung Ihrer Projektdaten von der MMC in Ihren Master wird über die MR-Taststellung des Betriebsart-Schalters angestoßen. Näheres hierzu finden Sie weiter unten unter "Projekt transferieren".

---

**Betriebszustände, Anlaufverhalten***Power On*

Der DP-Master wird mit Spannung versorgt. Sofern sich der Betriebsart-Schalter des DP-Master in Stellung RN befindet, erhält dieser automatisch von der CPU seine Projektierung, falls eine Hardwarekonfiguration für den DP-Master vorliegt und geht automatisch in RUN über.

*STOP*

Im STOP-Zustand und bei gültigen Parametern, sind die Ausgänge der zugeordneten Slaves auf 0 gesetzt. Es findet zwar keine Kommunikation statt, aber der Master ist mit seinen aktuellen Busparametern aktiv am Bus, und belegt die ihm zugeteilte Adresse am Bus. Zur Freigabe dieser Adresse ist am DP-Master der Profibus-Stecker zu ziehen.

*STOP → RUN*

In Stellung RN bootet der Master erneut. Bitte beachten Sie, dass hierbei immer die Projektierdaten aus dem Flash-ROM verwendet werden. **Eine schon vorhandenen Hardwarekonfiguration wird durch den Bootvorgang im DP-Master gelöscht.**

Zur erneuten Übertragung einer Hardwarekonfiguration ist an der CPU ein Power ON durchzuführen.

Bei einem STOP → RUN Übergang baut der DP-Master eine Kommunikation zu den Slaves auf. Während dieser Zeit brennt nur die R-LED. Bei erfolgter Kommunikation geht der DP-Master in RUN. Der DP-Master zeigt dies über die LEDs R und DE an.

Bei fehlerhaften Parametern bleibt der DP-Master im STOP-Zustand und signalisiert einen Parametrierfehler mit der IF-LED. Der DP-Master befindet sich nun mit folgenden Default-Parameter aktiv am Bus:

**Default-Bus-Parameter: Adresse: 1, Übertragungsrate: 1,5MBAud.**

*RUN*

Im RUN leuchten die R- und DE-LEDs. Jetzt können Daten ausgetauscht werden. Im Fehlerfall wie z.B. Slave-Ausfall, wird dies über die E-LED angezeigt und ein Alarm an die CPU abgesetzt.

*RUN → STOP*

Der Master geht in den STOP-Zustand über. Er beendet die Kommunikation worauf alle Ausgänge auf 0 gelegt werden und sendet einen Alarm an die CPU.

## IM 208DP - Master mit RS485 - Einsatz an einer CPU 21x

**Kommunikation** Über die IM 208 DP-Master-Module können bis zu 125 Profibus-DP-Slaves an eine System 200V CPU angekoppelt werden. Der Master kommuniziert mit den Slaves und blendet die Datenbereiche über den Rückwandbus im Adressbereich der CPU ein. Es dürfen maximal 1024 Byte Eingangs- und 1024 Byte Ausgangsdaten entstehen.

Bei Firmwareständen < V3.0.0 stehen nur jeweils 256Byte für Ein- und Ausgabedaten zur Verfügung.

Bei jedem Neustart der CPU holt sich diese von allen Mastern die I/O-Mapping-Daten.

**Alarmbearbeitung** Die Alarmbearbeitung ist aktiviert, d.h. eine IM 208-Fehlermeldung kann folgende Alarme auslösen, die die CPU veranlassen, die entsprechenden OBs aufzurufen:

- Prozessalarm: OB40
- Diagnosealarm: OB82
- Slaveausfall: OB86

Sobald das BASP-Signal (Befehlsausgabesperre) von der CPU kommt, stellt der DP-Master die Ausgänge der angeschlossenen Peripherie auf Null.



### Hinweis!

Das Prozessabbild der Eingänge behält nach einem Slaveausfall den Zustand wie vor dem Slaveausfall.

**Voraussetzungen** Bitte beachten Sie bei Einsatz des IM 208 Profibus-DP-Master, dass dieser einen Firmwarestand ab V3.0.0 besitzt; ansonsten kann dieser an einer CPU 21x mit Firmwarestand ab V3.0.0 nicht betrieben werden.

Den jeweiligen Firmwarestand finden Sie auf einem Aufkleber auf der Rückseite des Moduls.

Bei Fragen zum Firmware-Update wenden Sie sich bitte an den VIPA Support (support@vipa.de).

Nähere Angaben zur Anbindung an Ihre CPU finden Sie in der Dokumentation zu Ihrer CPU.



### Hinweis!

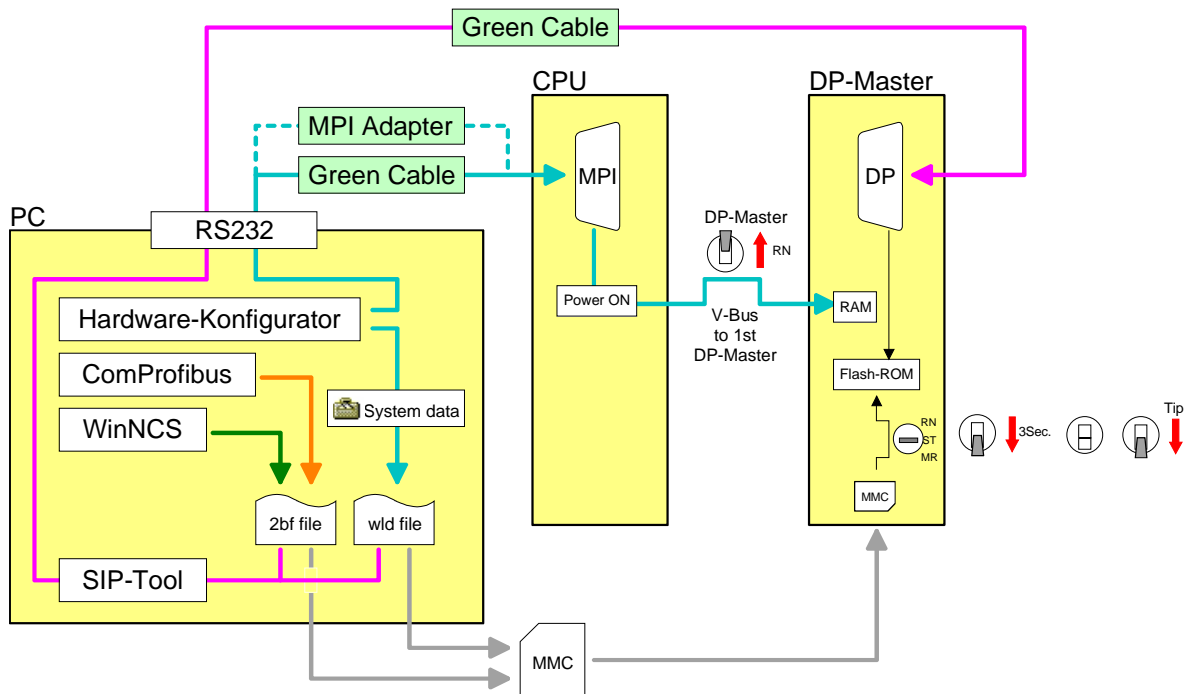
Für den Einsatz an einer CPU 24x ist die Firmware des DP-Masters auf Stand V1.0.7 zu bringen.

## IM 208DP - Master mit RS485 - Projektierung

### Übersicht

Zur Projektierung des DP-Masters haben Sie folgende Möglichkeiten:

- Projektierung im Hardware-Konfigurator von Siemens und Transfer als Hardwarekonfiguration über den Systembus. Hierüber können Sie ausschließlich den 1. Master (IM 208 bzw. CPU 21xDPM) projektieren.  
**Bitte beachten Sie, dass sich der Betriebsart-Schalter des DP-Master für die Projektierung über den Systembus in RN befinden muss.**
- Projektierung im Hardware-Konfigurator von Siemens und Export einer wld-Datei.
- Projektierung über WinNCS von VIPA bzw. ComProfibus von Siemens und Export einer 2bf-Datei.



### Erforderliche Firmwarestände

DP-Master und CPU sollen einen Firmwarestand ab V3.0.0 besitzen, da ansonsten der DP-Master an der CPU 21x nicht betrieben werden kann. Den jeweiligen Firmwarestand finden Sie auf einem Aufkleber auf der Rückseite des Moduls.

Firmwareversion

DP-Master	CPU	Eigenschaft
V3.0.0	V3.0.0	1024Byte Ein- und Ausgangsdaten
V3.0.4	V3.0.0	Projektierung über wld-Datei
V3.0.6	V3.3.0	Projektierung als Hardwarekonfiguration über MPI
V3.0.6	-----	Urlöschen des DP-Master



**Projektierung über Hardware-konfiguration**

Sie projektieren im Hardware-Konfigurator von Siemens Ihr SPS-System zusammen mit dem DP-Master. Diese "Hardwarekonfiguration" übertragen Sie via MPI in die CPU. Bei Power ON werden die Projektierdaten in den DP-Master übertragen sofern sich dessen Betriebsart-Schalter in Stellung RN befindet.



**Hinweis!**

Bitte beachten Sie, dass sich der Betriebsart-Schalter des DP-Masters in Stellung RN befinden muss, da ansonsten beim STOP-RUN-Übergang der Master neu bootet und die Projektierung gelöscht wird.

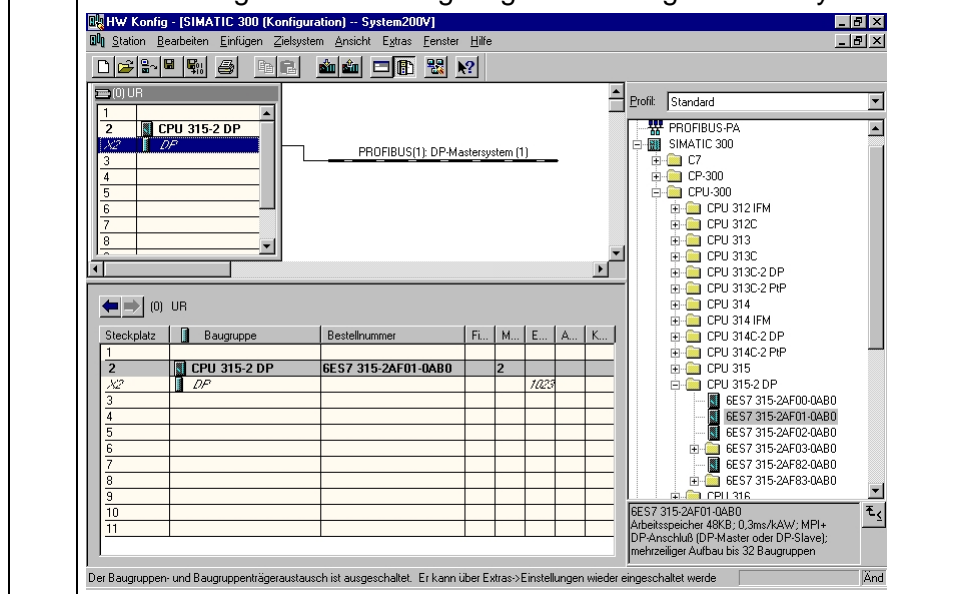
Weiter ist zu beachten, dass hierbei immer nur der 1. DP-Master im System projektiert werden kann! Zur Projektierung weiterer DP-Master ist das jeweilige Projekt als wld-Datei bzw. 2bf-Datei zu exportieren.

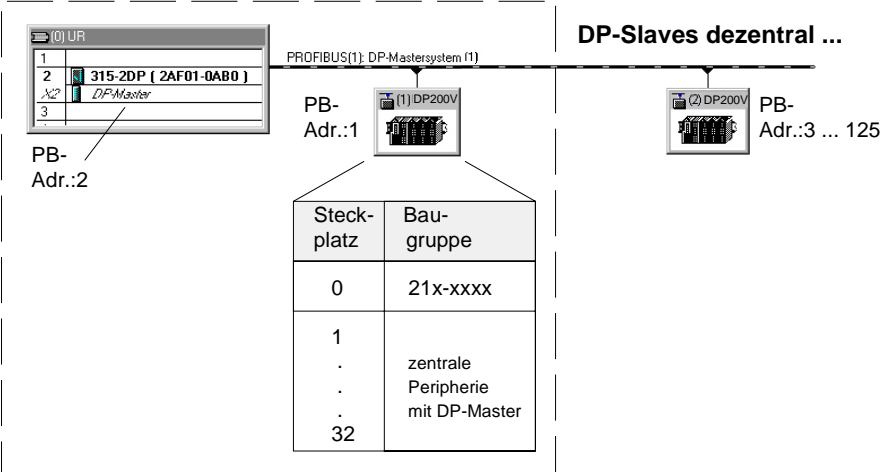

**Vorgehensweise zur Hardware-konfiguration**

Die Schritte zur Projektierung im Hardware-Konfigurator sollen hier kurz aufgezeigt werden:

1. Legen Sie ein neues Projekt System 300 an und fügen Sie aus dem Hardwarekatalog eine Profilschiene ein.
2. Fügen Sie die CPU 315-2DP ein. Sie finden die CPU mit Profibus-Master im Hardwarekatalog unter:  
*Simatic300 > CPU-300 > CPU315-2DP > 6ES7 315-2AF01-0AB0*
3. Geben Sie Ihrem Master eine Profibus-Adresse von 2 aufsteigend.
4. Klicken Sie auf DP und stellen Sie unter *Objekteigenschaften* die Betriebsart "DP-Master" ein. Bestätigen Sie Ihre Eingabe mit OK.
5. Durch Klick mit der rechten Maustaste auf "DP" öffnet sich das Kontextmenü. Wählen Sie "Mastersystem einfügen" aus. Legen Sie über NEU ein neues Profibus-Subnetz an.

Die nachfolgende Abbildung zeigt das erzeugte Mastersystem:



6. Um zum STEP<sup>®</sup>7 Projektiertool von Siemens kompatibel zu sein, ist die System 200V CPU explizit einzubinden.  
 Hängen Sie hierzu an das Subnetz das System "VIPA\_DP200V\_2". Sie finden dies im Hardware-Katalog unter:  
*PROFIBUS DP > Weitere Feldgeräte > IO > VIPA\_System\_200V > VIPA\_DP200V\_2.*  
 Geben Sie diesem Slave die Profibus-Adresse 1.  
 Platzieren Sie auf dem Steckplatz 0 die entsprechende CPU 21x von VIPA, indem Sie diese im Hardware-Katalog unter *VIPA\_CPU21x* auswählen.  
**Der Steckplatz 0 ist zwingend erforderlich!**
7. Zur Einbindung der am VIPA-Bus befindlichen Module ziehen Sie aus dem Hardware-Katalog unter *VIPA\_DP200V\_2* die entsprechenden System 200V Module auf die Steckplätze unterhalb der CPU. Beginnen Sie mit Steckplatz 1. Auf diese Weise platzieren Sie auch Ihren DP-Master.
8. Zur Projektierung von DP-Slaves, die an den DP-Master angekoppelt sind, verfahren Sie auf die gleiche Weise wie unter 6. (Projektierung des VIPA\_DP200V\_2-Systems).  
 Entnehmen Sie aus dem Hardware-Katalog das entsprechende Profibus-System und ziehen Sie dies auf das DP-Master Subnetz.  
 Vergeben Sie dem Slave eine Adresse > 3.  
**CPU 21x zentral**
- 
- | Steckplatz | Baugruppe                         |
|------------|-----------------------------------|
| 0          | 21x-xxxx                          |
| 1          | zentrale Peripherie mit DP-Master |
| .          |                                   |
| .          |                                   |
| 32         |                                   |
9. Klicken Sie auf  (speichern und übersetzen).

**Transfervarianten** Abhängig von der eingesetzten Firmware auf CPU und DP-Master haben Sie nun folgende Transfermöglichkeiten:

1. Transfer via MPI (nur für 1. Master am Bus)  
Ihre DP-Master-Projektierung wird zusammen mit Ihrem SPS-Programm an die CPU übertragen. Die CPU reicht bei Power ON die DP-Master-Projektierung automatisch weiter an den 1. DP-Master (IM 208DP oder CPU 21xDPM), der sich am Systembus befindet.
2. Export des Projekts als wld-Datei auf MMC  
Exportieren Sie Ihre Projektierung als wld-Datei und übertragen Sie dieses auf eine MMC-Karte. Die MMC-Karte stecken Sie in den entsprechenden DP-Master.
3. Export des Projekts als wld-Datei und Datentransfer via Green Cable unter Einsatz des SIP-Tools von VIPA.

**zu 1.  
Transfer via MPI**

10.	Verbinden Sie Ihr PG bzw. Ihren PC über MPI mit Ihrer CPU. Zur seriellen Punkt-zu-Punkt-Übertragung von Ihrem PC an die CPU können Sie auch das Green Cable von VIPA verwenden. Das Green Cable hat die Best.-Nr. VIPA 950-0KB00 und darf ausschließlich geeigneten Modulen der VIPA eingesetzt werden. Bitte beachten Sie hierzu die Hinweise zum Green Cable in den Grundlagen! Bei Einsatz des Green Cable von VIPA ist die MPI-Schnittstelle entsprechend zu konfigurieren (PC Adapter MPI, 38400Baud).
11.	Schalten Sie Ihren DP-Master in RUN.
12.	Schalten Sie die Spannungsversorgung Ihrer CPU ein.
13.	Übertragen vom Hardware-Konfigurator von Siemens über den Menüpunkt <b>Zielsystem</b> > <i>Laden in Baugruppe</i> Ihr Projekt in die CPU.
14.	Zur zusätzlichen Sicherung Ihres Projekts auf MMC stecken Sie eine MMC in die CPU 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.

Die Projektierung ist hiermit abgeschlossen.



**Hinweis!**

Bitte beachten Sie, dass durch die oben aufgeführte Methode immer nur der 1. DP-Master im System projiziert werden kann. Zur Projektierung weiterer DP-Master ist ein Einsatz einer MMC mit einer exportierten wld-Datei erforderlich.

### zu 2. Export als dpm.wld auf MMC

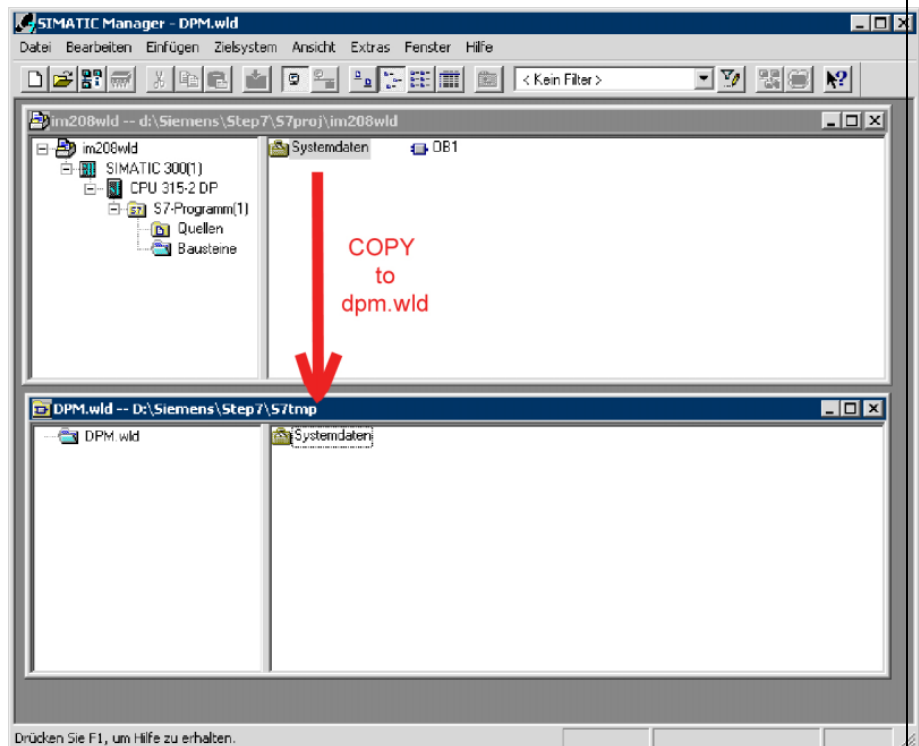
Zur Projektierung weiterer DP-Master exportieren Sie Ihr Projekt in Form einer wld-Datei auf eine MMC. Die MMC stecken sie in den entsprechenden DP-Master. Unter Zuhilfenahme des Betriebsartenschalters können Sie Ihr Projekt von der MMC in das Flash-ROM des Masters übertragen.

Nach der Übertragung kann die MMC wieder entnommen werden.

Auf diese Weise können Sie mehrere DP-Master mit einer MMC projektieren, die sich am gleichen Rückwandbus befinden.

#### Vorgehensweise

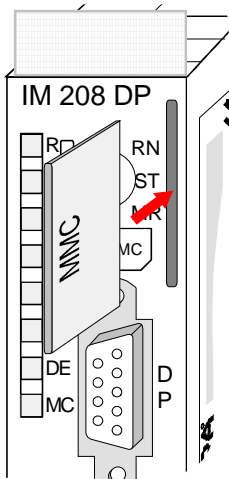
- |      |  |
|------|--|
| 10.. | Legen Sie mit <b>Datei</b> > <i>Memory Card</i> > <i>Neu...</i> eine neue wld-Datei an. Damit diese Datei vom Profibus-Master übernommen wird, muss diese den Namen <b>dpm.wld</b> besitzen.<br>→ Die Datei wird zusätzlich zum Projektfenster eingeblendet. |
| 11.  | Gehen Sie nun in Ihr Projekt in das Verzeichnis <i>Bausteine</i> und kopieren Sie das Verzeichnis "Systemdaten" in die neu angelegte dpm.wld-Datei.  |



#### Hinweis!

Soll ein schon bestehendes "Systemdaten"-Verzeichnis überschrieben werden, so ist zuvor das zu überschreibende Verzeichnis "Systemdaten" zu löschen.

**zu 2. Fortsetzung  
Daten von MMC in  
internes Flash-ROM  
transferieren**



12.	Übertragen Sie mit einem MMC-Lesegerät die wld-Datei auf eine MMC.
13.	Stecken Sie das MMC-Speichermodul in Ihren IM 208DP-Master
14.	Schalten Sie die Spannungsversorgung Ihres System 200V ein.
15.	Drücken Sie den Betriebsartenschalter des Master-Moduls in Stellung MR. Halten Sie diesen gedrückt, bis die blinkende MC-LED in Dauerlicht übergeht.
16.	<p>Lassen Sie den Schalter los und tippen Sie nochmals kurz in Stellung MR. → Die Daten werden von der MMC in das interne Flash-ROM übertragen. Während der Datenübertragung erlöschen alle LEDs.</p> <p>Bei erfolgreicher Datenübertragung blinkt die grüne R-LED 3mal. Bei Fehler blinkt die rote E-LED 3mal.</p>
17.	Die MMC können Sie jetzt entnehmen.
18.	<p>Schalten Sie den Master von STOP in RUN. → Der IM 208DP-Master startet nun mit dem neuen Projekt im internen Flash-ROM.</p> <p>Die RUN-LED (R) und DE leuchtet.</p>



**Hinweis!**

Befindet sich eine wld- und eine 2bf-Datei auf der MMC, so hat die wld-Datei Vorrang.

Die Projektierung ist hiermit abgeschlossen.

**zu 3.  
Export als dpm.wld  
Transfer über SIP-  
Tool von VIPA**

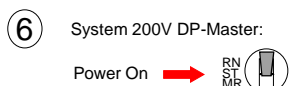
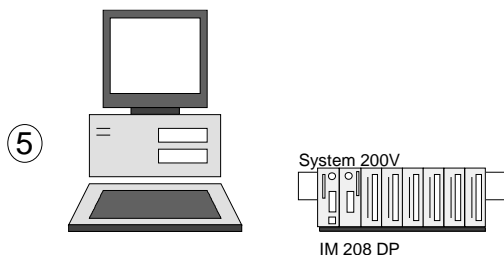
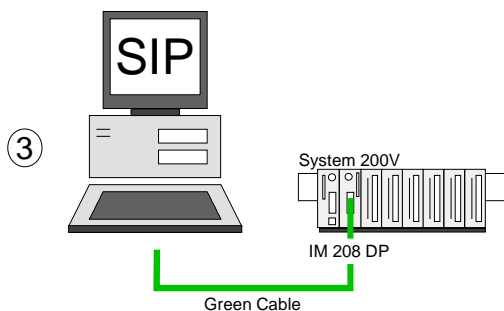
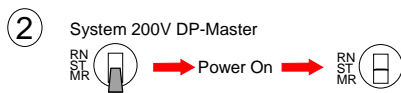
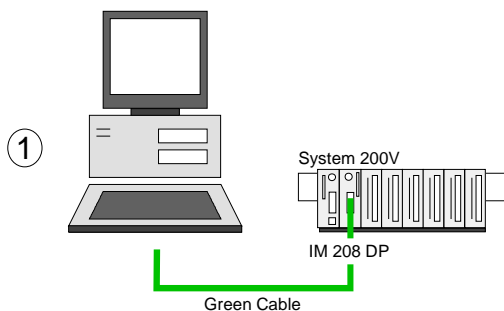
Das SIP-Tool ist ein Transfertool. Sie erhalten es zusammen mit WinNCS von VIPA. Hiermit können Sie unter Einsatz des Green Cable von VIPA Ihr Projekt in Form einer wld- bzw. 2bf-Datei seriell über die Profibus-Schnittstelle in Ihren DP-Master übertragen. Das übertragene Projekt wird im internen Flash-ROM des DP-Master abgelegt.

Das Green Cable ist bei VIPA erhältlich und hat die Best.-Nr.: VIPA 950-0KB00.



**Achtung!**

Bitte beachten Sie die Hinweise zum Einsatz des Green Cable in den "Grundlagen"!







12.	Ziehen Sie den Profibus-Stecker von Ihrem DP-Master.
13.	Stecken Sie das Green Cable auf die serielle Schnittstelle Ihres PC und auf die Profibus-Schnittstelle des DP-Masters.
14.	Halten Sie am Profibus-Master den Betriebsartenschalter in Stellung MR und schalten Sie die Spannungsversorgung ein. Lassen Sie den Betriebsartenschalter wieder los. → Ihr Profibus-Master kann nun über die Profibus-Schnittstelle seriell Daten empfangen.
15.	Schalten Sie den PC ein und starten Sie das mit WinNCS mitgelieferte SIP-Tool. Wählen Sie die entsprechende COM-Schnittstelle und stellen Sie über [Connect] eine Verbindung her. Sobald eine Verbindung aufgebaut ist, erscheint im SIP-Tool in der Statuszeile oben eine OK-Meldung ansonsten eine ERR-Meldung.
16.	Klicken Sie auf [Download], wählen Sie Ihre dpm.2bf- bzw. dpm.wld-Datei und übertragen Sie diese in den DP-Master
17.	Beenden Sie nach der Datenübertragung die Verbindung und das SIP-Tool.
18.	Ziehen Sie das Green Cable vom Master ab.
19.	Schalten Sie die Spannungsversorgung für Ihren Master aus.
20.	Verbinden Sie den Master mit dem Profibus-Netz und schalten Sie die Spannungsversorgung wieder ein.
21.	Bringen Sie Ihren Master in RUN (RN). → Ihr IM 208DP-Profibus-Master befindet sich nun mit den aktuellen Projektdaten am Netz. Die Daten sind im internen Flash-ROM abgelegt.

## Projektierung unter WinNCS

Die Projektierung können Sie auch unter WinNCS von VIPA durchführen und Ihr Projekt als 2bf-Datei auf eine MMC exportieren bzw. mit dem SIP-Tool in den DP-Master übertragen.

Die Schritte für die Projektierung unter WinNCS sollen hier kurz aufgezeigt werden. Näheres zum Einsatz von WinNCS finden Sie auch im zugehörigen Handbuch HB91.

1.	Starten Sie WinNCS und legen Sie mit <b>Datei</b> > <i>anlegen/öffnen</i> eine neue Projektdatei unter der Funktionalität "Profibus" an.
2.	Fügen Sie wenn noch nicht geschehen im Netzwerkfenster mit  eine <b>Profibus-Funktionsgruppe</b> ein und klicken Sie im Parameterfenster auf [übernehmen].
3.	Fügen Sie im Netzwerkfenster mit  einen <b>Profibus-Host/Master</b> ein und geben Sie im Parameterfenster die Profibusadresse Ihres Masters an.
4.	Fügen Sie im Netzwerkfenster mit  einen <b>Profibus-Slave</b> ein. Geben Sie im Parameterfenster die Profibusadresse, die Familie "I/O" und den Stationstyp "VIPA_DP200V_2" ein und klicken Sie auf [übernehmen].
5.	<p>Projektieren Sie über  der Reihe nach alle Peripherie-Module, die über den Rückwandbus mit dem entsprechenden Profibus-Slave verbunden sind.</p> <p>Über [Auto] können Sie die Peripherie automatisch adressieren lassen und die Adressbelegung über [MAP] anzeigen.</p> <p><b>Bitte beachten Sie bitte, dass es bei der automatischen Adressierung nicht zu Adresskonflikten mit der lokalen Peripherie kommt!</b></p> <p>Handelt es sich um ein intelligentes Modul, wie z.B. CP240, erscheinen die hierzu einstellbaren Parameter.</p>
6.	Nachdem Sie alle Slaves mit zugehöriger Peripherie projektiert haben, müssen die Busparameter für den Profibus berechnet werden. Aktivieren Sie hierzu im Netzwerkfenster die Funktionsgruppe Profibus. Klicken Sie im Parameterfenster auf das Register "Busparameter". Stellen Sie die gewünschte Baudrate ein und klicken Sie auf [calculate]. Die Busparameter werden berechnet - [übernehmen] Sie diese. Bei jeder Änderung der Modulzusammenstellung sind die Busparameter neu zu berechnen!
7.	Aktivieren Sie im Netzwerkfenster die Master-Ebene und exportieren Sie Ihr Projekt in die Datei dpm.2bf.
8.	Übertragen Sie Ihre dpm.2bf-Datei in Ihren DP-Master (Transfermöglichkeiten siehe weiter oben unter "Transfervarianten")



### Hinweis!

Da sich der IM 208 DP-Master gleich verhält wie die IM 308-C von Siemens, können Sie diesen auch als IM 308-C unter "ComProfibus" von Siemens projektieren und ihr Projekt als 2bf-Datei exportieren.

# IM 208DP - Master mit RS485 - Urlöschen

**Allgemeines**

Ab der Firmware-Version V3.0.6 des DP-Masters haben Sie die Möglichkeit am DP-Master ein Urlöschen durchzuführen.

Beim Urlöschen werden alle Daten im Flash-ROM gelöscht.

**Urlöschen durchführen**

1.	Schalten Sie die Spannungsversorgung Ihres System 200V ein.
2.	<p>Drücken Sie den Betriebsartenschalter des Master-Moduls in Stellung MR. Halten Sie diesen etwa 9s gedrückt</p> <p>→ es blinkt zunächst 3mal die MC-LED. Das Blinken geht für 3s in Dauerlicht über. Daraufhin blinkt die IF-LED 3mal und geht in Dauerlicht über.</p>
3.	<p>Lassen Sie den Schalter los und tippen Sie innerhalb von 3s nochmals kurz in Stellung MR.</p> <p>→ Die Inhalte des Flash-ROMs werden gelöscht. Der Vorgang ist beendet sobald die grüne R-LED 3mal blinkt und die IF-LED leuchtet.</p> <div style="text-align: center;"> </div> <p>Sobald Sie den Master in RUN schalten, läuft dieser hoch und befindet sich mit seinen Defaultparametern am Bus.</p> <p><b>Defaultparameter: Adresse: 1, Übertragungsrate: 1,5MBAud</b></p>

**Projektierung über CPU nach Netz-Ein**

Sollte eine Hardwareprojektierung in der CPU vorliegen, so wird diese automatisch nach einem Netz-Ein über den Rückwandbus in das RAM des DP-Master übertragen.

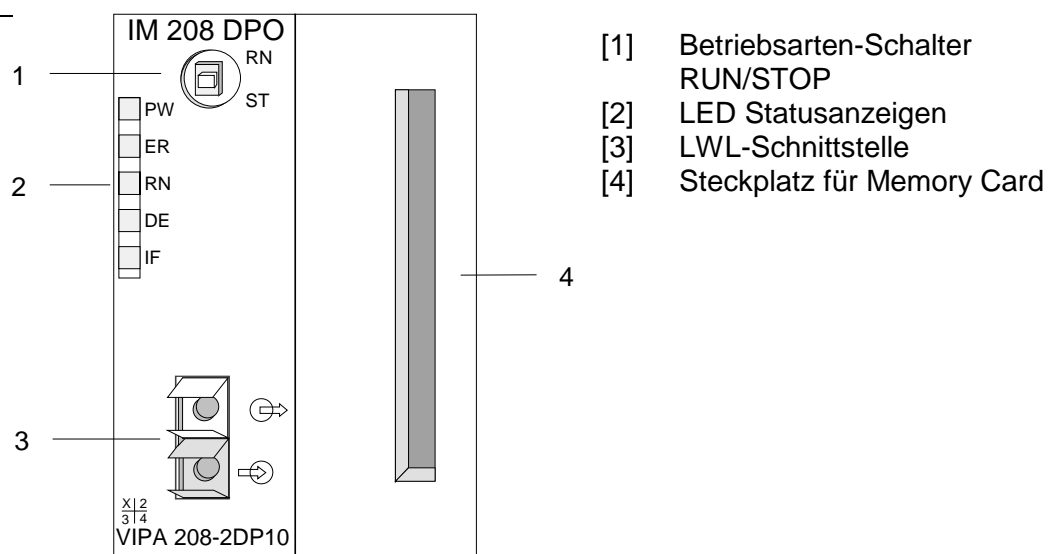


## IM 208DPO - Master mit LWL - Aufbau

### Eigenschaften

- Profibus-DP-Master der Klasse 1
- Max. 16 DPO-Slaves bei 1,5Mbaud an einen DPO Master angeschlossen
- Blendet Datenbereich der Slaves über den V-Bus im Adressbereich der CPU ein
- Projektierung unter WinNCS von VIPA bzw. ComProfibus von Siemens
- Diagnosefähig

### Frontansicht IM 208DPO

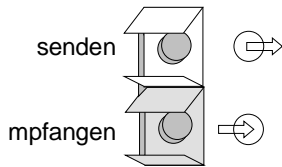


### Komponenten

#### LEDs

Die Master-Module besitzen verschiedene LEDs, die der Busdiagnose dienen und den eigenen Betriebszustand anzeigen. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
PW	Gelb	Signalisiert eine anliegende Betriebsspannung am Rückwandbus.
ER	Rot	Leuchtet bei Slave-Ausfall (ERROR).
RN	Grün	Leuchtet nur RN, befindet sich der Master im RUN. Die Slaves werden angesprochen und die Ausgänge sind 0 ("clear"-Zustand). Leuchten RN + DE, befindet sich der Master im "operate"-Zustand. Er tauscht Daten mit den Slaves aus.
DE	Gelb	DE (DataExchange) zeigt Kommunikation über Profibus an.
IF	Rot	Initialisierungsfehler für fehlerhafte Parametrierung

**LWL-Schnittstelle**

Über diese Buchse binden Sie den Profibus-Koppler über Lichtwellenleiter in Ihren Profibus ein. Die Anschlussbelegung dieser Schnittstelle zeigt die Abbildung links.

**Spannungsversorgung**

Der Profibus-Master bezieht seine Versorgungsspannung über den Rückwandbus.

**Betriebsart-Schalter**

Mit dem Betriebsart-Schalter können Sie zwischen den Betriebsarten STOP (ST) und RUN (RN) wählen.

Bei Betriebsartschalter auf RN und gültigen Parametern, geht der Master in den RUN-Zustand über.

Wird der Betriebsart-Schalter auf ST gestellt, geht der Master in den STOP-Zustand über. Er beendet die Kommunikation, worauf alle Ausgänge auf 0 gelegt werden und sendet einen Alarm an das übergeordnete System.

Eine ausführliche Erklärung finden Sie in diesem Kapitel unter "Betriebszustände".

**Flash Memory Card**

Zur Übertragung Ihrer Projektierung können Sie eine Flash Memory Card (Speicherkarte) verwenden, die Sie hier stecken können.

Die Memory Card erhalten Sie bei VIPA unter Best.-Nr.: VIPA 374-1KH21.

Bei Einsatz eines PGs mit Memory Card Steckplatz können Sie Ihr Projekt direkt auf die Speicherkarte schreiben.

Sollten Sie für die Projektierung einen PC einsetzen, können Sie ein EPROM-Programmiergerät verwenden, das sie bei VIPA, unter der Best.-Nr.: VIPA Multi-Prommer, erhalten können.

*Einsatz im IM 208DPO*

Sobald der DP-Master bei gesteckter Memory Card mit Spannung versorgt wird bzw. der Betriebsart-Schalter von ST auf RN geschaltet wird, werden die aktuellen Projektierdaten und die Bus-Parameter von der Memory Card in das interne RAM des DP-Master übertragen.

Sobald sich der DP-Master im Data Exchange befindet (RN und DE leuchten), können Sie die Memory Card entnehmen.

Die Memory Card dürfen Sie jederzeit entnehmen. Bitte beachten Sie aber, dass bei einem STOP-RUN-Übergang der DP-Master neu bootet und hierbei auf die Memory Card zugreift.

**Betriebszustände,  
Anlaufverhalten***Power On*

Der IM 208DPO wird mit Spannung versorgt. Die Projektierdaten werden von der Memory Card gelesen, auf Gültigkeit geprüft und im internen RAM des DP-Master abgelegt.

Bei Betriebsartschalter auf RUN und gültigen Parametern, geht der Master selbständig in den RUN-Zustand über. Im Run leuchten die LEDs RN, DE und ER. Sobald alle projektierten Slaves im DataExchange sind, erlischt auch die ER-LED.

*STOP*

Im STOP-Zustand und gültige Parameter vorausgesetzt, sind die Ausgänge der zugeordneten Slaves auf 0 gesetzt. Es findet zwar keine Kommunikation statt, aber der Master ist mit seinen aktuellen Busparametern aktiv am Bus und belegt die ihm zugeteilte Adresse am Bus. Zur Freigabe der Adresse ist am DP-Master der Lichtwellenleiter abzuziehen.

*STOP → RUN*

In Stellung RN bootet der Master erneut: Die Projektierdaten und Bus-Parameter werden von der Memory Card gelesen und im internen RAM des DP-Master abgelegt.

Nun wird die Kommunikation zu den Slaves aufgebaut. Während dieser Zeit brennt nur die RN-LED. Bei erfolgter Kommunikation gültiger Bus-Parameter, geht der DP-Master in RUN. Der DP-Master zeigt dies über die LEDs RN und DE an. Solange der DP-Master im RUN bleibt, können sie die Memory Card entnehmen.

Bei fehlerhaften Parametern oder wenn keine Memory Card gesteckt ist bleibt der DP-Master im STOP-Zustand und zeigt einen Parametrierfehler mit der IF-LED an. Der DP-Master ist dann mit folgenden Default-Bus-Parameter aktiv am Bus:

**Default-Bus-Parameter: Adresse: 1, Übertragungsrate: 1,5MBaud.**

*RUN*

Im RUN leuchten die RN- und DE-LEDs. Jetzt können Daten ausgetauscht werden. Im Fehlerfall wie z.B. Slave-Ausfall, wird dies am DP-Master über die ER-LED angezeigt und ein Alarm an das übergeordnete System abgesetzt.

*RUN → STOP*

Der DP-Master geht in den STOP-Zustand über. Er beendet die Kommunikation worauf alle Ausgänge auf 0 gelegt werden und sendet einen Alarm an das übergeordnete System.

## IM 208DPO - Master mit LWL - Einsatz an einer CPU 21x

**Kommunikation** Über die IM 208 DPO Master-Module können bis zu 16 Profibus-DP-Slaves an eine System 200V CPU angekoppelt werden. Der Master kommuniziert mit den Slaves und blendet die Datenbereiche über den Rückwandbus im Adressbereich der CPU ein. Es dürfen maximal 1024 Byte Eingangs- und 1024 Byte Ausgangsdaten entstehen.

Bei Firmwareständen < V3.0.0 stehen nur jeweils 256 Byte für Ein- und Ausgabedaten zur Verfügung.

Bei jedem Neustart der CPU holt sich diese von allen Mastern die I/O-Mapping-Daten.

**Alarmbearbeitung** Die Alarmbearbeitung ist aktiviert, d.h. eine IM 208-Fehlermeldung kann folgende Alarme auslösen, die die CPU veranlassen die entsprechenden OBs aufzurufen:

- Prozessalarm: OB40
- Diagnosealarm: OB82
- Slaveausfall: OB86

Sobald das BASP-Signal (**B**efehls**a**usgab**e**s**p**er**r**e) von der CPU kommt, stellt der DP-Master die Ausgänge der angeschlossenen Peripherie auf Null.



### Hinweis!

Das Prozessabbild der Eingänge behält nach einem Slaveausfall den Zustand wie vor dem Slaveausfall.

**Voraussetzungen** Bitte beachten Sie bei Einsatz des IM 208 Profibus-DP-Master, dass dieser einen Firmwarestand ab V3.0.0 besitzt; ansonsten kann dieser an einer CPU 21x mit Firmwarestand ab V3.0.0 nicht betrieben werden.

Den jeweiligen Firmwarestand finden Sie auf einem Aufkleber auf der Rückseite des Moduls.

Bei Fragen zum Firmware-Update wenden Sie sich bitte an den VIPA Support (support@vipa.de).

Nähere Angaben zur Einbindung in Ihre CPU finden Sie in der Dokumentation zu Ihrer CPU.



### Hinweis!

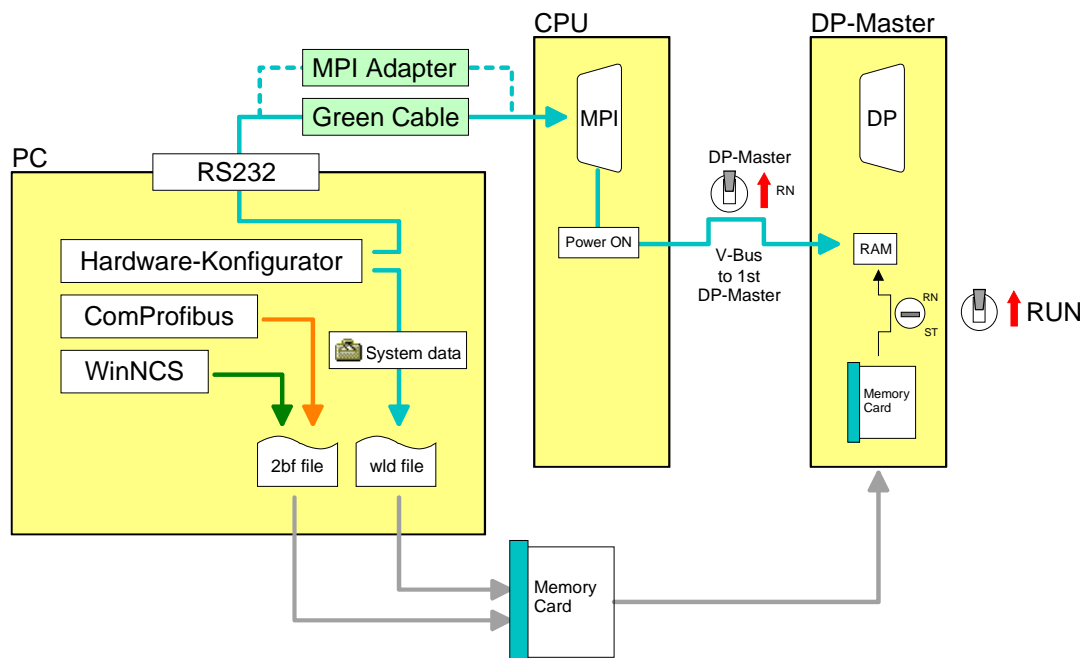
Für den Einsatz an einer CPU 24x ist die Firmware des DP-Masters auf Stand V1.0.7 zu bringen.

## IM 208DPO - Master mit LWL - Projektierung

### Übersicht

Zur Projektierung des DP-Masters haben Sie folgende Möglichkeiten:

- Projektierung im Hardware-Konfigurator von Siemens und Transfer als Hardwarekonfiguration über den Rückwandbus. Hierüber können Sie ausschließlich den 1. Master (IM 208 bzw. CPU 21xDPM) projektieren.  
**Bitte beachten Sie, dass sich der Betriebsart-Schalter des DP-Master für die Projektierung über den Systembus in RN befinden muss.**
- Projektierung im Hardware-Konfigurator von Siemens und Export einer wld-Datei auf Memory Card.
- Projektierung über WinNCS von VIPA bzw. ComProfibus von Siemens und Export einer 2bf-Datei auf Memory Card.



### Erforderliche Firmwarestände

DP-Master und CPU sollen einen Firmwarestand ab V3.0.0 besitzen, da ansonsten der DP-Master an der CPU 21x nicht betrieben werden kann. Den jeweiligen Firmwarestand finden Sie auf einem Aufkleber auf der Rückseite des Moduls.

Firmwareversion

DP-Master	CPU	Eigenschaft
V3.0.0	V3.0.0	1024Byte Ein- und Ausgangsdaten
V3.0.4	V3.0.0	Projektierung über wld-Datei
V3.0.6	V3.3.0	Projektierung als Hardwarekonfiguration über MPI

### Projektierung über Hardware-konfiguration

Sie projektieren im Hardware-Konfigurator von Siemens Ihr SPS-System zusammen mit dem DP-Master. Diese "Hardwarekonfiguration" übertragen Sie via MPI in die CPU. Bei Power ON werden die Projektierdaten in den DP-Master übertragen sofern sich dessen Betriebsart-Schalter in Stellung RN befindet.



#### Hinweis!

Bitte beachten Sie, dass sich der Betriebsart-Schalter des DP-Masters in Stellung RN befinden muss, da ansonsten beim STOP-RUN-Übergang der Master neu bootet und die Projektierung gelöscht wird.

Weiter ist zu beachten, dass bei hierbei immer nur der 1. DP-Master im System projektiert werden kann! Zur Projektierung weiterer DP-Master ist das jeweilige Projekt als wld-Datei bzw. 2bf-Datei zu exportieren.

### Vorgehensweise zur Hardware-konfiguration

Die Schritte zur Projektierung im Hardware-Konfigurator sollen hier kurz aufgezeigt werden:

1. Legen Sie ein neues Projekt System 300 an und fügen Sie aus dem Hardwarekatalog eine Profilschiene ein.
2. Fügen Sie die CPU 315-2DP ein. Sie finden die CPU mit Profibus-Master im Hardwarekatalog unter:  
*Simatic300 > CPU-300 > CPU315-2DP > 6ES7 315-2AF01-0AB0*
3. Geben Sie Ihrem Master eine Profibus-Adresse von 2 aufsteigend.
4. Klicken Sie auf DP und stellen Sie unter *Objekteigenschaften* die Betriebsart "DP-Master" ein. Bestätigen Sie Ihre Eingabe mit OK.
5. Durch Klick mit der rechten Maustaste auf "DP" öffnet sich das Kontextmenü. Wählen Sie "Mastersystem einfügen" aus. Legen Sie über NEU ein neues Profibus-Subnetz an.

Die nachfolgende Abbildung zeigt das erzeugte Mastersystem:

The screenshot shows the SIMATIC 300 hardware configuration interface. The main window displays a rack configuration with a CPU 315-2 DP in slot 2. The hardware catalog on the right shows the selected component: 6ES7 315-2AF01-0AB0. The bottom table shows the configuration details for the DP-Master system.

Steckplatz	Baugruppe	Bestellnummer	Fl.	M.	E.	A.	K.
1							
2	CPU 315-2 DP	6ES7 315-2AF01-0AB0	2				
3	DP				1023		
4							
5							
6							
7							
8							
9							
10							
11							

6. Um zum STEP<sup>®</sup>7 Projektiertool von Siemens kompatibel zu sein, ist die System 200V CPU explizit einzubinden.  
 Hängen Sie hierzu an das Subnetz das System "VIPA\_DP200V\_2". Sie finden dies im Hardware-Katalog unter:  
*PROFIBUS DP > Weitere Feldgeräte > IO > VIPA\_System\_200V > VIPA\_DP200V\_2.*  
 Geben Sie diesem Slave die Profibus-Adresse 1.  
 Platzieren Sie auf dem Steckplatz 0 die entsprechende CPU 21x von VIPA, indem Sie diese im Hardware-Katalog unter *VIPA\_CPU21x* auswählen.  
**Der Steckplatz 0 ist zwingend erforderlich!**
7. Zur Einbindung der am VIPA-Bus befindlichen Module ziehen Sie aus dem Hardware-Katalog unter *VIPA\_DP200V\_2* die entsprechenden System 200V Module auf die Steckplätze unterhalb der CPU. Beginnen Sie mit Steckplatz 1. Auf diese Weise platzieren Sie auch Ihren DP-Master.
8. Zur Projektierung von DP-Slaves, die an den DP-Master angekoppelt sind, verfahren Sie auf die gleiche Weise wie unter 6. (Projektierung des *VIPA\_DP200V\_2*-Systems).  
 Entnehmen Sie aus dem Hardware-Katalog das entsprechende Profibus-System und ziehen Sie dies auf das DP-Master Subnetz.  
 Vergeben Sie dem Slave eine Adresse > 3.  
**CPU 21x zentral**
- 
- | Steckplatz | Baugruppe                               |
|------------|---|
| 0          | 21x-xxxx                                |
| 1          | zentrale<br>Peripherie<br>mit DP-Master |
| .          |   |
| .          |   |
| 32         |   |
9. Klicken Sie auf (speichern und übersetzen).

**Transfervarianten**

Abhängig von der eingesetzten Firmware auf CPU und DP-Master haben Sie nun folgende Transfermöglichkeiten:

1. Transfer via MPI (nur für 1. Master am Bus)  
Ihre DP-Master-Projektierung wird zusammen mit Ihrem SPS-Programm an die CPU übertragen. Die CPU reicht die DP-Master-Projektierung automatisch weiter an den 1. DP-Master (IM 208DP oder CPU 21xDPM), der sich am Rückwandbus befindet.
2. Export des Projekts als wld-Datei auf Memory Card  
Exportieren Sie Ihre Projektierung als wld-Datei und übertragen Sie diese auf eine Memory Card (Speicherkarte). Die Memory Card stecken Sie in den entsprechenden DP-Master.  
Zur Übertragung Ihrer Projektierung in Ihren System 200V Profibus-Master ist eine Memory Card erforderlich und ein externes EPROM Programmiergerät mit Software. Die Memory Card erhalten Sie bei VIPA unter Best.-Nr.: VIPA 374-1KH21.

**zu 1.  
Transfer via MPI**

10.	Verbinden Sie Ihr PG bzw. Ihren PC über MPI mit Ihrer CPU. Zur seriellen Punkt-zu-Punkt-Übertragung von Ihrem PC an die CPU können Sie auch das Green Cable von VIPA verwenden.  Das Green Cable hat die Best.-Nr. VIPA 950-0KB00 und darf ausschließlich auf CPUs der VIPA eingesetzt werden. Bitte beachten Sie hierzu die Hinweise zum Green Cable in den Grundlagen!  Bei Einsatz des Green Cable von VIPA ist die MPI-Schnittstelle entsprechend zu konfigurieren (PC Adapter MPI, 38400Baud).
11.	Schalten Sie die Spannungsversorgung Ihrer CPU ein.
12.	Übertragen vom Hardware-Konfigurator von Siemens über den Menüpunkt <b>Zielsystem</b> > <i>Laden in Baugruppe</i> Ihr Projekt in die CPU.
13.	Zur zusätzlichen Sicherung Ihres Projekts auf MMC stecken Sie eine MMC in die CPU 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.

Die Projektierung ist hiermit abgeschlossen.



**Hinweis!**

Bitte beachten Sie, dass durch die oben aufgeführte Methode immer der 1. DP-Master am Bus projektiert werden kann. Zur Projektierung weiterer DP-Master ist ein Einsatz einer Memory Card mit einer exportierten wld-Datei erforderlich.



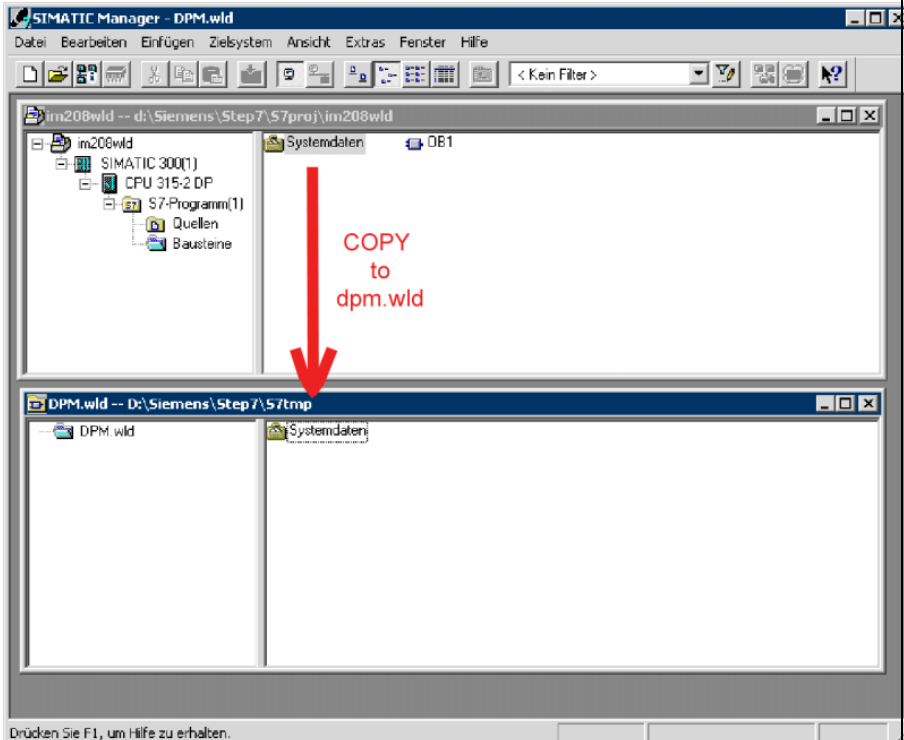
**zu 2.  
Export als dpm.wld  
auf Memory Card**

Zur Projektierung weiterer DP-Master exportieren Sie Ihr Projekt in Form einer wld-Datei auf eine Memory Card. Die Memory Card stecken sie in den entsprechenden DP-Master, der während des Hochlaufs die Projektierung von der Memory Card in sein RAM übernimmt. Solange der DP-Master mit Spannung versorgt wird können Sie die Memory Card entnehmen.

Auf diese Weise können Sie mehrere DP-Master projektieren, die sich am gleichen Rückwandbus befinden.

*Vorgehensweise*

10..	<p>Legen Sie mit <b>Datei &gt; Memory Card &gt; Neu...</b> eine neue wld-Datei an. Damit diese Datei vom Profibus-Master übernommen wird, muss diese den Namen <b>dpm.wld</b> besitzen.</p> <p>→ Die Datei wird zusätzlich zum Projektfenster eingeblendet.</p>
11.	<p>Gehen Sie nun in Ihr Projekt in das Verzeichnis <i>Bausteine</i> und kopieren Sie das Verzeichnis "Systemdaten" in die neu angelegte dpm.wld-Datei.</p>




**Hinweis!**

Soll ein schon bestehendes "Systemdaten"-Verzeichnis überschrieben werden, so ist zuvor das zu überschreibende Verzeichnis "Systemdaten" zu löschen.






Befindet sich eine wld- und eine 2bf-Datei auf der MMC, so hat die wld-Datei Vorrang.

Die Projektierung ist hiermit abgeschlossen.

## Projektierung unter WinNCS

Die Projektierung können Sie auch unter WinNCS von VIPA durchführen und Ihr Projekt als 2bf-Datei auf eine Memory Card exportieren.

Die Schritte für die Projektierung unter WinNCS sollen hier kurz aufgezeigt werden. Näheres zum Einsatz von WinNCS finden Sie auch im zugehörigen Handbuch HB91.

1.	Starten Sie WinNCS und legen Sie mit <b>Datei</b> > <i>anlegen/öffnen</i> eine neue Projektdatei unter der Funktionalität "Profibus" an.
2.	Fügen Sie wenn noch nicht geschehen im Netzwerkfenster mit  eine <b>Profibus-Funktionsgruppe</b> ein und klicken Sie im Parameterfenster auf [übernehmen].
3.	Fügen Sie im Netzwerkfenster mit  einen <b>Profibus-Host/Master</b> ein und geben Sie im Parameterfenster die Profibusadresse Ihres Masters an.
4.	Fügen Sie im Netzwerkfenster mit  einen <b>Profibus-Slave</b> ein. Geben Sie im Parameterfenster die Profibusadresse, die Familie "I/O" und den Stationstyp "VIPA_DP200V_2" ein und klicken Sie auf [übernehmen].
5.	<p>Projektieren Sie über  der Reihe nach alle Peripherie-Module, die über den Rückwandbus mit dem entsprechenden Profibus-Slave verbunden sind. Über [Auto] können Sie die Peripherie automatisch adressieren lassen und die Adressbelegung über [MAP] anzeigen.</p> <p><b>Bitte beachten Sie bitte, dass es bei der automatischen Adressierung nicht zu Adresskonflikten mit der lokalen Peripherie kommt!</b></p> <p>Handelt es sich um ein intelligentes Modul, wie z.B. CP240, erscheinen die hierzu einstellbaren Parameter.</p>
6.	Nachdem Sie alle Slaves mit zugehöriger Peripherie projiziert haben, müssen die Busparameter für den Profibus berechnet werden. Aktivieren Sie hierzu im Netzwerkfenster die Funktionsgruppe Profibus. Klicken Sie im Parameterfenster auf das Register "Busparameter". Stellen Sie die gewünschte Baudrate ein und klicken Sie auf [calculate]. Die Busparameter werden berechnet - [übernehmen] sie diese. Bei jeder Änderung der Modulzusammenstellung sind die Busparameter neu zu berechnen!
7.	<p>Aktivieren Sie im Netzwerkfenster die Master-Ebene und exportieren Sie Ihr Projekt in die Datei dpm.2bf. Übertragen Sie mit einem Programmiergerät die 2bf-Datei auf die Memory Card.</p> <p>WinNCS unterstützt den Datentransfer über eine Master-PC-Steckkarte von der Firma Softing. Mit dieser Karte können Sie über den Profibus eine Master-Master-Kommunikation aufbauen und Ihre 2bf-Datei mit den Modultransfer-Funktionen  in beide Richtungen transferieren.</p>



### Hinweis!

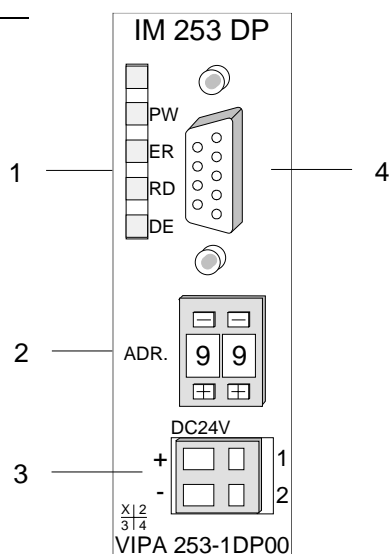
Da sich der IM 208 DP-Master gleich verhält wie die IM 308-C von Siemens, können Sie diesen auch als IM 308-C unter "ComProfibus" von Siemens projektieren und ihr Projekt als 2bf-Datei exportieren.

## IM 253DP - Slave (Standard) - Aufbau

### Eigenschaften

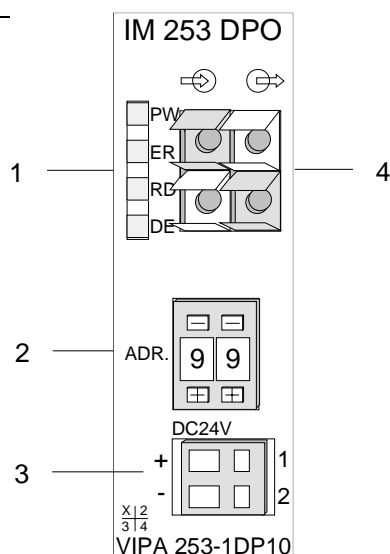
- Profibus-DP Slave für max. 32 Peripherie-Module (max. 16 Analog-Module)
- Max. 152Byte Eingabe- und 152Byte Ausgabe-Daten
- Internes Diagnoseprotokoll mit Zeitstempel
- Integriertes DC 24V-Netzteil zur Versorgung der Peripherie-Module (max. 3,5A)
- Unterstützung aller Profibus-Datenraten

### Frontansicht 253-1DP00



- [1] LED Statusanzeigen
- [2] Adress-Schalter
- [3] Anschluss für DC 24V-Spannungsversorgung
- [4] RS485-Schnittstelle

### Frontansicht 253-1DP10



- [1] LED Statusanzeigen
- [2] Adress-Schalter
- [3] Anschluss für DC 24V-Spannungsversorgung
- [4] LWL-Schnittstelle

**Komponenten**

**LEDs**

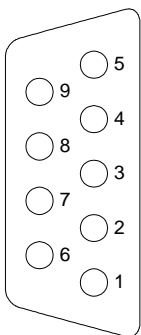
Die Profibus-Slave-Module besitzen verschiedene LEDs, die unter anderem auch der Busdiagnose dienen. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
PW	Gelb	Signalisiert eine anliegende Betriebsspannung (Power).
ER	Rot	Leuchtet bei Neustart kurz auf. Leuchtet bei internem Fehler. Blinkt bei Initialisierungsfehler.
RD	Grün	Blinkt abwechselnd mit RD bei fehlerhafter Konfiguration vom Master (Projektierungsfehler). Blinkt gleichzeitig mit RD bei fehlerhafter Parametrierung Leuchtet im Zustand "DataExchange" wenn der V-Bus-Zyklus schneller als der Profibus-Zyklus läuft. Ist ausgeschaltet im Zustand "DataExchange" wenn der V-Bus-Zyklus langsamer als der Profibus-Zyklus läuft. Blinkt bei positivem Selbsttest (READY) und erfolgreicher Initialisierung.
DE	Gelb	Blinkt abwechselnd mit ER bei fehlerhafter Konfiguration vom Master (Projektierungsfehler). Blinkt gleichzeitig mit ER bei fehlerhafter Parametrierung DE (DataExchange) zeigt an, dass eine Kommunikation mit dem Profibus stattfindet.

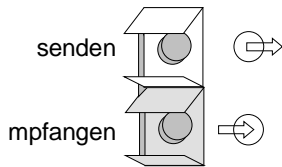
**RS485-Schnittstelle**

Über eine 9-polige RS485-Schnittstelle binden Sie Ihren Profibus-Slave in Ihren Profibus ein.

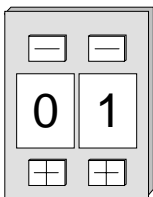
Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:



Pin	Belegung
1	Schirm
2	n.c.
3	RxD/TxD-P
4	CNTR-P
5	GND
6	5V (max. 70mA)
7	n.c.
8	RxD/TxD-N
9	n.c.

**LWL-Schnittstelle**

Über diese Buchse binden Sie den Profibus-Koppler über Lichtwellenleiter in den Profibus ein. Die Anschlussbelegung dieser Schnittstelle zeigt die Abbildung links.

**Adress-Schalter**

Mit dem Adress-Schalter können Sie für den DP-Slave die Profibus-Adresse einstellen. Erlaubte Adressen sind 1 bis 99. Jede Adresse darf nur einmal am Bus vergeben sein.

Die Slave-Adresse muss vor dem Einschalten des Buskopplers eingestellt werden.

Sobald Sie während des Betriebs die Adresse 00 einstellen, werden einmalig die Diagnosedaten im Flash-ROM gesichert. Bitte vergessen Sie nicht, die ursprüngliche Profibusadresse wieder einzustellen, damit beim nächsten PowerOn die richtige Profibusadresse verwendet wird.

**Spannungsversorgung**

Jeder Profibus-Slave besitzt ein eingebautes Netzteil. Das Netzteil ist mit DC 24V zu versorgen. Über die Versorgungsspannung werden neben der Buskopplerelektronik auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus mit maximal 3,5A versorgen kann.

Das Netzteil ist gegen Verpolung und Überstrom geschützt.

Profibus und Rückwandbus sind galvanisch voneinander getrennt.

**Achtung!**

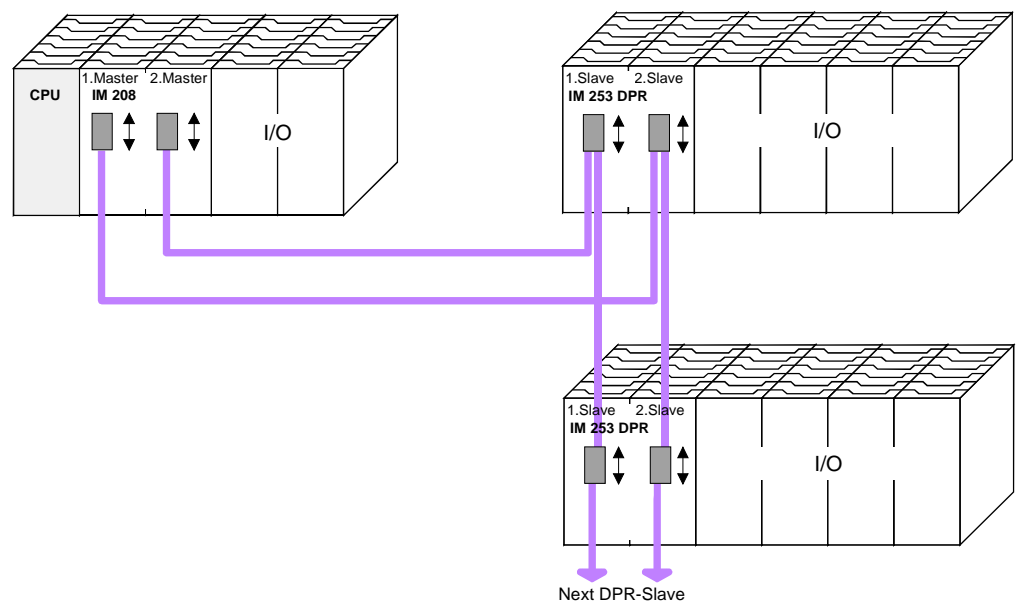
Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung!

## IM 253 DPR - Slave (redundant) - Aufbau

### Redundantes System

Prinzipiell besteht das IM 253DPR aus 2 Profibus-DP-Slave-Anschaltungen. Die beiden Profibus-Slaves überwachen gegenseitig ihre Betriebszustände. Beide Slaves befinden sich mit der gleichen Adresse am Profibus und kommunizieren mit einem redundanten DP-Master.

Beide Slaves lesen die Peripherie-Eingänge. Es kann immer nur ein Slave auf die Peripherie-Ausgänge zugreifen. Der andere Slave ist passiv und steht in Bereitschaft. Sobald der aktive Slave ausfällt, werden die Peripherie-Ausgänge vom passiven Slave angesteuert.



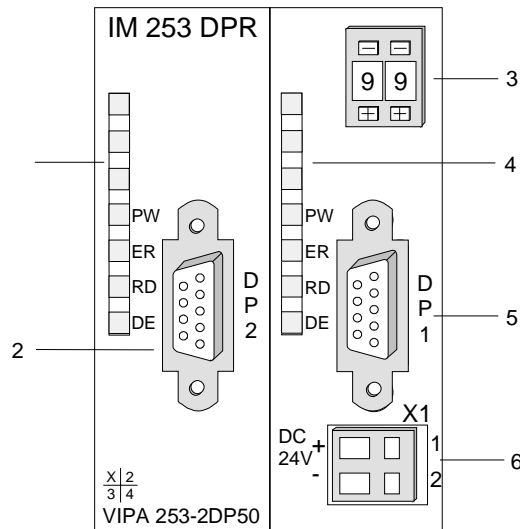
### Voraussetzungen für den Einsatz

Bitte beachten Sie, dass zum redundanten Einsatz des Moduls ein redundanter DP-Master zu verwenden ist. In jeder Master-Einheit müssen Projektierung und Buskonfiguration gleich sein.

### Eigenschaften IM 253 DPR

- 2 redundante Kanäle
- DPR-Slave für max. 32 Peripherie-Module (max. 16 Analog-Module)
- Max. 152Byte Eingabe- und 152Byte Ausgabe-Daten
- Internes Diagnoseprotokoll mit Zeitstempel
- Integrierte DC 24V-Spannungsversorgung für Peripherie-Module (max. 3,5A)
- Unterstützung aller Profibus-Datenraten

**Frontansicht  
253-2DP50**



- [1] LED Status DP2
- [2] RS485-Schnittstelle DP2
- [3] Adress-Schalter
- [4] LED Status DP1
- [5] RS485-Schnittstelle DP1
- [6] Anschluss für DC24V-Spannungsversorgung

**Komponenten**

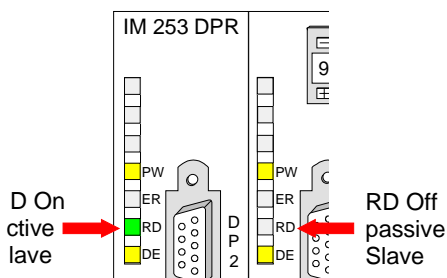
**LEDs**

Der redundante Slave besitzt je Slave-Einheit eine LED-Reihe, die unter anderem auch der Busdiagnose dienen. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bez.	Farbe	Bedeutung
PW	Gelb	Signalisiert eine anliegende Betriebsspannung (Power).
ER	Rot	Leuchtet bei Neustart kurz auf. Leuchtet bei internem Fehler. Blinkt bei Initialisierungsfehler. Blinkt abwechselnd mit RD bei fehlerhafter Konfiguration vom Master (Projektierungsfehler). Blinkt gleichzeitig mit RD bei fehlerhafter Parametrierung
RD	Grün	Blinkt bei positivem Selbsttest (READY) und erfolgreicher Initialisierung.
DE	Gelb	DE (DataExchange) zeigt Kommunikation mit dem Profibus an.

**LEDs bei  
redundantem  
Betrieb**

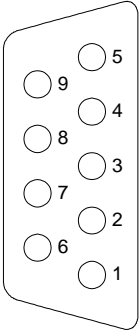
Im redundanten Betrieb zeigt der aktive Slave seine Aktivität über die grüne RD-LED an. Beim passiven Slave ist die RD-LED ausgeschaltet. Bei beiden Slaves leuchtet die PW- und DE-LED.



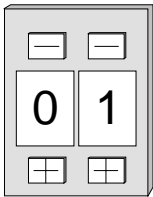
RD	DE	Bedeutung
ein	ein	aktiver Slave (schreiben und lesen)
aus	ein	passiver Backup-Slave (lesen)

**RS485-Schnittstelle**

Über zwei 9polige RS485-Schnittstellen binden Sie die 2 Kanäle in den Profibus ein. Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:



Pin	Belegung
1	Schirm
2	n.c.
3	RxD/TxD-P
4	CNTR-P
5	GND
6	5V (max. 70mA)
7	n.c.
8	RxD/TxD-N
9	n.c.

**Adress-Schalter**

Mit dem Adress-Schalter können Sie für beide DP-Slaves die Profibus-Adresse einstellen. Erlaubte Adressen sind 1 bis 99. Jede Adresse darf nur einmal am Bus vergeben sein.

Die Slave-Adresse muss vor dem Einschalten des Buskopplers eingestellt werden.

Sobald Sie während des Betriebs die Adresse 00 einstellen, werden einmalig die Diagnosedaten im Flash-ROM gesichert. Bitte vergessen Sie nicht, die ursprüngliche Profibusadresse wieder einzustellen, damit beim nächsten PowerOn die richtige Profibusadresse verwendet wird.

**Spannungsversorgung**

Der Profibus-Slave besitzt ein eingebautes Netzteil. Das Netzteil ist mit DC 24V zu versorgen. Über die Versorgungsspannung werden neben der Buskopplerelektronik auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus mit maximal 3,5A versorgen kann.

Das Netzteil ist gegen Verpolung und Überstrom geschützt.

Profibus und Rückwandbus sind galvanisch voneinander getrennt.

**Achtung!**

Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung!



## IM 253DP, DO 24xDC 24V - Aufbau

### Allgemeines

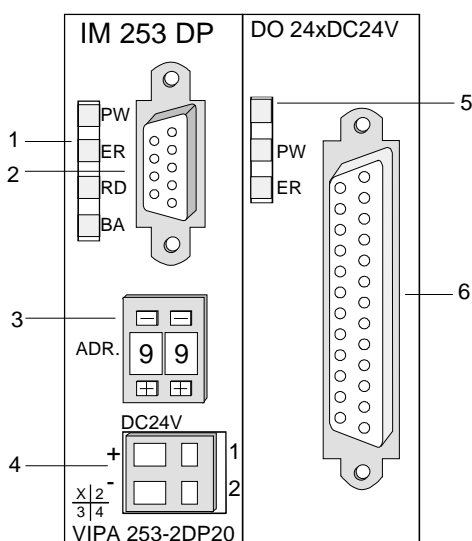
Das Modul besteht aus einem Profibus-Slave mit integrierter 24fach Ausgabe-Einheit. Direkt über den Profibus werden die 24 Ausgabekanäle angesteuert. Ein Ausgabekanal kann mit maximal 1A belastet werden. Hierbei ist zu beachten, dass ein Summenstrom von 4A nicht überschritten wird. Die Ausgänge sind potentialgebunden.

### Eigenschaften

Das Profibus-Ausgabe-Modul IM 253DP, DO 24xDC 24V zeichnet sich durch folgende Eigenschaften aus:

- Profibus-Slave
- 24 digitale Ausgänge
- potentialgebunden
- Ausgangs-Nennspannung DC 24V, max. 1A / Kanal
- Summenausgangsstrom max. 4A
- LED für Fehlermeldung bei Überlast, Überhitzung oder Kurzschluss
- Geeignet für Kleinmotoren, Lampen, Magnetschalter und Schütze, die über Profibus anzusteuern sind.

### Frontansicht IM 253DP, DO 24xDC 24V



- [1] LEDs Statusanzeige Profibus
- [2] Profibus-Buchse
- [3] Adresseinsteller
- [4] Anschluss für DC 24V-Spannungsversorgung
- [5] LEDs Statusanzeige Ausgabe-Einheit
- [6] 25polige Buchse für Digital-Ausgabe



### Achtung!

Die beiden Modulhälften müssen beim Standalone-Einsatz über den mitgelieferten 1fach-Busverbinder miteinander verbunden werden!

**Komponenten**

Die Komponenten des Profibus-Teils sind identisch mit den Komponenten der weiter oben beschriebenen Profibus-Slave-Module.

**LEDs Profibus**

Der Profibus-Teil besitzt verschiedene LEDs, die unter anderem auch der Busdiagnose dienen.

Bezeichnung	Farbe	Bedeutung
PW	Gelb	Signalisiert eine anliegende Betriebsspannung (Power).
ER	Rot	Leuchtet bei Neustart kurz auf. Leuchtet bei internem Fehler. Blinkt bei Initialisierungsfehler. Blinkt abwechselnd mit RD bei fehlerhafter Konfiguration vom Master (Projektierungsfehler). Blinkt gleichzeitig mit RD bei fehlerhafter Parametrierung
RD	Grün	Leuchtet im Zustand "DataExchange" wenn der V-Bus-Zyklus schneller als der Profibus-Zyklus läuft. Ist ausgeschaltet im Zustand "DataExchange" wenn der V-Bus-Zyklus langsamer als der Profibus-Zyklus läuft. Blinkt bei positivem Selbsttest (READY) und erfolgreicher Initialisierung. Blinkt abwechselnd mit ER bei fehlerhafter Konfiguration vom Master (Projektierungsfehler). Blinkt gleichzeitig mit ER bei fehlerhafter Parametrierung
DE	Gelb	DE (DataExchange) zeigt an, dass eine Kommunikation mit dem Profibus stattfindet.

**LEDs Digital-Ausgabe-Teil**

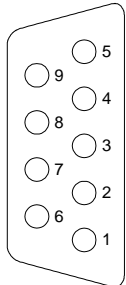
Auf dem digitalen Ausgabe-Teil befinden sich 2 LEDs, die folgende Funktion haben:

Bezeichnung	Farbe	Bedeutung
PW	Gelb	Signalisiert eine anliegende Betriebsspannung über den Profibus-Teil (Power).
ER	Rot	Leuchtet bei Kurzschluss, Überlast und Überhitzung

**Profibus RS485-Schnittstelle**

Über eine 9polige RS485-Schnittstelle binden Sie den Profibus-Slave in den Profibus ein.

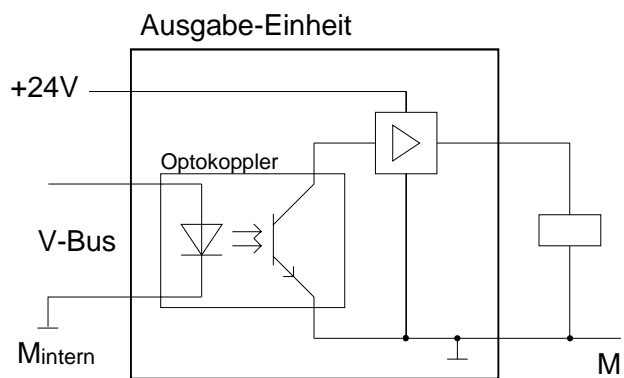
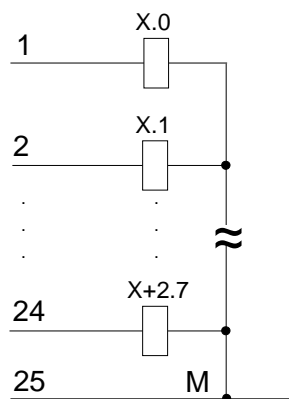
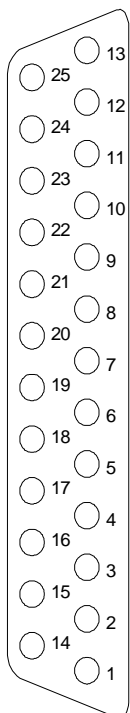
Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:

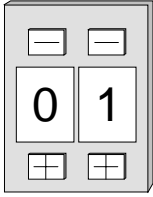


Pin	Belegung
1	Schirm
2	n.c.
3	RxD/TxD-P
4	CNTR-P
5	GND
6	5V (max. 70mA)
7	n.c.
8	RxD/TxD-N
9	n.c.

**Ausgabe-Einheit Anschluss- und Prinzipschaltbild**

Die DC 24V-Spannungsversorgung des Ausgabe-Teils erfolgt intern über die Spannungsversorgung des Slaves.



**Adress-Schalter**

Mit dem Adress-Schalter können Sie für den DP-Slave die Profibus-Adresse einstellen. Erlaubte Adressen sind 1 bis 99. Jede Adresse darf nur einmal am Bus vergeben sein.

Die Slave-Adresse muss vor dem Einschalten des Buskopplers eingestellt werden.

Sobald Sie während des Betriebs die Adresse 00 einstellen, werden einmalig die Diagnosedaten im Flash-ROM gesichert. Bitte vergessen Sie nicht, die ursprüngliche Profibusadresse wieder einzustellen, damit beim nächsten PowerOn die richtige Profibusadresse verwendet wird.

**Spannungsversorgung**

Jeder Profibus-Slave besitzt ein eingebautes Netzteil. Das Netzteil ist mit DC 24V zu versorgen. Über die Versorgungsspannung werden neben der Buskopplerelektronik auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus mit maximal 3,5A versorgen kann.

Das Netzteil ist gegen Verpolung und Überstrom geschützt.

Profibus und Rückwandbus sind galvanisch voneinander getrennt.

**Achtung!**

Sollte bei anliegender Spannung PW nicht leuchten, ist die interne Sicherung defekt!

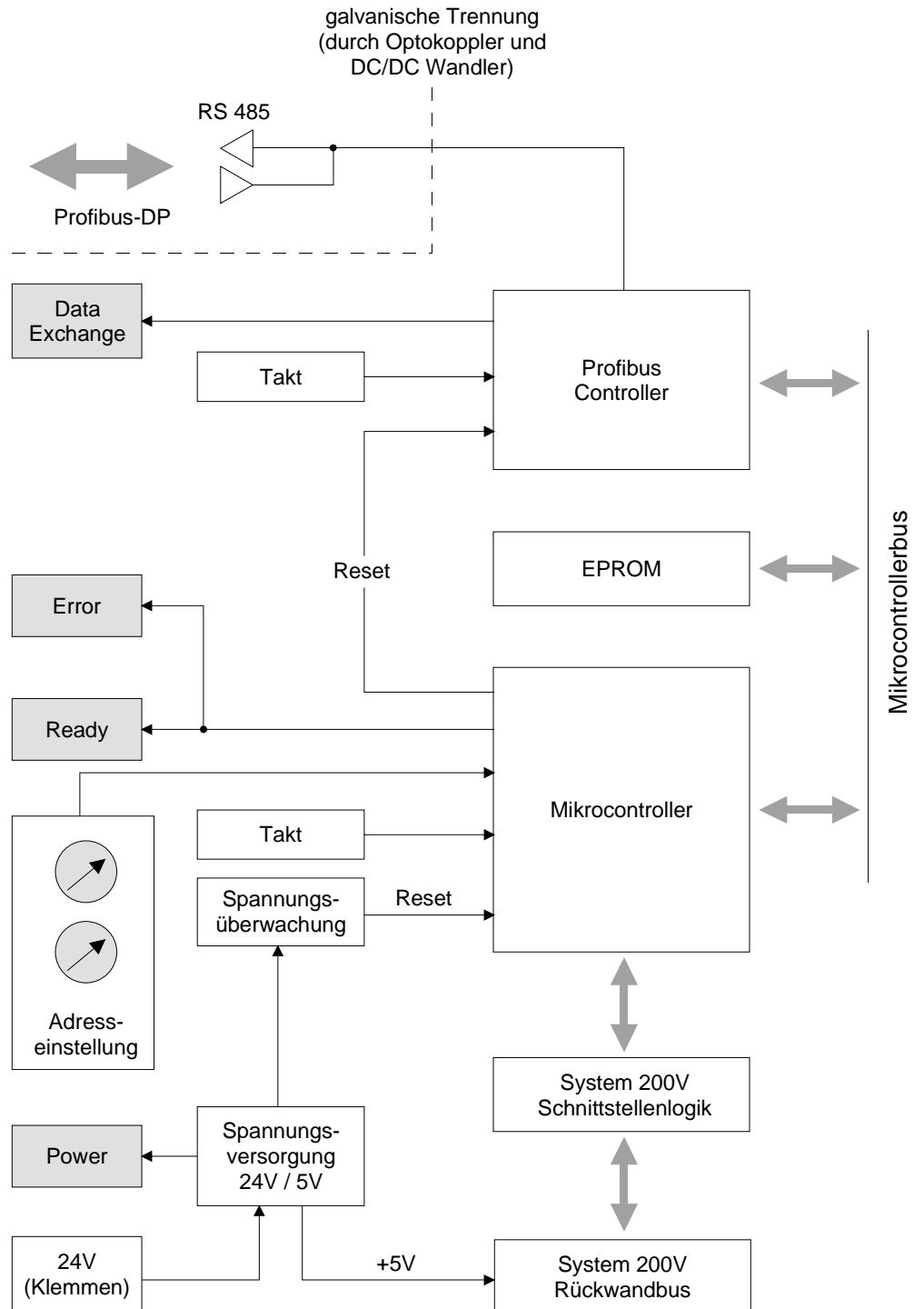
**Projektierung der Ausgänge**

Projektieren Sie den Slave wie weiter unten gezeigt; die Projektierung findet bei allen System 200V DP-Slaves von VIPA auf die gleiche Weise statt.

Zur Einbindung der 24 Ausgänge projektieren Sie zusätzlich das Modul VIPA 253-2DP20 auf dem ersten Steckplatz. Das Modul befindet sich hardwaretechnisch gesehen direkt neben dem Slave.

# IM 253DP - Slave - Blockschaltbild

Das nachfolgende Blockschaltbild zeigt den prinzipiellen Hardwareaufbau des Buskopplers und die Kommunikation, die intern stattfindet:



## IM 253DP - Slave - Projektierung

### Allgemeines

Die Parametrierung wird unter Ihrem Profibus DP-Master Projektiertool durchgeführt. Hierbei ordnen Sie Ihrem DP-Master die entsprechenden Profibus DP-Slave-Module zu.

Eine direkte Zuordnung erfolgt über die Profibus-Adresse, die Sie am DP-Slave einzustellen haben.

Bei der Hardwarekonfiguration werden die hier vorgestellten Slaves als "VIPA\_DP200V\_2" projiziert. Nach Installation der GSD finden Sie diesen beispielsweise im Hardware-Katalog von Siemens unter:

*Profibus-DP>Weitere Feldgeräte>I/O>VIPA\_System\_200V*

### GSD-Datei

Bei VIPA erhalten Sie jedes Profibus-Modul zusammen mit einer Diskette. Auf der Diskette befinden sich alle GSD- und Typdateien der Profibus-Module von VIPA.

Installieren Sie die entsprechenden Dateien von Ihrer Diskette in Ihrem Projektiertool. Nähere Hinweise zur Installation der GSD- bzw. Typdateien finden Sie im Handbuch zu Ihrem Projektiertool.

Bei Einsatz von WinNCS von VIPA sind alle GSD-Dateien bereits integriert!

### Einsatz IM 253DP, DO 24xDC 24V

Bei Einsatz von Profibus DP-Slave Kombi-Modulen wie z.B. dem VIPA 253-2DP20 wählen Sie als Modultyp "253-2DP20".

Das Modul muss immer auf Steckplatz 1 projiziert sein, da sich das Modul hardwaretechnisch gesehen direkt neben dem Slave befindet.



### Hinweis

Bei jeder Änderung der Modulzusammenstellung sind unter WinNCS von VIPA die Busparameter neu zu berechnen!

### Einsatz an einem IM 208 DP-Master von VIPA

Die Projektierung eines IM 253 DP-Slaves am DP-Master von VIPA finden Sie in der Beschreibung zum DP-Master in diesem Kapitel.

**Einsatz unter  
S7-400 von  
Siemens**

Das System S7-400 von Siemens verwendet bei der Projektierung die Doppelwortadressierung, d.h. bei der Projektierung wird je Modul ein Doppelwort vergeben. Gerade bei den Digital-Modulen bleiben die höheren Bytes der Doppelworte ungenutzt. Diesen Umstand können Sie umgehen, indem Sie die VIPA GSD-Datei für die S7-400 von Siemens einsetzen. Sie finden diese GSD-Datei im Verzeichnis `..\S7-400\` auf der mitgelieferten Diskette. Wenn Sie diese GSD-Datei einsetzen, projektieren Sie zunächst alle digitalen Eingänge, dann alle digitalen Ausgänge indem Sie die jeweilige Summe in Byte angeben. Gibt es keine Ein- bzw. Ausgangs-Module, müssen Sie 0Byte angeben.

Nach den digitalen Modulen projektieren Sie Ihre analogen Module in gewohnter Weise.

**Hinweis!**

Bitte beachten Sie hierbei, dass Sie im S7-400-System von Siemens unter Modulparametern der Analog-Module die Steckplatznummer angeben müssen. Hierbei befindet sich das erste Peripherie-Modul auf Steckplatz 0.

**Einsatz mit einer  
IM 308-B von  
Siemens**

Für die Projektierertools COM ET200 von Siemens (DOS-Version für IM 308B) und COM Profibus von Siemens (Windows-Version) befinden sich zusätzlich Typ-Dateien zur Projektierung auf der Diskette.

Hinweise hierzu finden Sie in `readme.txt` auf der Diskette.

**Hinweis!**

Bei Einsatz in Verbindung mit einer IM308-B von Siemens könnte es möglich sein, dass Sie für die einwandfreie Funktion Ihres Systems gewisse Modifikationen an der Typdatei vornehmen müssen. Setzen Sie sich bitte hierzu mit der VIPA-Hotline in Verbindung.

---

**Parametrierung in  
einem redundanten  
System**

Nur der Slave-Teil, der zuerst in DataExchange geht (systembedingt immer der linke Slave), wird automatisch zum aktiven Slave, hat Zugriff auf die Peripherie-Module und kann diese parametrieren.

Zur Umparametrierung von Peripherie-Modulen ist darauf zu achten, dass neue Parameter nur von einem aktiven Master-Slave-System übermittelt werden können. Vorher müssen sich beide Slaves im WAITPARAM-Zustand befinden.

---

**Anlaufverhalten  
IM 253DP-Slave**

Nach dem Einschalten durchläuft der DP-Slave einen Selbsttest. Hierbei überprüft er seine internen Funktionen und die Kommunikation über den Rückwandbus. Nach fehlerfreiem Hochlauf geht der Buskoppler in den Zustand "READY" über. Im Zustand READY erhält der DP-Slave vom DP-Master seine Parameter und geht bei gültigen Parametern in den Zustand "DataExchange" DE über (DE leuchtet).

Bei Kommunikationsstörungen am Rückwandbus geht der Profibus-Slave zunächst in STOP und läuft nach ca. 2 Sekunden erneut hoch. Sobald der Test positiv abgeschlossen ist, blinkt die RD-LED.

## IM 253DP - Slave - Parameter

**Übersicht** Bei Einsatz der in diesem Handbuch aufgeführten DP-Slaves haben Sie für die Parametrierung 4 Parameter, die je Slave individuell verwendet werden.

**Parameter** Folgende Parameter stehen zur Verfügung:

<i>Steckplatznummern</i>
<p>Aus Kompatibilitätsgründen können Sie hier einstellen, mit welchem Wert die Steckplatznummerierung beginnen soll. Für VIPA-Slaves mit Ausgabestand 4 und älter ist dieser Parameter erforderlich. Von DP-Slaves ab Ausgabestand 5 wird dieser Parameter ignoriert.</p> <p>Folgende Werte stehen zur Auswahl:</p> <ul style="list-style-type: none"> <li>0: Steckplatz 0 (default)</li> <li>1: Steckplatz 1</li> </ul>
<i>Sync Mode</i>
<p>Im Sync Mode werden V-Bus-Zyklus (VIPA-Rückwandbus-Kommunikation) und DP-Zyklus (Profibus-DP-Kommunikation) synchronisiert.</p> <p>Dies gewährleistet, dass pro V-Bus-Zyklus eine Profibus-Übertragung stattfindet.</p> <p>Folgende Werte stehen zur Auswahl:</p> <ul style="list-style-type: none"> <li>Sync Mode aus: DP- und V-Bus-Zyklus laufen asynchron (default)</li> <li>Sync Mode an: DP- und V-Bus-Zyklus laufen synchron</li> </ul>
<i>Diagnose</i>
<p>Über diesen Parameter können Sie die Diagnosefunktion des Slaves beeinflussen:</p> <p>Folgende Werte stehen zur Auswahl:</p> <ul style="list-style-type: none"> <li>aktiviert: Aktiviert die Diagnosefunktion des Slaves (default)</li> <li>deaktiviert: Schaltet die Diagnosefunktion des Slaves ab</li> </ul>
<i>Redundanz-Diagnose</i>
<p>Über diesen Parameter können Sie die Redundanz-Diagnosefunktion des Slaves beeinflussen. Dieser Parameter wird nun von den redundanten Slaves unterstützt.</p> <p>Folgende Werte stehen zur Auswahl:</p> <ul style="list-style-type: none"> <li>aktiviert: Aktiviert die Redundanz-Diagnosefunktion des Slaves (default)</li> <li>deaktiviert: Schaltet die Redundanz-Diagnosefunktion des Slaves ab</li> </ul>



## IM 253DP - Slave - Diagnosefunktionen

<b>Übersicht</b>	<p>Die umfangreichen Diagnosefunktionen von Profibus-DP ermöglichen eine schnelle Fehlerlokalisierung. Die Diagnosemeldungen werden über den Bus übertragen und beim Master zusammengefasst.</p> <p>Zusätzlich werden in jeden Profibus-Slave von VIPA die letzten 100 Diagnosemeldungen mit einem Zeitstempel in einem RAM gespeichert bzw. im Flash gesichert und können mit einer Software ausgewertet werden.</p> <p>Setzen Sie sich hierzu bitte mit der VIPA-Hotline in Verbindung!</p>
<b>Interne Diagnose Systemmeldungen</b>	<p>Das System legt auch Diagnosemeldungen ab wie die Zustände "Ready" bzw. "DataExchange", die nicht an den Master weitergeleitet werden.</p> <p>Mit jedem Zustandswechsel zwischen "Ready" und "DataExchange" sichert der Profibus-Slave den Diagnose-RAM-Inhalt in einem Flash-ROM und schreibt diesen mit jedem Neustart in das RAM zurück.</p>
<b>Diagnosedaten manuell sichern</b>	<p>Über die kurzzeitige Einstellung von 00 am Adress-Schalter können Sie die Diagnose-Daten während des "DataExchange" im Flash-ROM sichern.</p>
<b>Diagnosemeldung bei Spannungsausfall</b>	<p>Bei Spannungsausfall bzw. sinkender Spannung wird sofort ein Zeitstempel im EEPROM gespeichert. Sollte noch genügend Spannung vorhanden sein, erfolgt eine Diagnoseausgabe an den Master.</p> <p>Beim nächsten Neustart wird eine Unterspannung/Abschaltung-Diagnosemeldung aus dem Zeitstempel des EEPROMs generiert und im Diagnose-RAM abgelegt.</p>
<b>Diagnose direkt am Profibus-Slave-Modul</b>	<p>Bei den VIPA Profibus-Slaves können Sie über ein bei VIPA erhältliches Download-Kabel und der zugehörigen Software "Slave Info Tool" die aktuellen Diagnose-Daten direkt aus dem Modul in Ihren PC übertragen und dort analysieren.</p>
<b>Diagnosezusatz des IM 253DPR</b>	<p>Bei Einsatz eines redundanten Slaves wird an das Diagnose-Telegramm ein 8Byte großer Redundanzstatus angehängt. Dieser Diagnosezusatz wird nicht intern abgelegt. Durch zusätzliche Projektierung des Status-Moduls "Statusbyte IM253-2DP50" als letztes "Modul" (ganz rechts) haben Sie die Möglichkeit 2Byte des Redundanzstatus im Peripheriebereich einzublenden.</p> <p>Dieses virtuelle Status-"Modul" ist ab GSD-Version 1.30 verfügbar.</p>

**Aufbau der Diagnosedaten über Profibus**

Die Diagnose-Meldungen, die vom Profibus-Slave erzeugt werden, haben immer eine Länge von 23Byte. Man nennt diese auch *Gerätebezogene Diagnose-Daten*.

Sobald der Profibus-Slave an den Master eine Diagnose sendet, werden den 23Byte Diagnosedaten 6Byte Normdiagnose-Daten und 1Byte Header vorangestellt:

Byte 0 ... Byte 5	Normdiagnose-Daten	wird nur bei Transfer über Profibus an den Master vorangestellt
Byte 6	Header für gerätebezogene Diagnose	
<b>Byte 7 ... 29</b>	<b>Gerätebezogene Diagnose</b>	<b>Diagnose, die intern abgelegt wird</b>
Byte x ... Byte x+8	Redundanzstatus eines redundanten DP-Slaves	wird nur bei Transfer über Profibus und bei Einsatz des redundanten DP-Slaves an den Master angehängt

**Norm-Diagnosedaten**

Bei der Übertragung einer Diagnose an den Master werden die Slave-Norm-Diagnose-Daten und ein Header-Byte den gerätebezogenen Diagnosebyte vorangestellt. Nähere Angaben zum Aufbau der Slave-Normdiagnose-Daten finden Sie in den Normschriften der Profibus Nutzer Organisation. Die Slave-Normdiagnosedaten haben folgenden Aufbau:

Byte	Bit 7 ... Bit 0
0	Bit 0: fest auf 0 Bit 1: Slave nicht bereit für Datenaustausch Bit 2: Konfigurationsdaten stimmen nicht überein Bit 3: Slave hat externe Diagnosedaten Bit 4: Slave unterstützt angeforderte Funktion nicht Bit 5: fest auf 0 Bit 6: Falsche Parametrierung Bit 7: fest auf 0
1	Bit 0: Slave muss neu parametrierung werden Bit 1: Statistische Diagnose Bit 2: fest auf 1 Bit 3: Ansprechüberwachung aktiv Bit 4: Freeze-Kommando erhalten Bit 5: Sync-Kommando erhalten Bit 6: reserviert Bit 7: fest auf 0
2	Bit 0 ... Bit 6: reserviert Bit 7: Diagnosedaten Überlauf
3	Masteradresse nach Parametrierung FFh: Slave ist ohne Parametrierung
4	Identnummer High Byte
5	Identnummer Low Byte

**Header für  
gerätebezogene  
Diagnose**

Dieses Byte wird nur bei der Übertragung über den Profibus den gerätebezogenen Diagnosedaten vorangestellt.

Byte	Bit 7 ... Bit 0
6	Bit 0 ... Bit 5: Länge gerätebezogene Diagnose inkl. Byte 6 Bit 6 ... Bit 7: fest auf 0

**Gerätebezogene  
Diagnose**

Byte	Bit 7 ... Bit 0
7 ... 29	Gerätebezogene Diagnosedaten, die intern im Slave gespeichert und ausgewertet werden können.

**Aufbau der  
gerätebezogenen  
Diagnosedaten  
im DP-Slave**

Ab dem Ausgabestand 6 werden alle Diagnosen, die der Profibus-Slave erzeugt, zusammen mit einem Zeitstempel in einem Ringpuffer abgelegt. In dem Ringpuffer befinden sich immer die letzten 100 Diagnose-Meldungen.

Diese Meldungen können Sie über das "Slave Info Tool" auswerten.

Da Normdiagnosedaten (Byte 0 ... Byte 5) und Header (Byte 6) nicht gespeichert werden, entsprechen Byte 0 ... Byte 23 den Bytes 7 ... Byte 30 bei Übertragung über Profibus.

Die gerätebezogenen Diagnosedaten haben folgenden Aufbau:

Byte	Bit 7 ... Bit 0
0	Meldung 0Ah: DP-Parameterfehler 14h: DP-Konfigurationsfehler Länge 15h: DP-Konfigurationsfehler Eintrag 1Eh: Unterspannung/Abschaltung 28h: V-Bus Parametrierfehler 29h: V-Bus Initialisierungsfehler 2Ah: V-Bus Busfehler 2Bh: V-Bus Quittungsverzug 32h: Diagnosealarm System 200 33h: Prozessalarm System 200 3Ch: Neue DP-Adresse wurde gesetzt 3Dh: Slave im Ready-Zustand (nur intern) 3Eh: Slave im DataExchange-Zustand (nur intern)
1	Modul-Nr. bzw. Steckplatz 1 ... 32: Modul-Nr. bzw. Steckplatz 0: Modul-Nr. bzw. Steckplatz unbekannt
2 ... 23	Zusatzinformationen zur Meldung in Byte 0

**Übersicht der Diagnose-Meldungen**

Nachfolgend sind alle Meldungen aufgeführt, die Bestandteil einer Diagnose sein können. Entsprechend der Meldung (Byte 0) gestaltet sich der Aufbau von Byte 2 ... Byte 23. Bei Übertragung der Diagnose über Profibus in den Master entspricht im Master Byte 7 dem Byte 0 im Slave. Die Längenangabe steht für die "Länge der Diagnosedaten" bei Übertragung über Profibus.

**0Ah**

*DP-Parameterfehler*

Länge: 8

Das Parametertelegramm ist zu kurz oder zu lang

Byte	Bit 7 ... Bit 0
0	0Ah: DP-Parameterfehler
1	Modul-Nr. bzw. Steckplatz 1 ... 32: Modul-Nr. bzw. Steckplatz 0: Modul-Nr. bzw. Steckplatz unbekannt
2	Länge User-Parameterdaten
3	Modus 0: Standard-Modus 1: 400-er Modus
4	Anzahl der Digital-Module (Slave)
5	Anzahl der Analog-Module (Slave)
6	Anzahl der Analog-Module (Master)

**14h**

*DP-Konfigurationsfehler - Länge*

Länge: 6

Abhängig vom Modus wird die Länge des Konfigurationstelegramms mit der Länge der Defaultkonfiguration (erkannte Module am V-Bus) verglichen.

Byte	Bit 7 ... Bit 0
0	14h: DP-Konfigurationsfehler - Länge
1	Modul-Nr. bzw. Steckplatz 1 ... 32: Modul-Nr. bzw. Steckplatz 0: Modul-Nr. bzw. Steckplatz unbekannt
2	Anzahl der Konfigurationsdaten (Master)
4	Anzahl der Konfigurationsdaten (Slave)
3	Modus 0: Standard-Modus 1: 400-er Modus

- 15h** *DP-Konfigurationsfehler - Eintrag* Länge: 6  
 Abhängig vom Modus und nach Übereinstimmung der Konfigurationslängen, werden die einzelnen Einträge im Konfigurationstelegramm mit der Default-Konfiguration verglichen.

Byte	Bit 7 ... Bit 0
0	15h: DP-Konfigurationsfehler - Eintrag
1	Modul-Nr. bzw. Steckplatz 1 ... 32: Modul-Nr. bzw. Steckplatz 0: Modul-Nr. bzw. Steckplatz unbekannt
2	Konfigurationsbyte Master (Modulkennung)
4	Konfigurationsbyte Slave (Modulkennung)
3	Modus 0: Standard-Modus 1: 400-er Modus

- 1Eh** *Unterspannung/Abschaltung* Länge: 2  
 Bei Spannungsausfall bzw. sinkender Spannung wird sofort ein Zeitstempel im EEPROM gespeichert. Sollte noch genügend Spannung vorhanden sein, erfolgt eine Diagnoseausgabe an den Master.  
 Beim nächsten Neustart wird eine Spannungsfehler-Diagnosemeldung aus dem Zeitstempel des EEPROMs generiert und im RAM abgelegt.

Byte	Bit 7 ... Bit 0
0	1Eh: Unterspannung/Abschaltung

- 28h** *V-Bus Parametrierfehler* Länge: 3  
 Die Parametrierung auf dem angegebenen Steckplatz schlug fehl.

Byte	Bit 7 ... Bit 0
0	28h: V-Bus Parametrierfehler
1	Modul-Nr. bzw. Steckplatz 1 ... 32: Modul-Nr. bzw. Steckplatz 0: Modul-Nr. bzw. Steckplatz unbekannt

- 29h** *V-Bus Initialisierungsfehler* Länge: 2  
 Allgemeiner Rückwandbusfehler

Byte	Bit 7 ... Bit 0
0	29h: V-Bus Initialisierungsfehler

- 2Ah** *V-Bus Busfehler* Länge: 2  
 Hardwarefehler oder Modul ausgefallen

Byte	Bit 7 ... Bit 0
0	2Ah: V-Bus Fehler

**2Bh** *V-Bus Quittungsverzug* Länge: 2  
Lesen oder Schreiben der Digital-Module schlug fehl

Byte	Bit 7 ... Bit 0
0	2Bh: V-Bus Quittungsverzug

**32h** *Diagnosealarm System 200V* Länge: 16

Byte	Bit 7 ... Bit 0
0	32h: Diagnosealarm System 200V
1	Modul-Nr. bzw. Steckplatz 1 ... 32: Modul-Nr. bzw. Steckplatz 0: Modul-Nr. bzw. Steckplatz unbekannt
2 ... 14	Daten Diagnosealarm

**33h** *Prozessalarm System 200V* Länge: 16

Byte	Bit 7 ... Bit 0
0	33h: Prozessalarm System 200V
1	Modul-Nr. bzw. Steckplatz 1 ... 32: Modul-Nr. bzw. Steckplatz 0: Modul-Nr. bzw. Steckplatz unbekannt
2 ... 14	Daten Prozessalarm

**3Ch** *Neue DP-Adresse wurde gesetzt* Länge: 2  
Nach Empfang des Dienstes mit "Set Slave Address" sendet der Slave die Meldung dieser Diagnose und bootet neu. Danach ist er mit der neuen Adresse am Bus.

Byte	Bit 7 ... Bit 0
0	3Ch: Neue DP-Adresse wurde gesetzt

**3Dh** *Slave im Ready-Zustand* Länge: keine (nur intern)  
Die Angabe, dass der Slave sich im READY-Zustand befindet wird nur intern abgelegt und nicht über Profibus weitergeleitet.

Byte	Bit 7 ... Bit 0
0	3Dh: Slave im READY-Zustand

**3Eh** *Slave im DataExchange Zustand* Länge: keine (nur intern)  
Die Angabe, dass der Slave sich im DataExchange-Zustand befindet wird nur intern abgelegt und nicht über Profibus weitergeleitet.

Byte	Bit 7 ... Bit 0
0	3Eh: Slave im DataExchange-Zustand

### Redundanzstatus bei Einsatz des IM 253DPR

Bei Einsatz eines redundanten Slaves wird die Diagnose-Meldung um 8Byte Redundanzstatus-Daten erweitert. Dieser Diagnosezusatz wird nicht intern im Diagnosepuffer abgelegt. Der Redundanzstatus hat folgenden Aufbau:

#### *Redundanzstatus*

Byte	Beschreibung
X	08h: Länge Redundanzstatus fest auf 8
X+1	80h: Typ Redundanzstatus
X+2	00h: reserviert, fest auf 00h
X+3	00h: reserviert, fest auf 00h
X+4	00h: reserviert, fest auf 00h
X+5	Red_State Slave, der mit dem entsprechenden Master kommuniziert) Bit 0 = Slave ist Backup-Slave Bit 1 = Slave ist Primary-Slave Bit 2 = reserviert Bit 3 = reserviert Bit 4 = Slave ist im DataExchange Bit 5 = reserviert Bit 6 = reserviert Bit 7 = reserviert
X+6	Red_State des anderen Slaves
X+7	00h: reserviert, fest auf 00h

### Redundanzstatus im Peripherie-Bereich einblenden

Ab der GSD-Version 1.30 von VIPA ist im Hardwarekatalog das virtuelle Modul "Statusbyte IM253-2DP50" verfügbar. Bei Einsatz dieses Moduls in der Projektierung können Sie einen 2Byte großen Adress-Bereich angeben, in dem das "Red-State"-Byte beider Slaves abgelegt werden soll. Bitte beachten Sie, dass dieses Modul in der Steck-Reihenfolge immer als letztes Modul zu projektieren ist, ansonsten meldet der Slave einen Parametrierfehler.

### Diagnose (de)aktivieren

Über das Parametrierfenster der Slaves können Sie das Diagnoseverhalten bestimmen, indem Sie die Diagnose oder den Redundanzstatus aktivieren bzw. deaktivieren.

## Aufbaurichtlinien

### Profibus allgemein

- Ein VIPA Profibus-DP-Netz darf nur in Linienstruktur aufgebaut werden.
- Profibus-DP besteht aus mindestens einem Segment mit mindestens einem Master und einem Slave.
- Ein Master ist immer in Verbindung mit einer CPU einzusetzen.
- Profibus unterstützt max. 125 Teilnehmer.
- Pro Segment sind max. 32 Teilnehmer zulässig.
- Die maximale Segmentlänge hängt von der Übertragungsrate ab:
 

9,6 ... 187,5kBaud	→	1000m
500kBaud	→	400m
1,5MBaud	→	200m
3 ... 12MBaud	→	100m
- Maximal 10 Segmente dürfen gebildet werden. Die Segmente werden über Repeater verbunden. Jeder Repeater zählt als Teilnehmer.
- Alle Teilnehmer kommunizieren mit der gleichen Baudrate. Die Slaves passen sich automatisch an die Baudrate an.

### Optisches System

- Es darf nur ein optischer Master in einer Linie verwendet werden.
- Mehrere Master dürfen mit einer CPU eingesetzt werden, sofern sich diese auf dem gleichen Rückwandbus befinden (max. Stromaufnahme beachten).
- Die maximale LWL-Länge darf zwischen zwei Slaves, unabhängig von der Übertragungsrate max. 300m bei HCS-LWL bzw. max. 50m bei POF-LWL betragen.
- Die Anzahl der Busteilnehmer richtet sich nach der Baudrate:
 

$\leq 1,5$ MBaud	→	17 Teilnehmer inkl. Master
3MBaud	→	15 Teilnehmer inkl. Master
6MBaud	→	7 Teilnehmer inkl. Master
12MBaud	→	4 Teilnehmer inkl. Master
- Der Bus muss nicht abgeschlossen werden.



#### Hinweis!

Sie sollten bei einem optischen Teilnehmer am Busende die Buchse für den nachfolgenden Teilnehmer abdecken, ansonsten besteht Blendungsgefahr und das Empfangsteil kann durch Fremdeinstrahlung gestört werden. Verwenden Sie hierzu die mitgelieferten Gummi-Stäbchen und stecken Sie die Stäbchen in die zwei Öffnungen des LWL-Anschlusses.

### Elektrisches System

- Der Bus ist an beiden Enden abzuschließen.
- Master und Slaves sind beliebig mischbar.



**Gemischtes System**

- Ein LWL-Master darf nur direkt über einen **Optical Link Plug (OLP)** in ein elektrisches System eingekoppelt werden, d.h. zwischen Master und OLP darf sich kein Slave befinden.
- Zwischen zwei Mastern darf sich maximal eine Umsetzung (OLP) befinden.

**Aufbau und Einbindung in Profibus**

- Bauen Sie Ihr Profibus-System mit den entsprechenden Modulen auf.
- Stellen Sie an Ihrem Buskoppler eine Adresse ein, die in Ihrem Bus noch nicht verwendet wird.
- Übertragen Sie die mitgelieferte GSD-Datei in Ihr Projektiersystem und projektieren Sie Ihr System.
- Übertragen Sie Ihre Projektierung auf Ihren Master.
- Schließen Sie das Profibus-Kabel am Koppler an und schalten Sie die Spannungsversorgung ein.

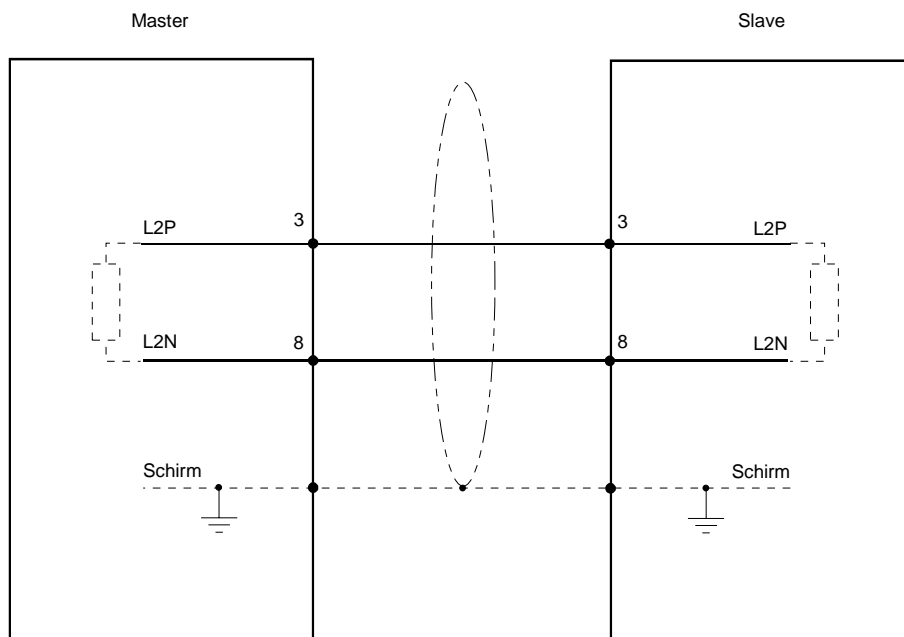


**Hinweis!**

Die Profibus-Leitung muss mit ihrem Wellenwiderstand abgeschlossen werden. Bitte beachten Sie, dass Sie bei dem jeweilig letzten Teilnehmer den Bus durch Zuschalten eines Abschlusswiderstands abschließen. Bei einem LWL-Profibus-System ist ein Abschluss nicht erforderlich!

**Profibus mit RS485**

Profibus verwendet als Übertragungsmedium eine geschirmte, verdrehte Zweidrahtleitung auf Basis der RS485-Schnittstelle. Die Abbildung zeigt eine Profibusverbindung unter RS485 mit ange deuteten Abschlusswiderständen:

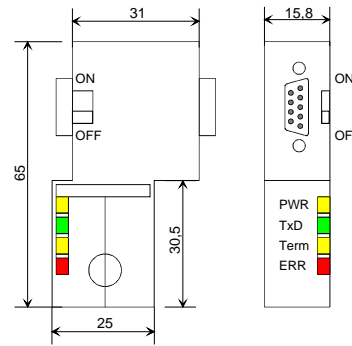


**Busanschluss**



In Systemen mit mehr als zwei Stationen werden alle Teilnehmer parallel verdrahtet. Hierzu ist das Buskabel unterbrechungsfrei durchzuschleifen.

Unter der Best.-Nr. VIPA 972-0DP10 erhalten Sie von VIPA den Stecker "EasyConn". Dies ist ein Busanschlussstecker mit zuschaltbarem Abschlusswiderstand und integrierter Busdiagnose.

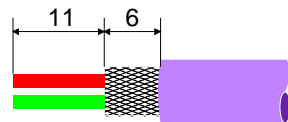


alle Maße in mm

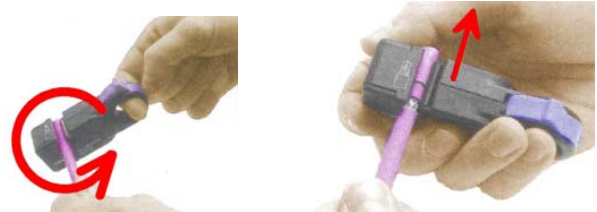


Zum Anschluss dieses Steckers verwenden Sie bitte die Standard Profibus-Leitung Typ A mit Drahtseele nach EN50170.

Von VIPA erhalten Sie unter der Best.-Nr. VIPA 905-6AA00 das "EasyStrip" Abisolierwerkzeug, das Ihnen den Anschluss des EasyConn-Steckers sehr vereinfacht.



alle Maße in mm

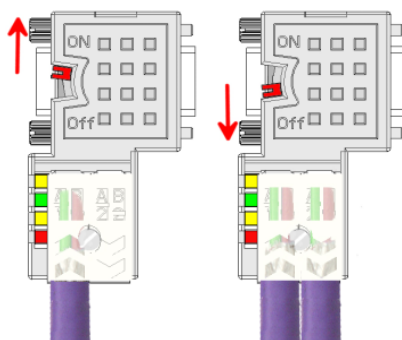


**Achtung!**

Zur Vermeidung von Übertragungsproblemen durch Reflexionen muss das Buskabel immer an den Leitungsenden mit dem Wellenwiderstand abgeschlossen werden!

**Leitungsabschluss**

Auf dem Busanschlussstecker befindet sich unter anderem ein Schalter, mit dem Sie einen Abschlusswiderstand zuschalten können.



**Achtung!**

Der Abschlusswiderstand wird nur wirksam, wenn der Stecker an einem Slave gesteckt ist und der Slave mit Spannung versorgt wird.

**Hinweis!**

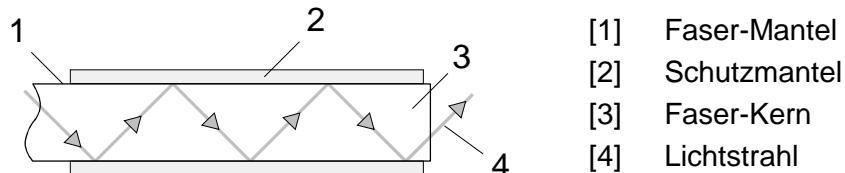
Eine ausführliche Beschreibung zum Anschluss und zum Einsatz der Abschlusswiderstände liegt dem Stecker bei.

**Profibus mit LWL**

Der Lichtwellenleiter (LWL) dient zur Übertragung von Signalen mit Hilfe elektromagnetischer Wellen im Bereich optischer Frequenzen. Da die Brechzahl des Faser-Mantels niedriger ist als die des Faser-Kerns, findet eine Totalreflexion statt. Aufgrund der Totalreflexion kann der Lichtstrahl im Lichtleiter nicht austreten und wird bis zum Faser-Ende geführt.

Die LWL-Faser ist mit einer Schutzumhüllung (Coating) versehen.

Den prinzipiellen Aufbau eines Lichtwellenleiters sehen Sie in der folgenden Abbildung:



Das Lichtwellenleitersystem arbeitet mit Lichtimpulsen von monochromatischem Licht bei 650nm Wellenlänge. Der Lichtwellenleiter ist, wenn nach den Verlegerichtlinien der LWL-Hersteller verlegt wurde, völlig unempfindlich gegenüber Störspannungen von außen. Ein Lichtwellenleitersystem wird in Linienstruktur aufgebaut. Jedes Gerät ist mit einem Hin- und Rückleiter zu verbinden (Zweileiter). Ein Abschluss am letzten Gerät ist nicht erforderlich.

Für ein Profibus-LWL-Netz sind maximal 126 Teilnehmer (einschließlich Master) zulässig. Die maximale Strecke, die zwischen zwei Geräten liegen darf, beträgt max. 50m.

**Vorteile LWL gegenüber Kupferkabel**

- große Übertragungsbandbreite
- niedrige Signaldämpfung
- kein Übersprechen zwischen den Adern
- keine Beeinflussung durch äußere elektrische Störfelder
- keine Potenzialdifferenzen
- Blitzschutz
- verlegbar in explosionsgefährdetem Umfeld
- leichter und flexibler
- korrosionsbeständig
- abhörsicher

**Verkabelung mit Lichtwellenleiter unter Profibus**

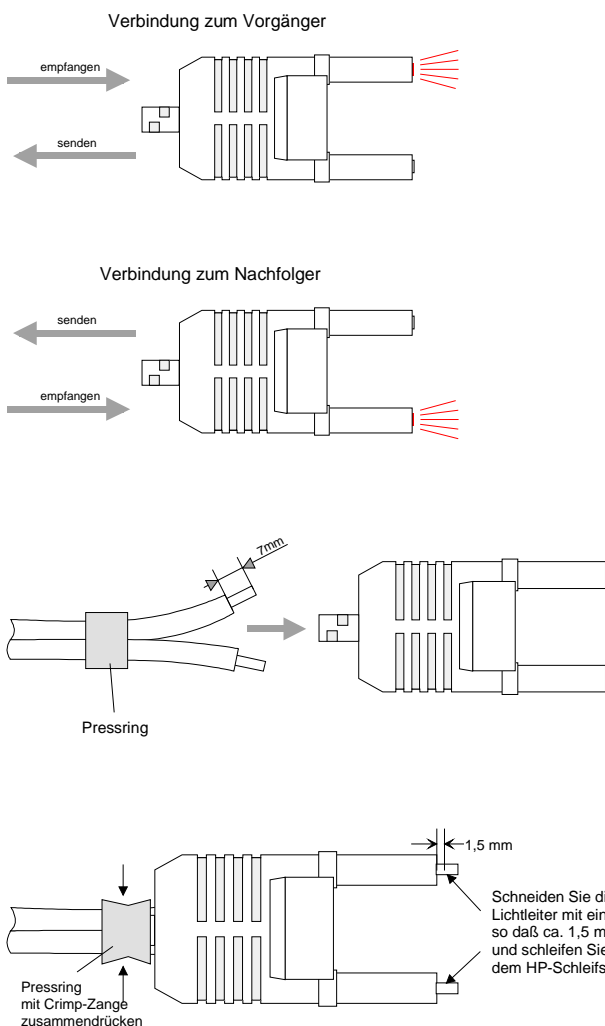
Der VIPA Profibus-Koppler mit Lichtwellenleiter-Interface verwendet als Übertragungsmedium Kunststoff-Lichtwellenleiter in Zweileiterausführung. Beim Anschluss Ihres Profibus-LWL-Kopplers ist folgendes zu beachten: Vorgänger und Nachfolger sind jeweils mit einem Zweileiter-LWL-Kabel zu verbinden.

Auf dem VIPA Bus-Koppler befinden sich 4 LWL-Anschlüsse. An der Buchsenfarbe können Sie die Kommunikationsrichtung erkennen (dunkel: empfangen, hell: senden).

Bei eingeschaltetem Bus erkennen Sie am Licht die Faser für den Empfang und die dunkle Faser für das Senden. VIPA empfiehlt Ihnen, die LWL-Stecker von der Firma Hewlett Packard (HP) zu verwenden. Diese Stecker sind in zwei Ausführungen erhältlich:

- LWL-Stecker mit Crimp-Montage
- LWL-Stecker ohne Crimp-Montage

**LWL-Stecker mit Crimp-Montage**



**HP-Best.-Nr.: HFBR-4506 (grau)  
HFBR-4506B (schwarz)**

Vorteil: Verpolungssicherheit

Sie können den Stecker nur so in den Koppler stecken, dass die hier gezeigte Steckerseite nach rechts gerichtet ist.

Nachteil: Spezial-Zange erforderlich

Für die Montage des Pressrings für die Zugentlastung benötigen Sie eine spezielle Crimp-Zange von Hewlett Packard (HP-Best.-Nr.: HFBR-4597).

**Steckermontage**

Für die Steckermontage schieben Sie zuerst den Pressring über den Zweileiter. Trennen Sie die zwei Adern auf einer Länge von ca. 5cm voneinander.

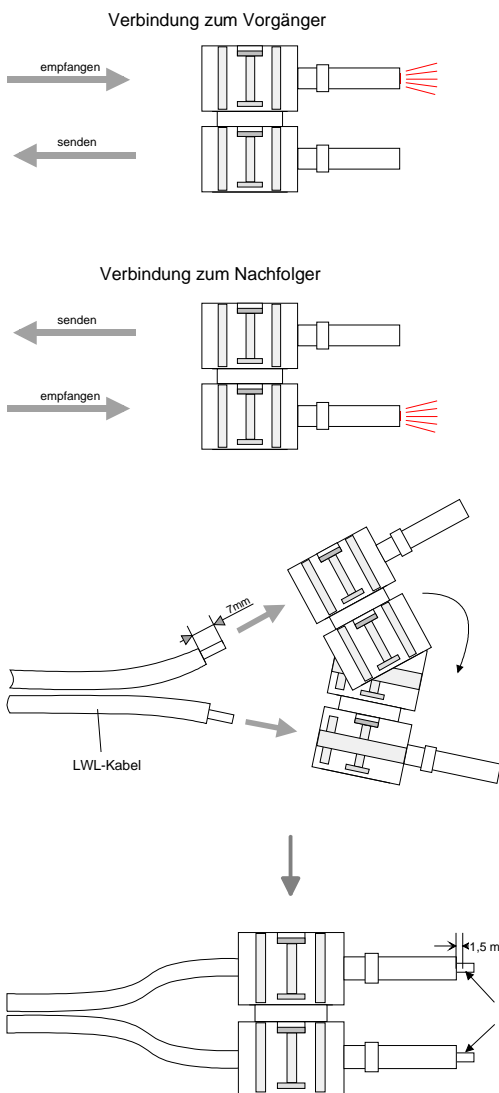
Mit einer Abisolierzange entfernen Sie die Schutzumhüllung, dass ca. 7mm der Faser sichtbar werden.

Nun schieben Sie beide Adern in den Stecker, so dass die Lichtleiterenden vorn herauschauen. Achten Sie bitte hierbei auf die Polarität der Adern (s.o.).

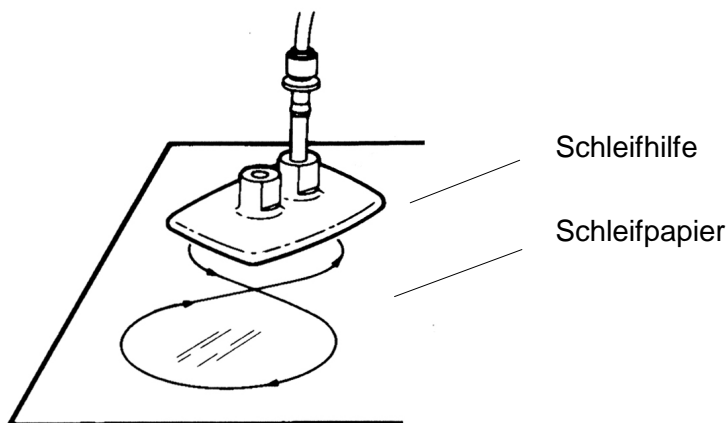
Schieben Sie den Pressring auf den Stecker und quetschen Sie den Ring mit der Crimp-Zange zusammen.

Ein Beschreibung, wie sie die Lichtleiterenden abschneiden und polieren, finden Sie weiter unten nach der Beschreibung des 2. Steckertyps.

**LWL-Stecker ohne Crimp-Montage**



**LWL-Enden abschneiden und schleifen**



**HP-Best.-Nr.: HFBR-4531**

Vorteil: keine Spezial-Zange erforderlich  
 Bei diesem Steckertyp ist die Zugentlastung in das Steckergehäuse integriert.

Durch einfaches Zusammendrücken der zwei Steckergehäuse-Hälften wird der Lichtwellenleiter sicher eingeklemmt.

Mit diesem System lassen sich Simplex- und Duplexstecker erstellen. Sie können durch Zusammendrücken zweier Steckerhälften einen Simplexstecker und durch Zusammendrücken zweier Stecker einen Duplexstecker erzeugen.

Nachteil: nicht verpolungssicher.

Sie können den Stecker in zwei Positionen stecken. Die Polarität prüfen Sie im eingeschalteten Zustand. Die leuchtende Faser ist die Faser für den Empfang.

**Steckermontage:**

Für die Montage eines Duplexsteckers sind 2 Stecker erforderlich. Trennen Sie die zwei Adern auf einer Länge von ca. 5cm voneinander. Mit einer Abisolierzange entfernen Sie die Schutzumhüllung, dass ca. 7mm der Faser sichtbar werden.

Nun schieben Sie beide Adern in den Stecker, so dass die Lichtleiterenden vorn heraus schauen. Achten Sie bitte hierbei auf die Polarität der Adern (s.o.).

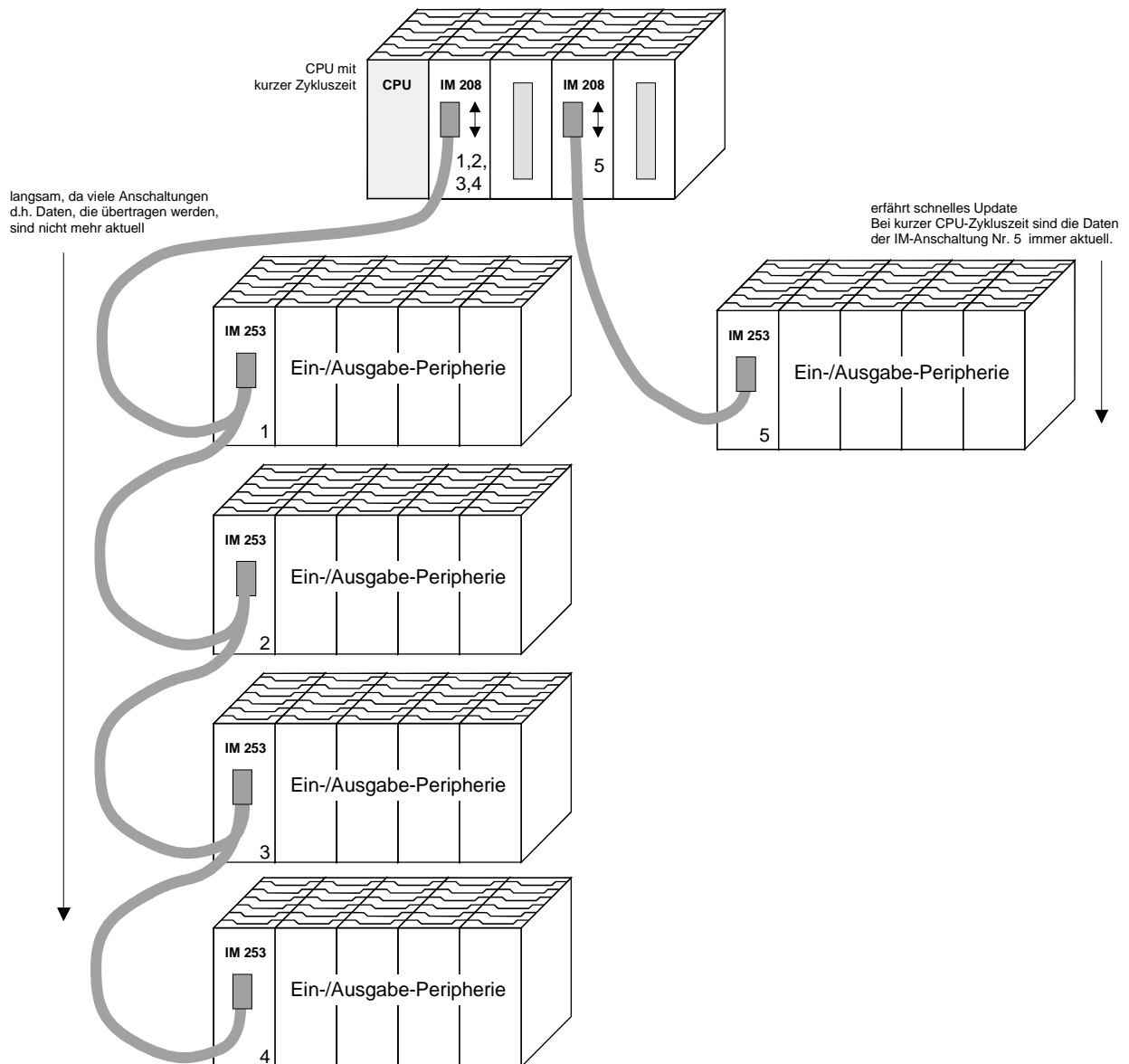
Schneiden Sie mit einer Klinge die Faser ab, so dass 1,5mm Faser noch sichtbar sind. Verwenden Sie zum Planschleifen das Schleifset von HP (HP-Best.-Nr.:HFBR-4593).

Stecken Sie den Stecker in die Schleifhilfe und schleifen Sie die Faserenden plan, wie auf dem Bild gezeigt. In der Bedienungsanleitung, die diesem Set beiliegt, finden Sie eine nähere Beschreibung zur Vorgehensweise.

### Beispiele für Profibus-Netze

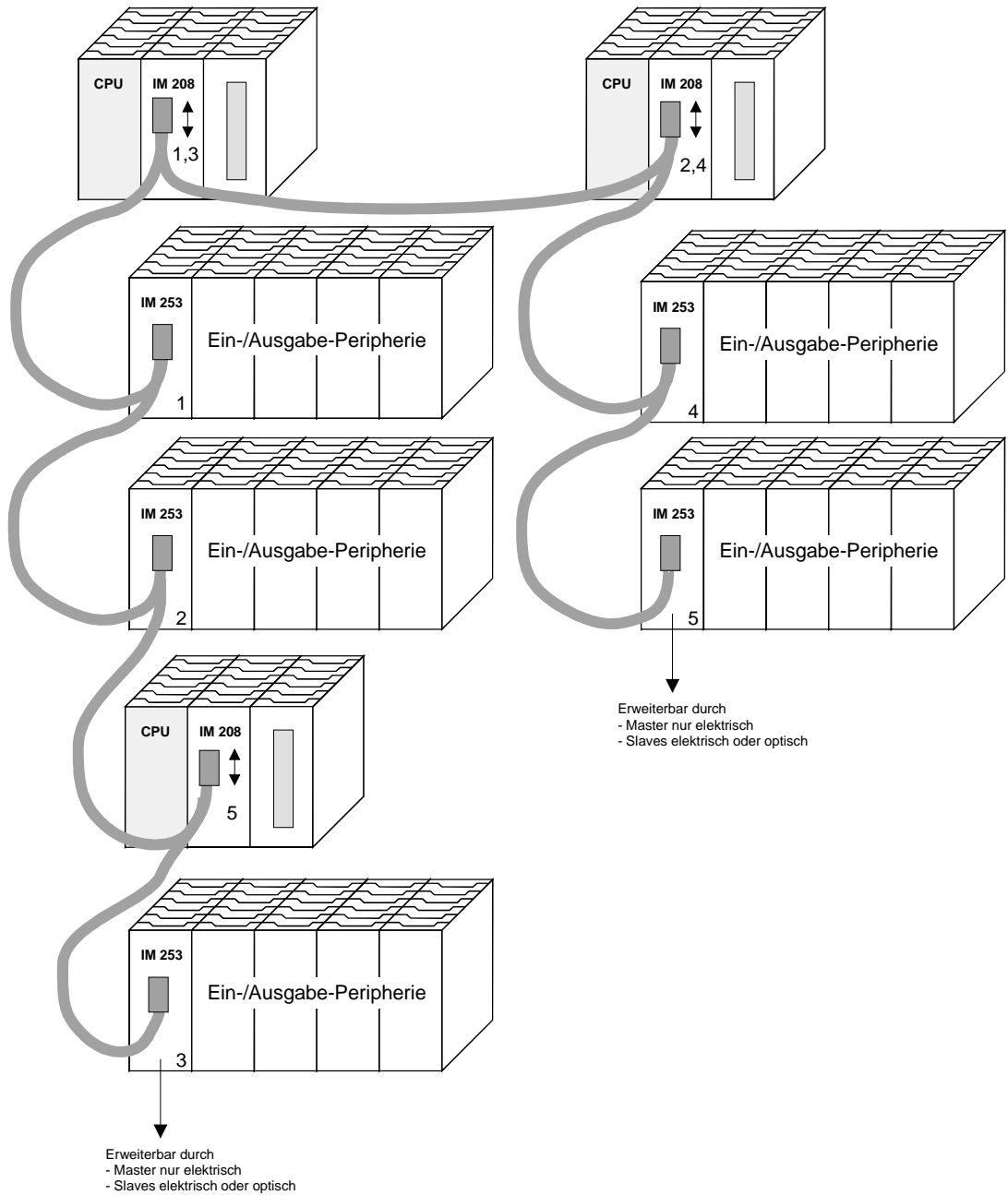
#### Eine CPU und mehreren Master-Anschaltungen

Die CPU sollte eine kurze Zykluszeit haben, so ist gewährleistet dass die Daten von Slave Nr. 5 (rechts) immer aktuell sind. Dieser Aufbau ist nur sinnvoll, wenn am langsamen Strang (links) Slaves angekoppelt sind, deren Daten-Aktualität unwichtig ist. Hier sollten auch keine Module liegen, die einen Alarm auslösen.

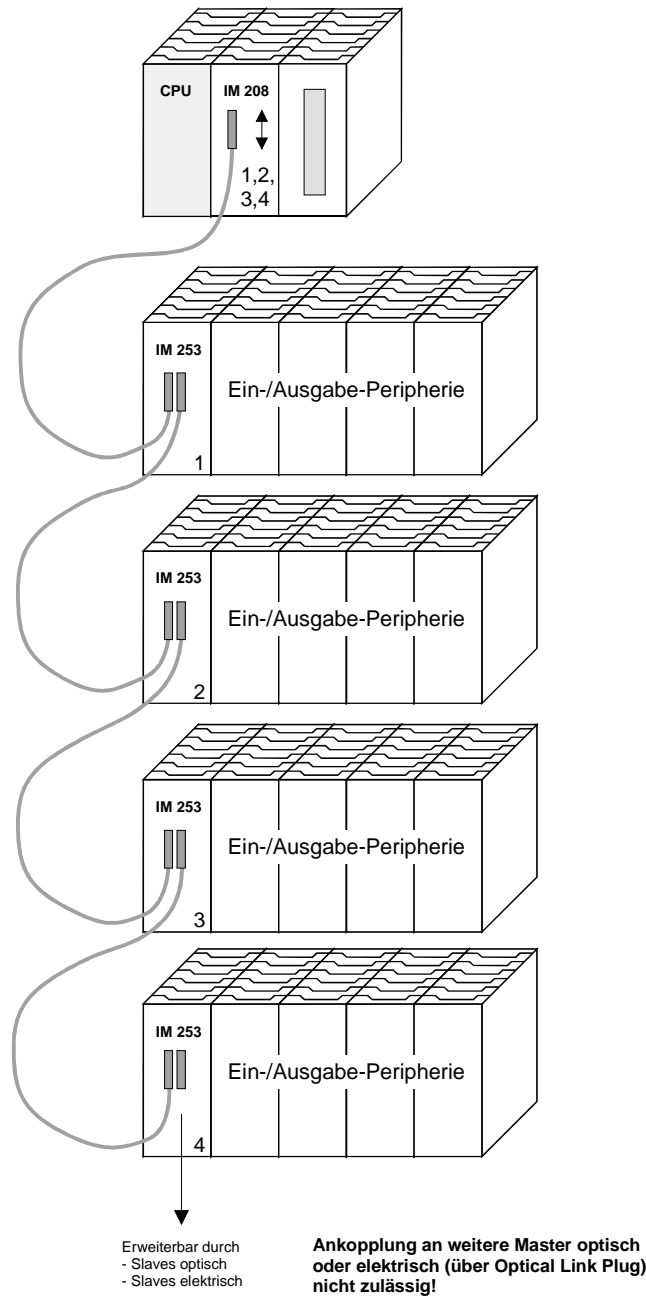


**Multi Master System**

Mehrere Master-Anschaltungen an einem Bus zusammen mit mehreren Slaves:



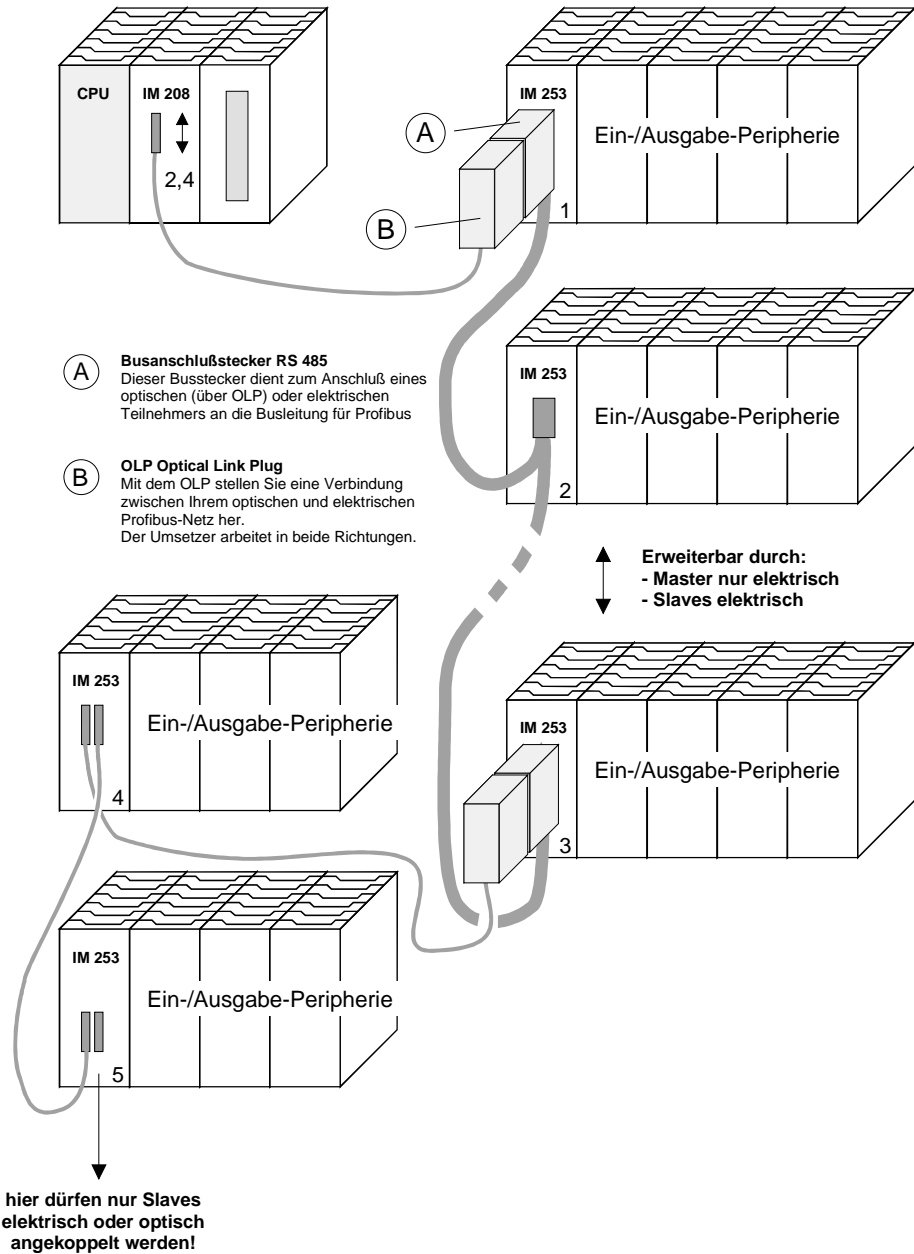
Optischer Profibus





**Gemischt  
optischer und  
elektrischer  
Profibus**

Bei einem gemischt optischen Profibus darf sich immer nur eine Umsetzung (OLP) zwischen zwei Mastern befinden!



## Inbetriebnahme

### Übersicht

- Bauen Sie Ihr Profibus-System auf.
- Projektieren Sie Ihr Mastersystem.
- Transferieren Sie Ihr Projekt in den Master.
- Verbinden Sie die Master- und Slave-Module mit dem Profibus.
- Schalten Sie die Spannungsversorgung ein.

### Aufbau

Bauen Sie Ihr Profibus-System mit den gewünschten Peripherie-Modulen auf.

Jeder VIPA Profibus-Slave-Koppler besitzt ein eingebautes Netzteil. Das Netzteil ist mit 24V Gleichspannung zu versorgen. Über die Spannungsversorgung werden neben dem Buskoppler auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus mit maximal 3A versorgen kann.

Profibus und Rückwandbus sind galvanisch voneinander getrennt.

### Adressierung

Stellen Sie an den Profibus-Slave-Modulen die entsprechende Profibus-Adresse ein.

### Projektierung im Mastersystem

Projektieren Sie Ihre Profibus-Master in Ihrem Master-System. Für die Projektierung können Sie WinNCS von VIPA einsetzen.

### Projekt transferieren

Aufgrund unterschiedlicher Hardwarevarianten gibt es bei den Profibus-Master-Modulen von VIPA verschiedene Transfermethoden, die in der Masterprojektierung der jeweiligen Hardwarevarianten näher beschrieben sind.

### System mit Profibus verbinden

In Systemen mit mehr als einer Station werden alle Teilnehmer parallel verdrahtet. Hierzu ist das Buskabel unterbrechungsfrei durchzuschleifen.

**Achten Sie hierbei immer auf richtige Polarität!**

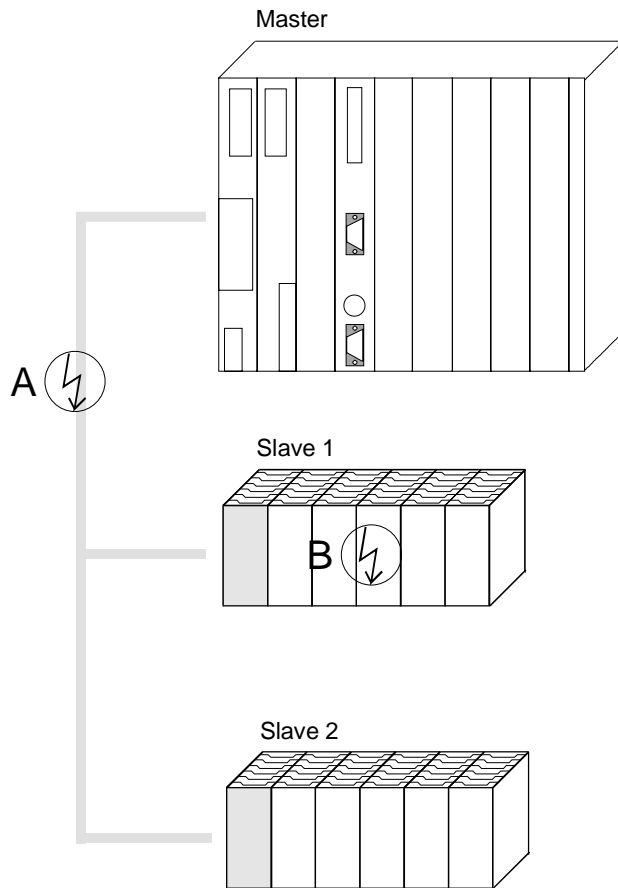


### Hinweis!

An den Leitungsenden muss das Buskabel immer mit dem Wellenwiderstand abgeschlossen werden um Reflexionen und damit Übertragungsprobleme zu vermeiden!

## Einsatz der Diagnose-LEDs

Das folgende Beispiel zeigt die Reaktion der LEDs bei unterschiedlichen Netzwerkunterbrechungen.



### Unterbrechung Position A

Der Profibus ist unterbrochen.

### Unterbrechung Position B

Die Kommunikation über den Rückwandbus ist unterbrochen.

LED Slave 1	Unterbrechung Position	
LED	A	B
RD	blinkt	aus
ER	aus	an
DE	aus	aus

LED Slave 2	Unterbrechung Position	
LED	A	B
RD	blinkt	an
ER	aus	aus
DE	aus	an

## Beispiele zur Profibus-Kommunikation

### Beispiel 1

#### Aufgabenstellung

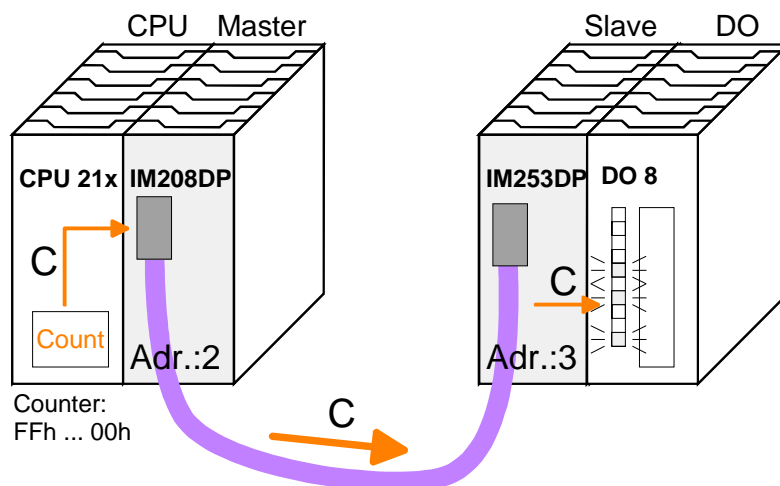
In diesem Beispiel soll eine Kommunikation zwischen einem Master- und einem Slave-System gezeigt werden.

Das Master-System besteht aus einer CPU 21x und einem DP-Master IM 208DP. Dieses System kommuniziert über Profibus mit einem IM 253DP und einem Ausgabe-Modul.

Über dieses System sollen Zählerstände über Profibus ausgetauscht und auf dem Ausgabe-Modul dargestellt werden. Die Zählerstände sind in der CPU zu generieren.

#### Aufgabenstellung im Detail

Die CPU soll von FFh ... 00h zählen und den Zählerstand zyklisch in den Ausgabebereich des Profibus-Masters übertragen. Der Master hat diesen Wert an den DP-Slave zu schicken. Der empfangene Wert soll auf dem Ausgabe-Modul (auf Adresse 0) ausgegeben werden.



#### Projektierdaten

##### CPU und IM 208DP (Master)

Zählerstand: MB 0 (FFh ... 00h)

Profibus-Adresse: 2

##### IM 253DP und DO (Slave)

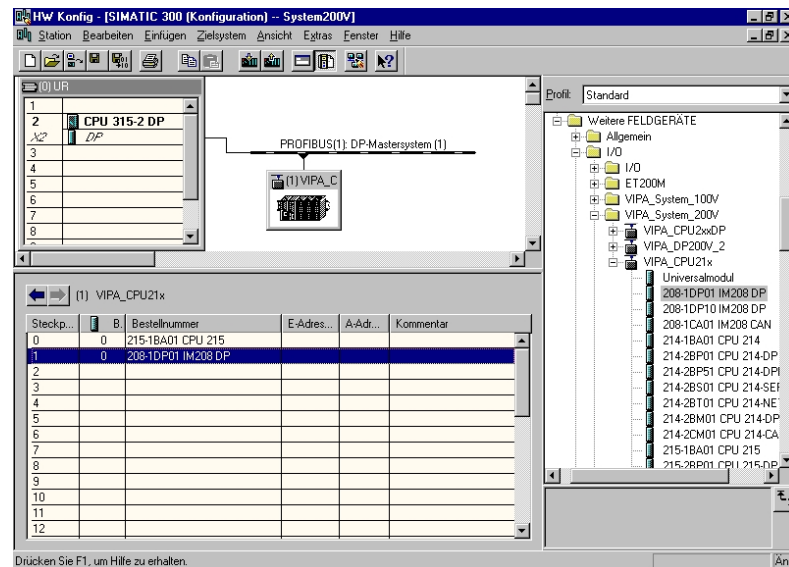
Profibus-Adresse: 3

Ausgangsbereich: Adresse 0, Länge: 1 Byte

## Projektierung IM 208DP

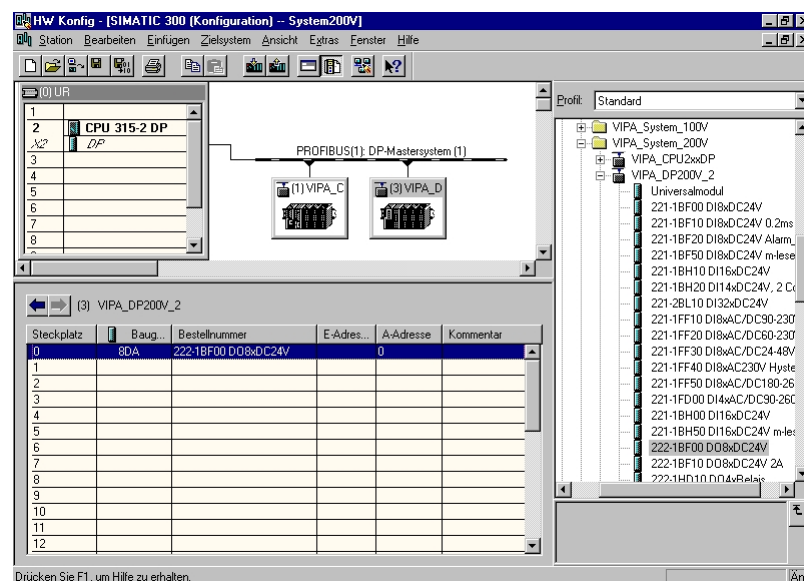
Um kompatibel mit dem STEP<sup>®</sup>7 Projektierool von Siemens zu sein, sind für das System 200V folgende Schritte durchzuführen:

- Starten Sie den Hardware-Konfigurator von Siemens
- Projektieren Sie eine CPU 315-2DP mit DP-Maste (Master-Adresse 2)
- Fügen Sie einen "virtuellen" Profibus-Slave "VIPA\_CPU21x" mit Adresse 1 an.
- Binden Sie auf Steckplatz 0 des Slave-Systems die CPU 21x ein.
- Binden Sie auf Steckplatz 1 den DP-Master 208-1DP01 ein.



Zur Ankopplung Ihres realen DP-Slaves IM 253DP sind, nachdem Sie die GSD-Datei eingebunden haben, folgende Schritte erforderlich:

- Fügen Sie den Profibus-Slave "VIPA\_DP200V\_2" mit Adresse 3 an.
- Binden Sie auf Steckplatz 0 das Digitale Ausgabe-Modul 222-1BF00 ein.
- Geben Sie die Ausgabe-Adresse 0 an.



**Anwender-  
programm in CPU**

Für das Anwenderprogramm in der CPU verwenden wir den OB35. Der OB35 ist ein Zeit-OB, dessen Aufrufzyklus Sie in den CPU-Eigenschaften einstellen können.

**OB 35 (Zeit-OB)**

```
L   MB  0      Zähler von FFh bis 00h
L   L    1
-I
T   MB  0      neuen Zählerstand merken
T   AB  0      neuen Zählerstand an Ausgabe-Byte 0 via
                    Profibus übertragen
BE
```

Den Aufrufzyklus des OB35 können Sie in den "Eigenschaften" Ihrer CPU 315-2DP unter *Weckalarm* einstellen. Geben Sie hier beispielsweise 100ms an.

**Projekt trans-  
ferieren und  
ausführen**

Die Programmierung ist jetzt abgeschlossen. Übertragen Sie Ihr Projekt in die CPU und führen Sie Ihr Programm aus.

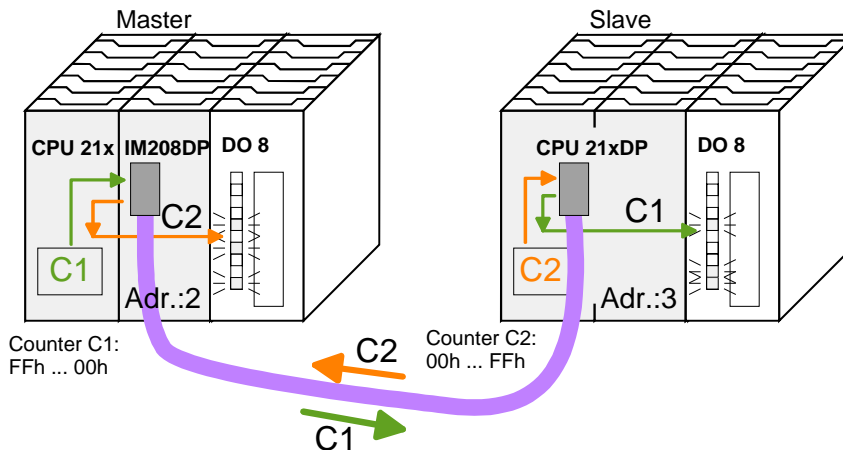
**Beispiel 2**

**Aufgabenstellung**

In diesem Beispiel soll eine Kommunikation zwischen einer CPU 21x mit IM 208 DP-Master und einer CPU 21xDP (DP-Slave) gezeigt werden. Hierbei sollen Zählerstände über den Profibus ausgetauscht und diese auf dem Ausgabe-Modul des jeweiligen Partners dargestellt werden.

**Aufgabenstellung im Detail**

Die CPU 21x soll von FFh ... 00h zählen und den Zählerstand zyklisch in den Ausgabebereich des Profibus-Masters übertragen. Der Master hat diesen Wert an den Slave der CPU 21xDP zu schicken. Der empfangene Wert soll in der CPU im Eingangs-Peripheriebereich abgelegt und über den Rückwandbus auf dem Ausgabe-Modul (auf Adresse 0) ausgegeben werden. Umgekehrt soll die CPU 21xDP von 00h bis FFh zählen. Auch dieser Zählerstand ist im Ausgabe-Bereich des CPU-Slaves abzulegen und über den Profibus in den Master zu transferieren. Dieser Wert ist auf dem Ausgabe-Modul (Adresse 0) der CPU 21x auszugeben.



**Projektierdaten**

**CPU 21x und DP-Master**

Zählerstand: MB 0 (FFh ... 00h)  
 Profibus-Adresse: 2  
 Eingangsbereich: Adresse 10      Länge: 2 Byte  
 Ausgangsbereich: Adresse 20      Länge: 2 Byte

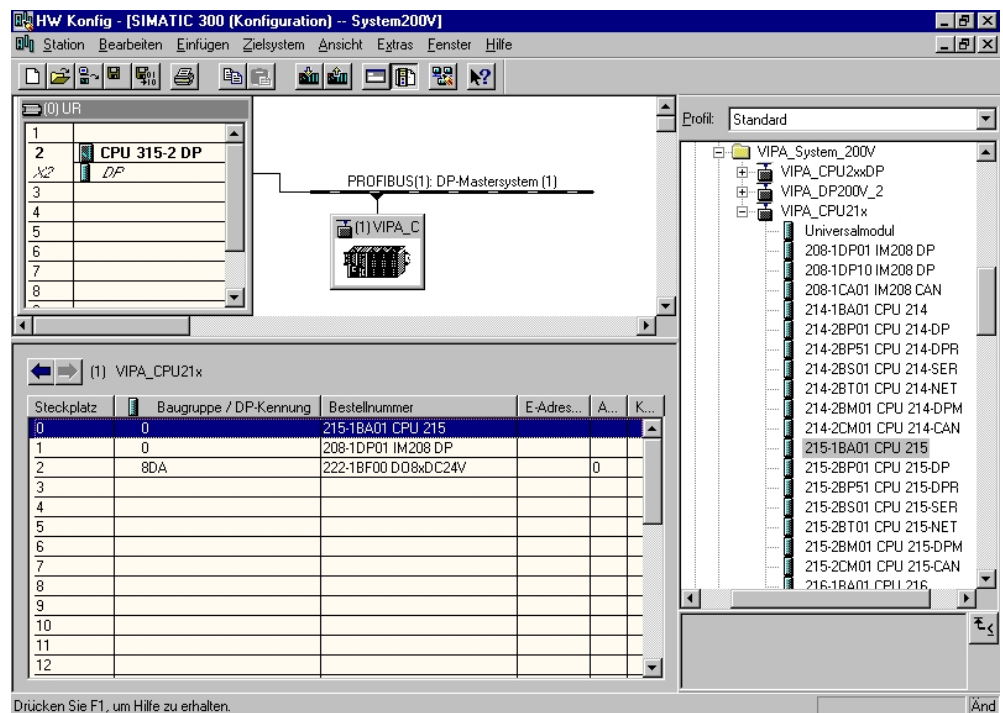
**CPU 21xDP**

Zählerstand: MB 0 (00h...FFh)  
 Eingangsbereich: Adresse 30      Länge: 2 Byte  
 Ausgangsbereich: Adresse 40      Länge: 2 Byte  
 Parameterdaten: Adresse 800      Länge: 24 Byte (fest)  
 Diagnosedaten: Adresse 900      Länge: 6 Byte (fest)  
 Statusdaten: Adresse 1020      Länge: 2 Byte (fest)  
 Profibus-Adresse: 3

### Projektierung CPU 21x des DP-Masters

Um kompatibel mit dem STEP®7 Projektierool von Siemens zu sein, sind für die CPU 21x und DP-Master folgende Schritte durchzuführen:

- Starten Sie den Hardware-Konfigurator von Siemens.
- Projektieren Sie eine CPU 315-2DP mit DP-Master-System (Adresse 2)
- Fügen Sie einen "virtuellen" Profibus-Slave "**VIPA\_CPU21x**" mit Adresse 1 an.
- Binden Sie auf Steckplatz 0 des Slave-Systems eine CPU 21x-1BA01 ein (wie beispielsweise die CPU 215-1BA01).
- Binden Sie auf Steckplatz 1 den DP-Master 208-1DP01 ein
- Binden Sie auf Steckplatz 2 das Ausgabe-Modul 222-1BF00 ein.



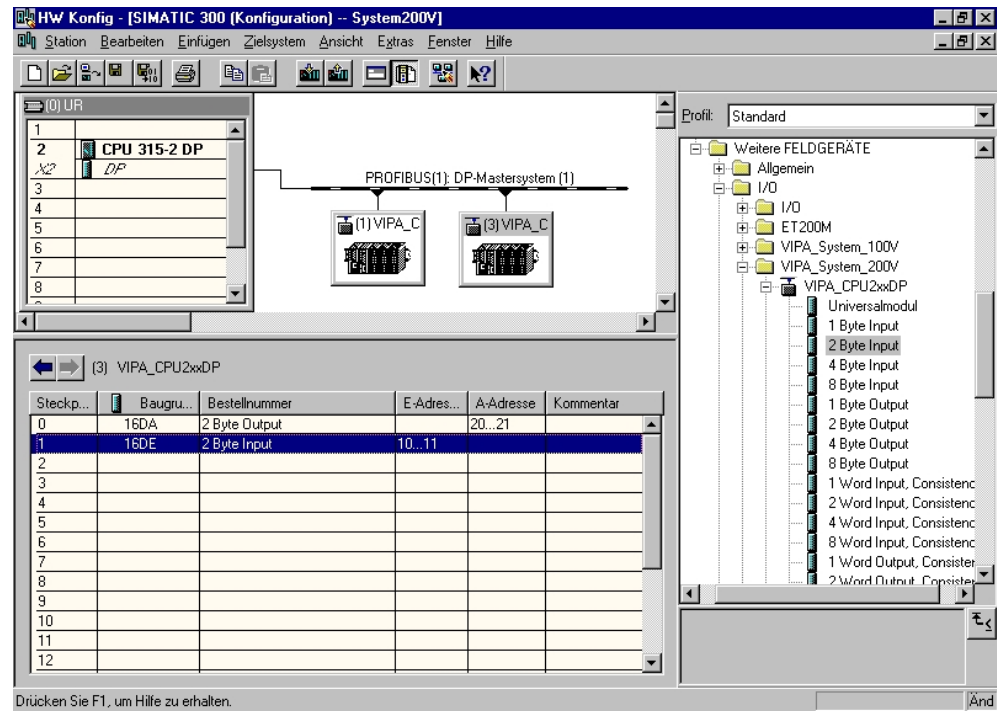
### Profibus-Ankopplung der CPU 21xDP

Zur Ankopplung Ihrer CPU 21xDP sind, nachdem Sie die GSD-Datei eingebunden haben, folgende Schritte erforderlich:

- Fügen Sie den Profibus-Slave "**VIPA\_CPU2xxDP**" an (Adresse 3)
- Binden Sie auf Steckplatz 0 das "2 Byte Output" Element ein und stellen Sie die Ausgabe-Adresse 20 ein
- Binden Sie auf Steckplatz 1 das "2 Byte Input" Element ein und stellen Sie die Eingabe-Adresse 10 ein
- Speichern Sie Ihr Projekt

Die Hardwarekonfiguration für CPU 21x und DP-Master ist jetzt abgeschlossen. Auf der Folgeseite ist die Projektierung nochmals dargestellt.





### Anwenderprogramm in CPU 21x des DP-Masters

Das Anwenderprogramm in der CPU 21x hat zwei Aufgaben, die auf zwei OBs verteilt werden:

- Über Kontrollbyte die Kommunikation testen.  
Vom Profibus das Eingangs-Byte laden und den Wert auf dem Ausgabe-Modul ausgeben.

#### OB 1 (zyklischer Aufruf)

```

L   B#16#FF
T   AB 20           Kontrollbyte für Slave-CPU

L   B#16#FE           Kontrollwert 0xFE laden
L   EB 10           Wurde Kontrollbyte von der Slave
<>I           CPU richtig übermittelt?
BEB           Nein -> Ende

-----
L   EB 11           Datenaustausch via Profibus
T   AB 0           Lade Eingangsbyte 11 (Ausgangsdaten
                   der CPU214DP) und
                   transferiere ins Ausgangsbyte 0

BE

```

- Zählerstand aus dem MB 0 lesen, dekrementieren, in MB 0 speichern und über Profibus an CPU 21xDP ausgeben.

#### OB 35 (Zeit-OB)

```

L   MB 0           Zähler von 0xFF bis 0x00
L   1
-I
T   MB 0
T   AB 21           Transferiere ins Ausgangsbyte 21
                   (Eingangsdaten der CPU214DP)

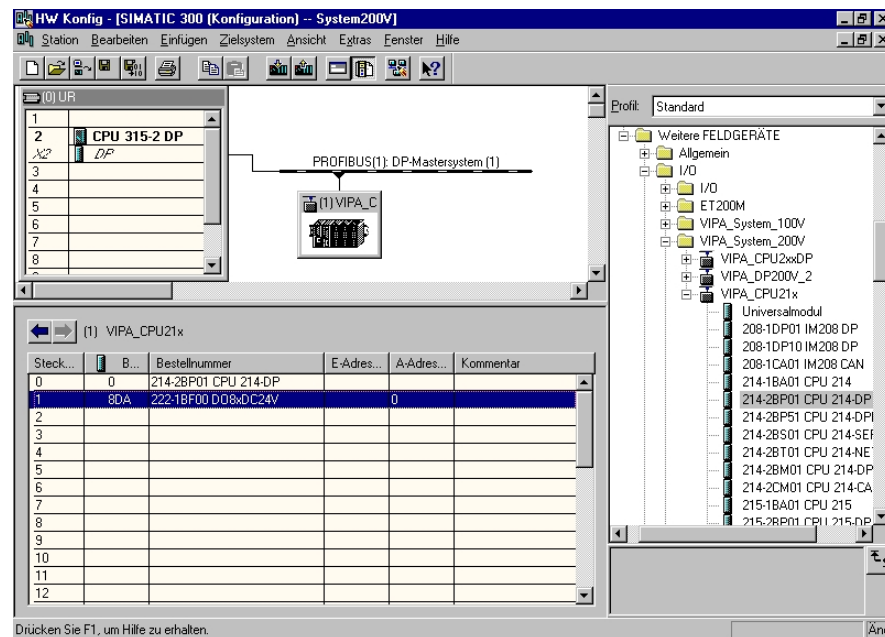
BE

```

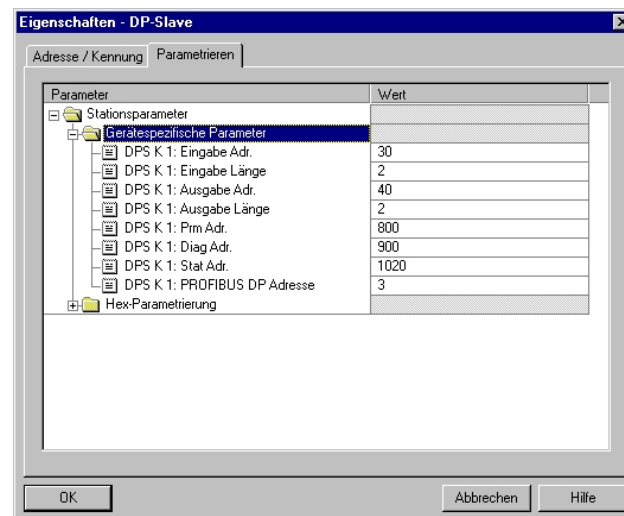
**Projektierung CPU 21xDP**

Um kompatibel mit dem STEP®7 Projektierool von Siemens zu sein, sind für die CPU 21xDP folgende Schritte durchzuführen:

- Starten Sie den Hardware-Konfigurator von Siemens
- Projektieren Sie eine CPU 315-2DP mit DP-Master-System (Adresse 2)
- Fügen Sie einen "virtuellen" Profibus-Slave "VIPA\_CPU21x" mit Adresse 1 an
- Binden Sie auf Steckplatz 0 des Slave-Systems eine 21x-2BP01-CPU ein (wie beispielsweise die CPU 214-2BP01).
- Binden Sie auf Steckplatz 1 das Ausgabe-Modul 222-1BF00 ein



- Stellen Sie folgende Parameter im Parametrierfenster der CPU 214-2BP01 ein:



- Speichern Sie Ihr Projekt

**Anwender-  
programm in  
CPU 21xDP**

Das Anwenderprogramm hat wie schon weiter oben gezeigt zwei Aufgaben, die auch bei dieser CPU auf zwei OBs verteilt werden:

- Vom Profibus-Slave das Eingangs-Byte laden und den Wert auf dem Ausgabe-Modul ausgeben.

**OB 1 (zyklischer Aufruf)**

```

L   PEW 100           Statusdaten laden und in Merker-
T   MW 100           wort speichern

UN  M 100.5          Inbetriebnahme durch DP-Master
BEB                               erfolgt? Nein -> Ende

U   M 101.4          Empfangsdaten gültig?
BEB                               Nein -> Ende
L   B#16#FF          Kontrollwert laden und mit
L   PEB 30           Kontrollbyte (1. Eingangsbyte)
<>I                               vergleichen
BEB                               Empfangene Daten haben keine
                                   gültigen Werte

L   B#16#FE          Kontrollbyte für Master-CPU
T   PAB 40           -----
                                   Datenaustausch via Profibus

L   PEB 31           Lade Peripheriebyte 31 (Eingangs-
T   AB 0             daten vom Profibus-Slave) und
                                   transferiere ins Ausgangsbyte 0

BE

```

- Zählerstand aus dem MB 0 lesen, inkrementieren, in MB 0 speichern und über Profibus an CPU 21x ausgeben.

**OB 35 (Zeit-OB)**

```

L   MB 0             Zähler von 0x00 bis 0xFF
L   1
+I
T   MB 0

T   PAB 41          Transferiere Zählerwert ins
                                   Peripheriebyte 41 (Ausgangsdaten
                                   des Profibus-Slaves)

BE

```

## Technische Daten

### Profibus-DP- Master

#### IM 208DP

Elektrische Daten	VIPA 208-1DP01
Spannungsversorgung	über Rückwandbus
Stromaufnahme	max. 380mA
Potenzialtrennung	≥ AC 500V
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse/Schnittstellen	9polige SubD-Buchse      Profibus-Ankopplung
Profibus Schnittstelle	
Ankopplung	9polige SubD-Buchse
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden, Stichleitungen sind möglich.
Medium	Abgeschirmtes verdrilltes Twisted Pair-Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen.
Übertragungsrate	9,6kBaud bis 12MBaud
Gesamtlänge	ohne Repeater 100m bei 12MBaud; mit Repeater bis 1000m
max. Teilnehmeranzahl	32 Stationen in jedem Segment ohne Repeater. Mit Repeater erweiterbar auf 126.
Kombination mit Peripheriemodulen	
max. Anzahl Slaves	125
max. Anzahl Eingangs-Byte	256 (ab V3.0.0: 1024Byte)
max. Anzahl Ausgangs-Byte	256 (ab V3.0.0: 1024Byte)
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	110g

**IM 208DPO**

Elektrische Daten	VIPA 208-2DP10
Spannungsversorgung	über Rückwandbus
Stromaufnahme	max. 380mA
Potentialtrennung	≥ AC 500V
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse/Schnittstellen	2polige Lichtwellenleiter-Buchse Profibus-Ankopplung
Profibus Schnittstelle	
Ankopplung	2polige Lichtwellenleiter-Buchse
Netzwerk Topologie	Linienstruktur mit LWL-Zweileiter, Busabschluss am Ende nicht erforderlich
Medium	Lichtwellen-Zweileiter-Kabel
Übertragungsrate	12MBaud
Gesamtlänge	bei POF-LWL: max. 50m zwischen den Teilnehmern bei HCS-LWL: max. 300m zwischen den Teilnehmern
max. Teilnehmeranzahl	17 Teilnehmer inkl. Master.
Kombination mit Peripheriemodulen	
max. Anzahl Slaves	16
max. Anzahl Eingangs-Byte	256 (ab V3.0.0: 1024Byte)
max. Anzahl Ausgangs-Byte	256 (ab V3.0.0: 1024Byte)
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x76
Gewicht	110g

**Maximale Anzahl der Teilnehmer**

Die maximale Anzahl der DPO-Teilnehmer ist von der verwendeten Baudrate abhängig. Die maximale Anzahl inkl. Master können Sie der nachfolgenden Tabelle entnehmen:

Baudrate	max. Teilnehmeranzahl
≤ 1,5MBaud	17
3MBaud	15
6MBaud	7
12MBaud	4

## Profibus-DP- Slave (Standard)

### IM 253DP

Elektrische Daten	VIPA 253-1DP00
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme	max. 1A
Ausgangsstrom Rückwandbus	max. 3,5A
Potenzialtrennung	≥ AC 500V
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse/Schnittstellen	9polige SubD-Buchse          Profibus-Ankopplung
Profibus Schnittstelle	
Ankopplung	9polige SubD-Buchse
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden, Stichleitungen sind möglich.
Medium	Abgeschirmtes verdrehtes Twisted Pair-Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen.
Übertragungsrate	9,6kBaude bis 12Mbaude (automatische Einstellung)
Gesamtlänge	ohne Repeater 100m bei 12Mbaude; mit Repeater bis 1000m
max. Teilnehmeranzahl	32 Stationen in jedem Segment ohne Repeater. Mit Repeater erweiterbar auf 126.
Diagnosefunktionen	
Standard-Diagnose	Speicherung der letzten 100 Diagnosen mit Zeitstempel im Flash-ROM. Auslesen mit Spezialtool und Kabel möglich.
Erweiterte Diagnosemöglichkeit	-
Kombination mit Peripheriemodulen	
max. Modulanzahl	32 (abhängig von der Stromaufnahme)
max. digital	32
max. analog	16
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g

**IM 253DPO**

Elektrische Daten	VIPA 253-1DP10
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme	max. 1A
Ausgangsstrom Rückwandbus	max. 3,5A
Potenzialtrennung	≥ AC 500V
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse/Schnittstellen	4polige Lichtwellenleiter-Buchse Profibus-Ankopplung
Profibus Schnittstelle	
Ankopplung	4polige Lichtwellenleiter-Buchse
Netzwerk Topologie	Linienstruktur mit LWL-Zweileiter, Busabschluss am Ende nicht erforderlich
Medium	Lichtwellen-Zweileiter-Kabel
Übertragungsrate	12MBaud
Gesamtlänge	bei POF-LWL: max. 50m zwischen den Teilnehmern bei HCS-LWL: max. 300m zwischen den Teilnehmern
max. Teilnehmeranzahl	17 Teilnehmer inkl. Master (siehe unten)
Diagnosefunktionen	
Standard-Diagnose	Speicherung der letzten 100 Diagnosen mit Zeitstempel im Flash-ROM. Auslesen mit Spezialtool und Kabel möglich.
Erweiterte Diagnosemöglichkeit	-
Kombination mit Peripheriemodulen	
max. Modulanzahl	32 (abhängig von der Stromaufnahme)
max. digital	32
max. analog	16
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g

**Maximale Anzahl der Teilnehmer**

Die maximale Anzahl der DPO-Teilnehmer ist von der verwendeten Baudrate abhängig. Die maximale Anzahl inkl. Master können Sie der nachfolgenden Tabelle entnehmen:

Baudrate	max. Teilnehmeranzahl
≤ 1,5MBaud	17
3MBaud	15
6MBaud	7
12MBaud	4

### Profibus-DP-Slave (redundant)

#### IM 253DPR

Elektrische Daten	VIPA 253-2DP50
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme	max. 1A
Ausgangsstrom Rückwandbus	max. 3,5A
Potenzialtrennung	≥ AC 500V
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse/Schnittstellen	9polige SubD-Buchse (2x) Profibus-Ankopplung
2 Kanäle	DP1 / DP2
Profibus Schnittstelle	
Ankopplung	9polige SubD-Buchse (2x)
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden, Stichleitungen sind möglich.
Medium	Abgeschirmtes verdrehtes Twisted Pair-Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen.
Übertragungsrate	9,6kBaude bis 12MBaude (automatische Einstellung)
Gesamtlänge	ohne Repeater 100m bei 12MBaude; mit Repeater bis 1000m
max. Teilnehmeranzahl	32 Stationen in jedem Segment ohne Repeater. Mit Repeater erweiterbar auf 126.
Diagnosefunktionen	
Standard-Diagnose	Speicherung der letzten 100 Diagnosen mit Zeitstempel im Flash-ROM. Auslesen mit Spezialtool und Kabel möglich.
Erweiterte Diagnosemöglichkeit	-
Kombination mit Peripheriemodulen	
max. Modulanzahl	32 (abhängig von der Stromaufnahme)
max. digital	32
max. analog	16
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x76
Gewicht	120g



## Profibus-DP- Slave (Kombi-Modul)

### IM 253DP DO 24xDC 24V

Elektrische Daten	VIPA 253-2DP20
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme	max. 5A
Ausgangsstrom Rückwandbus	max. 3,5A
Potenzialtrennung	≥ AC 500V
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse/Schnittstellen	9polige SubD-Buchse Profibus-Ankopplung
Profibus Schnittstelle	
Ankopplung	9-polige SubD-Buchse,
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden
Medium	Abgeschirmtes verdrehtes Twisted Pair-Kabel, Schirmung darf abhängig von Umgebungsbedingungen entfallen.
Übertragungsrate	9,6kBaud bis 12MBaud (automatische Einstellung)
Gesamtlänge	ohne Repeater 100m bei 12MBaud; mit Repeater bis 1000m
max. Teilnehmeranzahl	32 Stationen in jedem Segment ohne Repeater. Mit Repeater max. 126.
Statusanzeige	über LEDs auf der Frontseite
Kombination mit Peripheriemodulen	
max. Modulanzahl	31 (abhängig von der Stromaufnahme)
max. digital E/A	31
max. analog E/A	16
Ausgabe-Einheit	
Anzahl der Ausgänge	24
Nennlastspannung	DC 24V (18...35V) intern über Profibus-Koppler versorgt
Ausgangsstrom je Kanal	1A (Summenstrom max. 4A)
Statusanzeige	Power (PW) Sicherung intakt, Error (ER) Kurzschluss, Überlast
Programmierdaten	
Ausgabedaten	4Byte (3Byte benutzt)
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x76
Gewicht	150g



## Teil 3 Interbus

### Überblick

In diesem Kapitel erhalten Sie alle Informationen, die zur Anbindung Ihrer System 200V Peripherie an Interbus erforderlich sind.

Nach den Interbus-Grundlagen folgt die Beschreibung des Interbus-Kopplers, dessen Inbetriebnahme und Parametrierung.

Die Technischen Daten finden Sie am Ende des Kapitels.

Nachfolgend ist beschrieben:

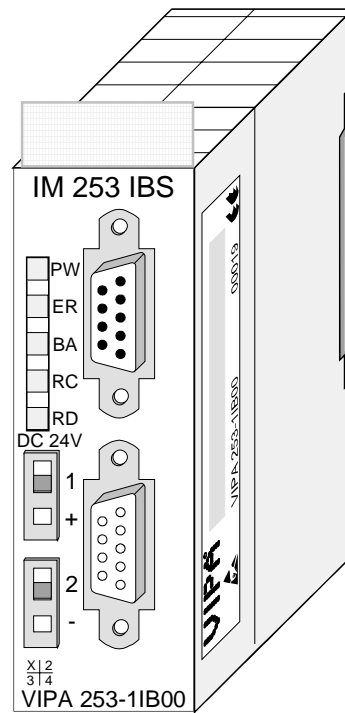
- Systemübersicht und Interbus-Grundlagen
- Hardwareaufbau, Einsatz und Inbetriebnahme des Interbus-Koppler
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 3 Interbus .....</b>	<b>3-1</b>
Systemübersicht .....	3-2
Grundlagen.....	3-3
IM 253IBS - Interbus-Koppler - Aufbau .....	3-7
Anschluss an Interbus .....	3-10
Einsatz im Interbus .....	3-11
Inbetriebnahme.....	3-15
Technische Daten.....	3-18

## Systemübersicht

Mit dem Interbus-Slave von VIPA können Sie bis zu 16 Eingangs- und 16 Ausgangsmodule des System 200V in Ihren Interbus einbinden.  
Zur Zeit ist ein Interbus-Slave-Modul von VIPA verfügbar.



### Bestelldaten

Bestellnummer	Beschreibung
VIPA 253-1IB00	Interbus Slave

## Grundlagen

- Allgemeines** Interbus ist ein reines Master/Slave System, welches aufgrund seines geringen Protokolloverheads speziell auf den Sensor-/Aktor-Bereich zugeschnitten ist. Interbus wurde Mitte der 80er Jahre gemeinsam von PHOENIX CONTACT, digital Equipment und der Fachhochschule Lemgo entwickelt, erste Systemkomponenten waren 1988 verfügbar. Bis heute ist das Übertragungsprotokoll praktisch unverändert, so dass auch Geräte der ersten Generation mit den aktuellen Masteranschlüssen (Generation 4) betrieben werden können.
- Interbus für Sensor- und Aktorbereich** Die breite Anwendung im Sensor/Aktor-Bereich ist nicht zuletzt auf eine relativ einfache Schnittstellenimplementierung durch fertige Protokollchips zurückzuführen, welche die direkte Anbindung von Ein- und Ausgabepunkten mit nur wenigen externen Bauteilen ermöglichen.  
Für Interbus-Teilnehmer ist die DIN-Norm 19258 maßgeblich, welche unter anderem die Schichten 1 und 2 des Protokolls beschreibt.
- Interbus als Schieberegister** Das Interbus-System ist als Datenring mit einem zentralen Master-Slave-Zugriffsverfahren aufgebaut. Es hat die Struktur eines räumlich verteilten Schieberegisters. Jedes Gerät ist mit seinen Registern unterschiedlicher Länge ein Teil dieses Schieberegisterrings, durch den die Daten seriell vom Master aus hindurch geschoben werden. Die Verwendung der Ringstruktur bietet dabei die Möglichkeit des zeitgleichen Sendens und Empfangens von Daten. Beide Datenrichtungen des Rings sind in einem Kabel untergebracht.
- ID-Register** Jeder Teilnehmer im Interbus hat ein ID-Register (Identifikations-Register). In diesem Register befinden sich Informationen über den Modultyp, die Anzahl der Ein- und Ausgangsregister sowie Status- und Fehlerzustände.
- Interbus-Master** Mit dem Interbus-Koppler können die Peripheriebaugruppen des Systems 200V über Interbus gesteuert werden. Der Buskoppler ersetzt in diesem Fall die CPU. Das Lesen und Schreiben der Ein- bzw. Ausgänge erfolgt durch den Interbus-Master. Der Master ist das Bindeglied zu anderen Systemen. Ein Master kann bis zu 4096 Ein-/Ausgabepunkte verwalten. Diese können entweder direkt im Hauptstrang liegen, oder über Buskoppler auf untergeordnete Strukturen verteilt sein.  
An dem vom Master ausgehenden Hauptring können zur Strukturierung des Gesamtsystems Subringsysteme angeschlossen werden. Über Busklemmen erfolgt die Ankopplung solcher Subringsysteme. Über diese Busklemmen können Sie auch Teilnehmer über große Distanzen ankoppeln.

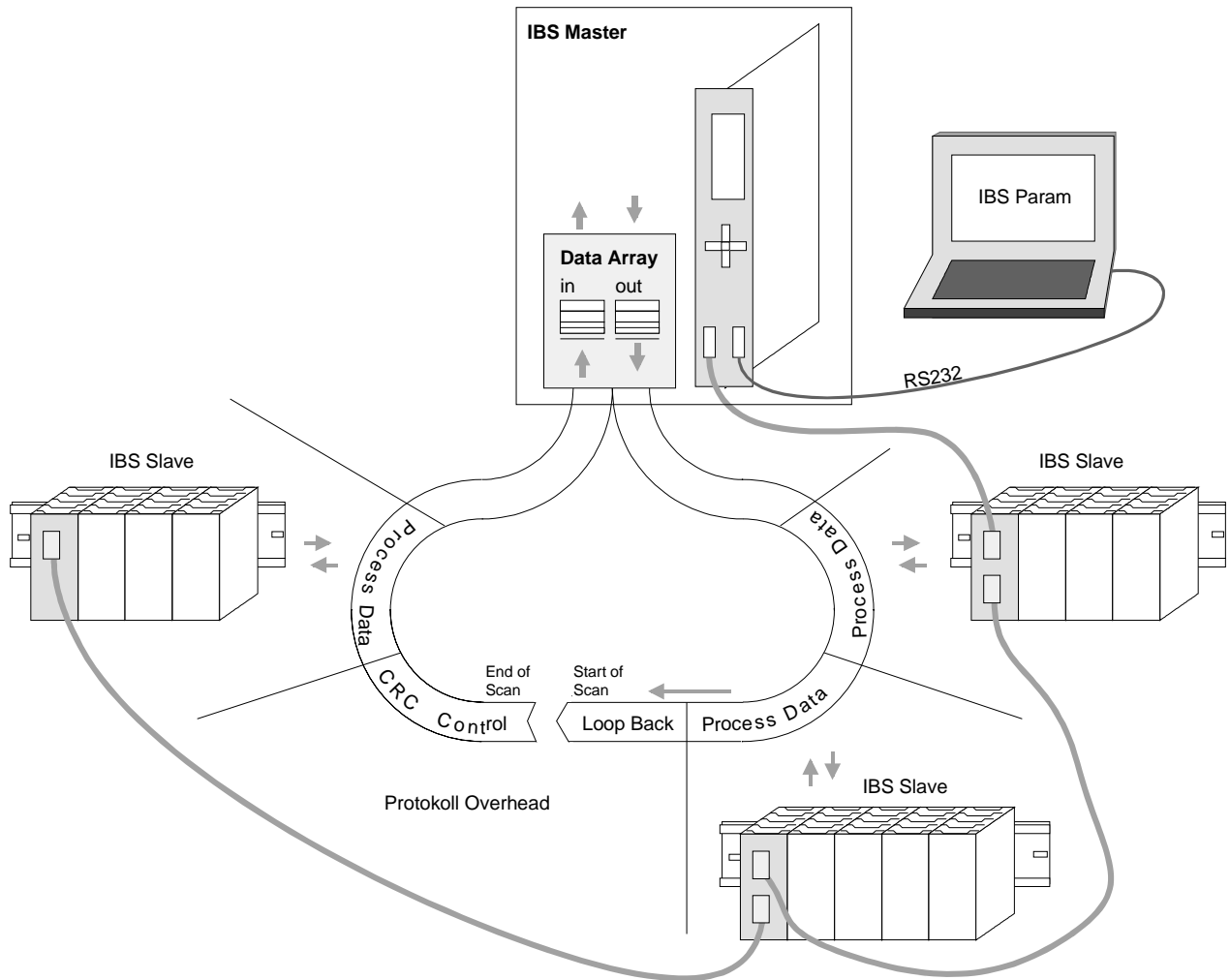
<b>Beschränkung der Datenbreite</b>	<p>Mit zunehmender Datenbreite steigt der Hardwareaufwand für einen Interbus-Teilnehmer. Aus diesem Grund wurde die Datenbreite auf max. 20Byte Ein- und 20Byte Ausgangsdaten beschränkt.</p> <p>Untergeordnete Interbus-Segmente (Peripheriebus) können über die zugehörigen Buskoppler zu- oder abgeschaltet werden, so dass beispielsweise bei einem Defekt in einem Peripheriebusabzweig der Bus weiterbetrieben werden kann, indem das entsprechende Segment aus dem Bus geschaltet wird.</p>
<b>Betriebsarten</b>	<p>Interbus hat zwei Betriebsarten:</p> <ul style="list-style-type: none"><li>• ID-Zyklus Der ID-Zyklus wird zur Initialisierung des Interbus-Systems und auf Anforderung durchgeführt. Im ID-Zyklus liest der Bus-Master von allen Teilnehmern am Bussystem die ID-Register aus und baut anhand dieser Informationen das Prozessabbild auf.</li><li>• Datenzyklus Der Datenzyklus wickelt die eigentliche Datenübertragung ab. Im Datenzyklus werden von allen Geräten die Eingabedaten aus den Registern in den Master und Ausgabedaten vom Master an die Geräte übertragen. Die Datenübertragung erfolgt vorduplex.</li></ul>
<b>Übertragungsmedium</b>	<p>Auch wenn Interbus rein äußerlich als Linienstruktur ausgeführt wird (nur ein Leitungszug vom Master bis zum letzten Modul), handelt es sich im Grunde um eine Ringstruktur, bei der Hin- und Rückleiter in einer Leitung untergebracht sind. Der Ring wird durch den letzten Teilnehmer geschlossen. Bei den meisten Geräten geschieht dies automatisch, sobald keine weiterführende Leitung angeschlossen ist.</p> <p>Die physikalische Ebene des Interbus wird mit dem RS422-Standard realisiert. Zur Übertragung der Signale werden verdrehte Twisted Pair Leitungen eingesetzt. Bei Interbus wird die Datenhin- und die Datenrückleitung innerhalb eines Kabels und durch jeden Teilnehmer geführt. Für die Kommunikation sind aufgrund dieser Ringstruktur und des gemeinsamen Logic-grounds zwischen 2 Teilnehmern 5 Adern erforderlich. Bei einer Datenübertragungsrate von 500kbaud können 2 benachbarte Ringteilnehmer bis zu 400m entfernt sein. Durch die integrierte Repeaterfunktion in jedem Teilnehmer erreichen Sie eine Gesamtausdehnung von bis zu 13km. Die Gesamtanzahl ist auf maximal 512 Teilnehmer begrenzt.</p>

**Prozessdaten übertragen**

Interbus basiert auf einem physikalischen Ring, der als zyklisch getaktetes Ringschieberegister arbeitet. Jeder Interbus-Teilnehmer fügt sich hierbei mit einem Schieberegisterbereich, dessen Länge durch die Anzahl der Prozessdatenpunkte des Teilnehmers festgelegt wird, in den Ring ein. Durch die Aneinanderkopplung aller Teilnehmer und Rückführung des letzten Schieberegisterausgangs auf den Busmaster, ergibt sich ein Ringschieberegister, dessen Länge und Struktur dem physikalischen Aufbau des Interbus Gesamtsystems entspricht.

Interbus arbeitet mit einem Master-Slave-Zugriffsverfahren, wobei der Bus-Master gleichzeitig die Kopplung an das überlagerte Steuerungssystem realisiert. Durch das Ringsystem sind alle Teilnehmer aktiv in einen, in sich geschlossenen Übertragungsweg eingekoppelt.

Im Gegensatz zu teilnehmerorientierten Busprotokollen, bei denen Daten nur dann ausgetauscht werden, wenn ein Teilnehmer einen entsprechenden, an ihn adressierten Befehl erhält, erfolgt die Datenübertragung im Interbus zyklisch in zeitäquidistanten Intervallen, wobei in jedem Datenzyklus alle Teilnehmer angesprochen werden.



**Steuer- u. Kontroll-  
informationen  
übertragen**

Neben den Prozessdaten werden zusätzlich Steuer- und Kontrollinformationen übertragen. Diese Zusatzinformationen werden in jedem Datenzyklus nur einmal vor, bzw. im Anschluss an die Prozessdaten übertragen, weshalb man auch von einem Summenrahmenverfahren spricht.

**Prinzip der  
Kommunikation**

Das Prinzip der Kommunikation ist unabhängig von der Art der übertragenen Daten:

Die Prozessdaten, die an die Peripherie ausgegeben werden sollen, sind entsprechend der physikalischen Reihenfolge der angeschlossenen Ausgabestationen im Ausgabebuffer des Masters hinterlegt. Ein Übertragungszyklus erfolgt nun dadurch, dass der Master das "Loopback-Wort" durch den Ring schiebt. Hinter dem Loopback-Wort werden nacheinander alle Ausgabedaten auf den Bus und damit durch das Schieberegister getaktet. Während diese Datenausgabe durchgeführt wird, erfolgt gleichzeitig der Rückfluss von Prozessinformationen als Eingabedaten in den Eingangspuffer des Masters.

Nachdem so das gesamte Summenrahmentelegramm ausgegeben und gleichzeitig wieder eingelesen wurde, sind alle Ausgabedaten in den Schieberegistern der einzelnen Teilnehmer richtig positioniert. Über ein spezielles Steuerkommando teilt der Master den Teilnehmern das Ende des Übertragungszyklus mit.

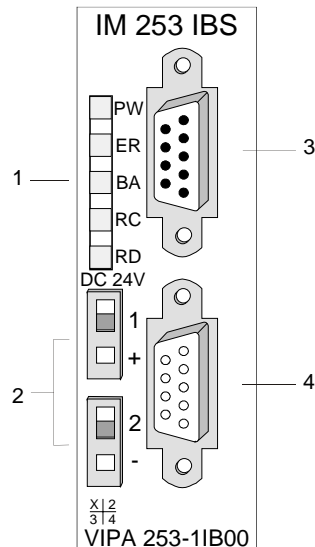
Nach der Durchführung einer Datensicherungssequenz werden dann die Prozessausgabeinformationen aus den Schieberegistern übernommen, in den Teilnehmern gespeichert und an die Peripherie weitergegeben. Gleichzeitig werden neue Peripherieinformationen in die Schieberegister der Eingabestationen eingelesen und somit der nächste Eingabezyklus vorbereitet. Der beschriebene Vorgang wird zyklisch wiederholt, so dass die Ein- und Ausgabebuffer des Masters zyklisch aktualisiert werden. Somit erfolgt die Datenübertragung im Interbus voll duplex, d.h. mit einem Datenzyklus werden sowohl Ausgangs- als auch Eingangswerte übertragen.

Durch die Schieberegisterstruktur entfällt die in anderen Feldbussystemen übliche Adresseinstellung der Teilnehmer, da die Adresse durch die Position des Teilnehmers im Ring eindeutig festgelegt ist.



## IM 253IBS - Interbus-Koppler - Aufbau

### Aufbau



- [1] LED Statusanzeigen
- [2] Spannungsversorgung Anschluss für externe 24V
- [3] Interbus-Stecker ankommende Schnittstelle
- [4] Interbus-Buchse weiterführende Schnittstelle

### Komponenten

#### LEDs

Das Modul besitzt verschiedene LEDs, die der Busdiagnose dienen. Die Verwendung und die jeweiligen Farben dieser Diagnose-LEDs finden Sie in der nachfolgenden Tabelle:

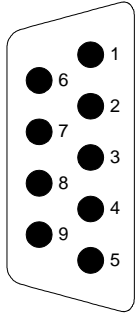
Bez.	Farbe	Bedeutung
PW	Grün	Power-LED Signalisiert eine anliegende Betriebsspannung.
ER	Rot	Error Fehler in der Applikation.
BA	Grün	Bus aktiv Mit Hilfe der BA-LED (bus active) wird ein Datentransfer über Interbus angezeigt.
RC	Grün	Remotebus Check Über die RC LED (Remotebus Check) wird angezeigt, ob die Verbindung zum vorhergehenden Interbus Gerät in Ordnung ist (ein), oder ob diese Verbindung unterbrochen ist (aus).
RD	Rot	Remotebus disabled Ist der weiterführende Fernbus abgeschaltet, so wird dies über die RD LED (Remotebus disabled) gemeldet.

**Buchsen und Stecker**

Es befindet sich je eine Schnittstelle für die ankommende und die weiterführende Busleitung auf der Frontseite des Geräts, jeweils in Form einer 9poligen SubD-Verbindung ausgeführt.

Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:

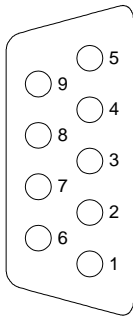
Ankommende Busleitung (9pol SubD-Stecker)



Pin	Belegung
1	DO
2	DI
3	GND1
4	GND <sup>*)</sup>
5	nicht belegt
6	/DO
7	/DI
8	+5V <sup>*)</sup> (90 mA)
9	reserviert

<sup>\*)</sup>Spannungsversorgung für Lichtwellenleiterumsetzer  
Diese Spannung ist nicht galvanisch getrennt!

Weiterführende Busleitung (9pol SubD-Buchse)



Pin	Belegung
1	DO
2	DI
3	GND
4	reserviert
5	+ 5V (90 mA)
6	/DO
7	/DI
8	reserviert
9	RBST

**Spannungsversorgung**

Der Interbus-Koppler besitzt ein eingebautes Netzteil. Das Netzteil ist mit 24V Gleichspannung zu versorgen. Über die Versorgungsspannung werden neben der Buskopplerelektronik auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus mit maximal 3,5A versorgen kann.

Das Netzteil ist gegen Verpolung und Überstrom geschützt.  
Interbus und Rückwandbus sind galvanisch voneinander getrennt.



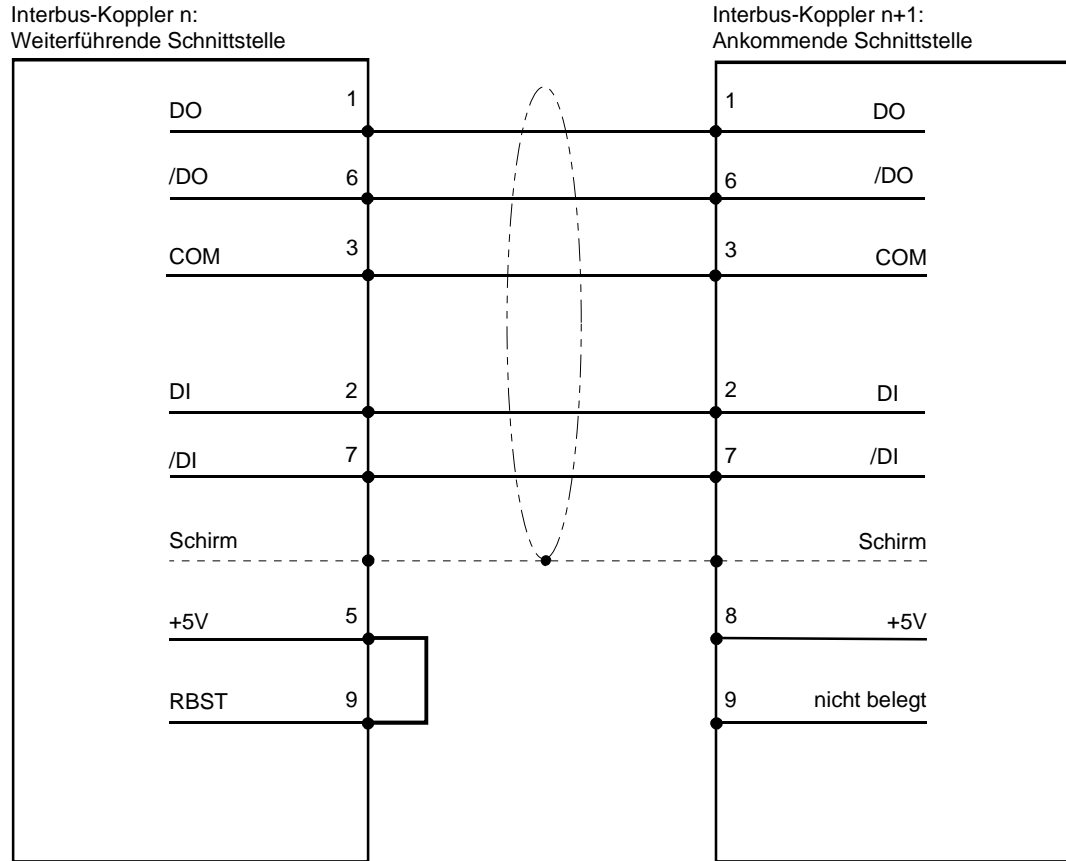
**Hinweis!**

Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung!



## Anschluss an Interbus

### Verkabelung unter Interbus



#### Potentialtrennung

Da Interbus-Fernbussegmente eine große räumliche Ausdehnung erreichen, müssen die einzelnen Segmente zur Vermeidung einer Potentialverschleppung galvanisch getrennt werden. Gemäß den Empfehlungen des Interbus-Clubs genügt jedoch eine galvanische Trennung der ankommenden Fernbus-Schnittstelle vom Rest der Schaltung. Die weiterführende Fernbus-Schnittstelle liegt demnach auf dem Potential der übrigen Schaltung und des Rückwandbus.

Verwenden Sie metallisierte Steckergehäuse und legen Sie den Kabelschirm auf das Steckergehäuse.



#### Hinweis!

Bitte beachten Sie, dass am Stecker für die "Weiterführende Schnittstelle" die Brücke zwischen Pin 5 und 9 vorhanden ist, ansonsten würden die nachfolgenden Slaves nicht erkannt werden!

## Einsatz im Interbus

### Prozessdaten-zuordnung

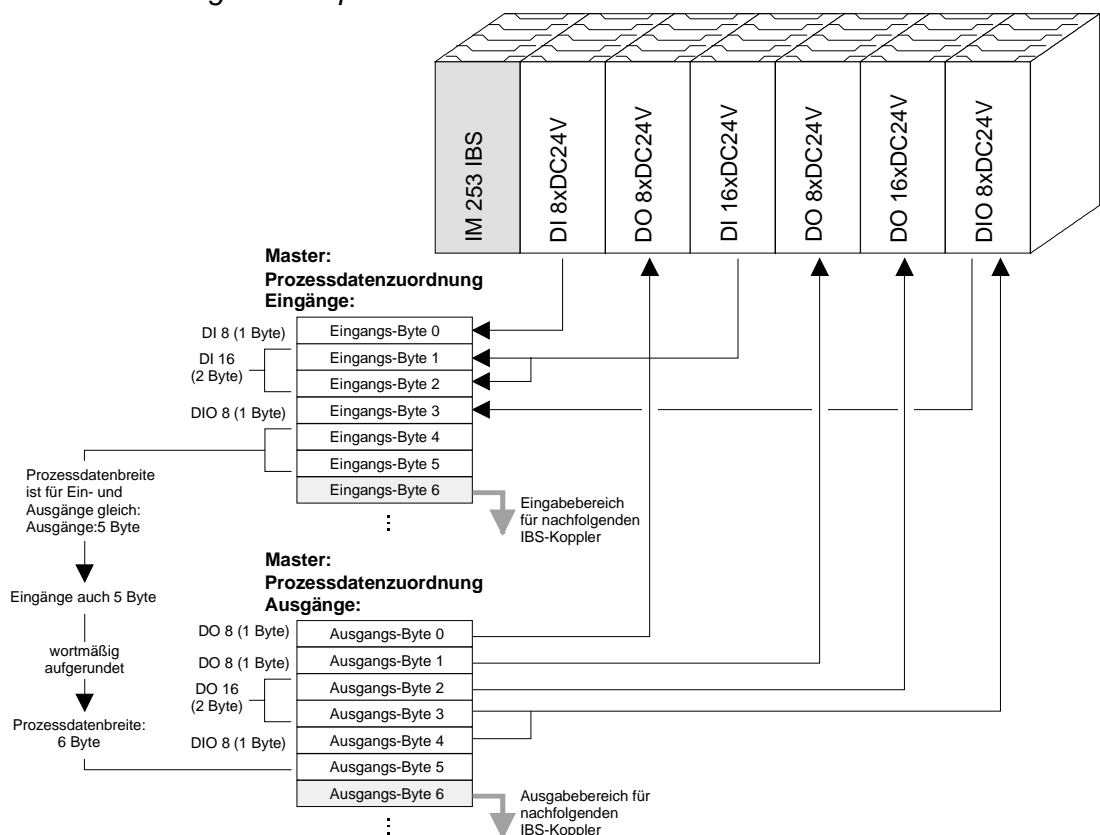
Nach dem Einschalten ermittelt der Buskoppler die Konfiguration der gesteckten Module und trägt diese in ein internes Prozessabbild ein. Dieses Prozessabbild schickt er an den Master. Der Master erstellt aus den Prozessabbildern eine Prozessdatenliste aller am Bus befindlichen Koppler. Die Prozessdatenzuordnungsliste finden Sie auch in den nachfolgenden zwei Abbildungen.

Bei der Erstellung des internen Prozessabbilds geht der Buskoppler nach folgenden Regeln vor:

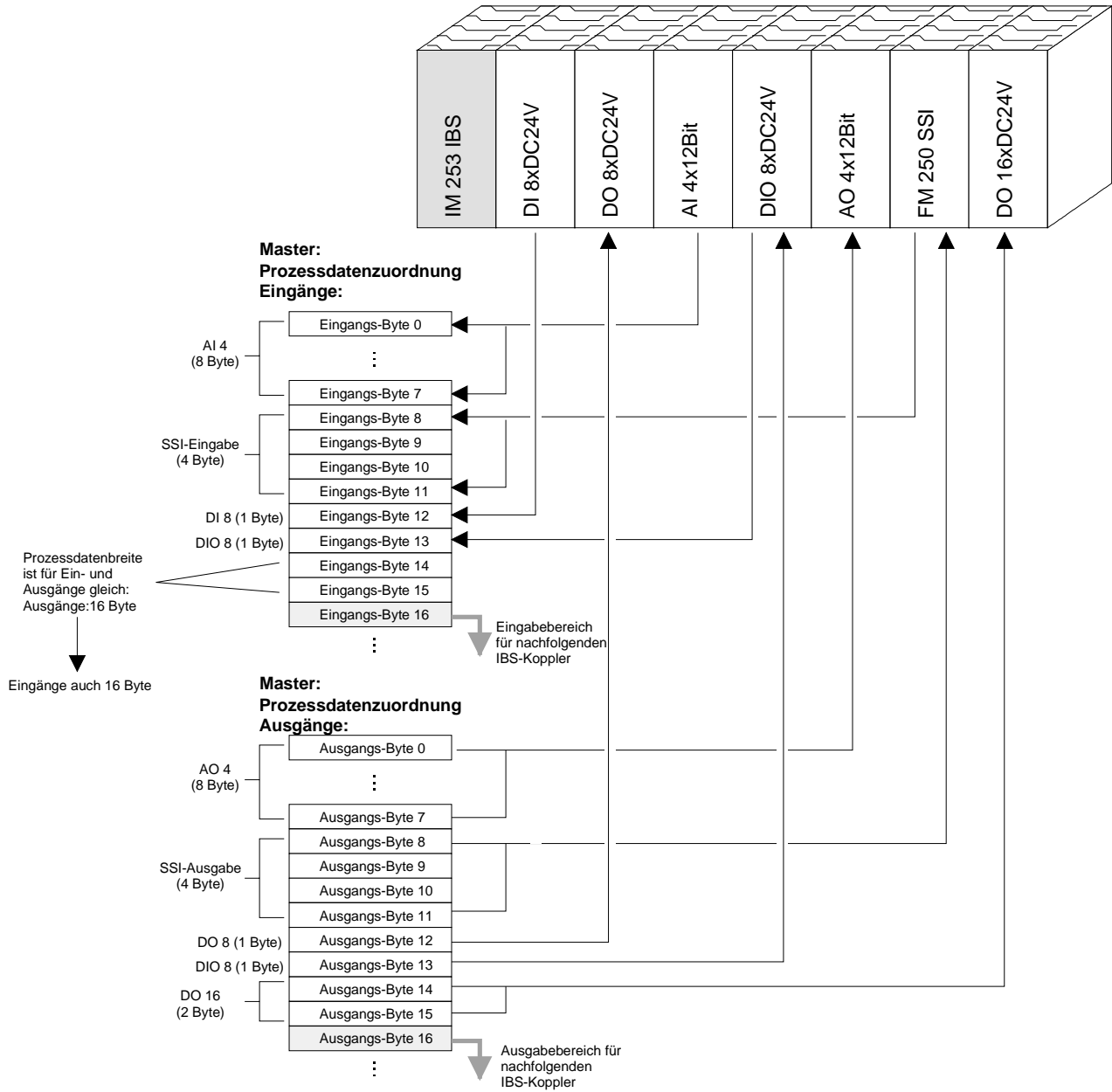
- Digitale Signale sind bitorientiert d.h. jedem Kanal ist ein Bit im Prozessabbild zugeordnet.
- Es gibt getrennte Bereiche für Ein- und Ausgangs-Daten.
- In den Ein- bzw. Ausgangsbereichen kommen an den Anfang immer die nicht digitalen Module und dann die digitalen Module.
- Die Reihenfolge der Zuweisung richtet sich nach der Steckplatzposition ausgehend vom Buskoppler.
- Bei einer unterschiedlichen Datenbreite von Ein- und Ausgängen ist der jeweils größere Wert für die Datenbreite eines Interbus-Kopplers maßgeblich, wobei immer wortmäßig aufgerundet wird (max. 20Byte).

Die nachfolgenden zwei Abbildungen sollen nochmals die Zuordnung der Prozessdaten innerhalb des Interbus-Masters verdeutlichen.

#### Rein digitale Peripherie



Gemischt digitale/analoge Peripherie



**Zyklischer Prozessdatenaustausch**

Der Austausch von Ein- und Ausgangsdaten erfolgt über ein Prozessabbild. Für die Kommunikation mit digitalen Ein- und Ausgängen steht je ein Speicherbereich zur Verfügung in dem die Ein- und Ausgangszustände der Module abgelegt werden.



**Datenkonsistenz**

Daten bezeichnet man als konsistent, wenn sie inhaltlich zusammengehören. Inhaltlich gehören zusammen: das High- und Low-Byte eines Analogwerts (wortkonsistent) und das Kontroll- und Status-Byte mit zugehörigem Parameterwort für den Zugriff auf die Register.

Die Datenkonsistenz der Daten einer Station ist durch das Übertragungsprotokoll des Interbus sichergestellt. Die Konsistenz über das gesamte Prozessabbild wird durch das synchrone Abtasten gewährleistet. Durch den asynchronen Zugriff der Steuerungs-CPU auf den Datenbereich des Interbus-Masters kann es zu Inkonsistenzen kommen. Hinweise zu sicheren Zugriffsverfahren auf die Masteranschaltung finden Sie in den zugehörigen Handbüchern.

Die Datenkonsistenz ist grundsätzlich nur für 1Byte sichergestellt. Das heißt, die Bits eines Bytes werden zusammen eingelesen bzw. ausgegeben. Für die Verarbeitung digitaler Signale ist eine byteweise Konsistenz ausreichend. Für Daten, deren Länge ein Byte überschreiten, wie z.B. bei Analogwerten, muss die Datenkonsistenz erweitert werden. Bitte beachten Sie, dass Sie die konsistenten Daten auf die richtige Art vom Interbus-Master in Ihre SPS übernehmen.

Hinweise hierzu finden Sie im Handbuch zu Ihrem Interbus-Master.

**Einschränkungen**

Sie können maximal 16 Eingangs- und 16 Ausgangsmodule mit einem Interbus-Koppler frei kombinieren. Sie haben für Eingangs- und Ausgangsdaten jeweils eine Datenbreite von maximal 10 Worten.

Eine Parametrierung des Buskopplers oder der Peripheriebaugruppen über das Interbus-PCP-Protokoll wird nicht unterstützt.

Während der Initialisierung des Buskopplers werden für die System 200V Peripherie-Module Adressen vergeben, mit deren Hilfe der Buskoppler im normalen Betrieb mit der Baugruppe kommuniziert. Da diese Adressen nur bei POWER-ON bzw. RESET vergeben werden können und sich während des Betriebs die Datenbreite von Interbus-Teilnehmern nicht verändern darf, dürfen während des Betriebs keine Module entfernt oder hinzugefügt werden.

Aufgrund der Datenübertragung nach RS422 darf ein Fernbus-Segment (= Abstand zwischen zwei Teilnehmern) bis zu 400m lang sein. Die maximale Gesamtausdehnung des Systems beträgt 12,8km.

**Hinweis!**

Vor einer Veränderung muss der entsprechende Buskoppler spannungslos gemacht werden. Bitte beachten Sie, dass Sie bei einer Veränderung der Peripherie die Initialisierung im Master anpassen!



## Inbetriebnahme

### Aufbau und Einbindung in Interbus

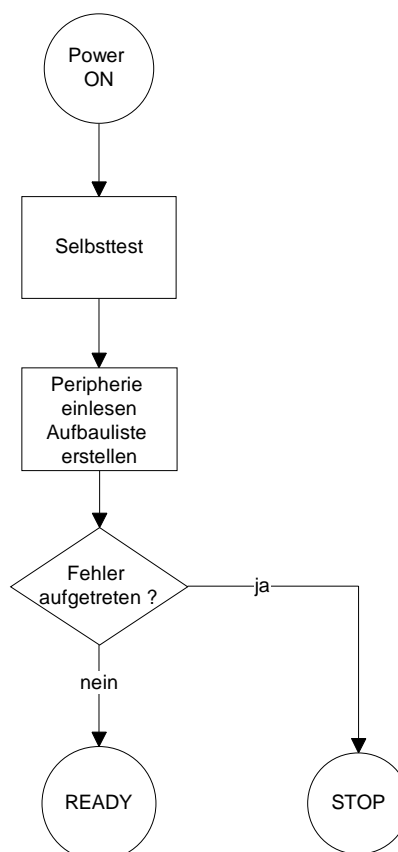
- Bauen Sie Ihren Interbus-Koppler mit den entsprechenden Modulen auf.
- Projektieren Sie den Interbus-Koppler mit dem mit dem Master mitgelieferten Projektierool.
- Schließen Sie das Interbus-Kabel am Koppler an und schalten Sie die Spannungsversorgung ein.

### Initialisierungsphase

Nach dem Einschalten überprüft der Buskoppler in einem Selbsttest die Funktionen seiner Bauteile und die Kommunikation mit dem Rückwandbus. Der Selbsttest wird angezeigt, indem nur die PW LED brennt. Nach erfolgreichem Test brennen RC und BA.

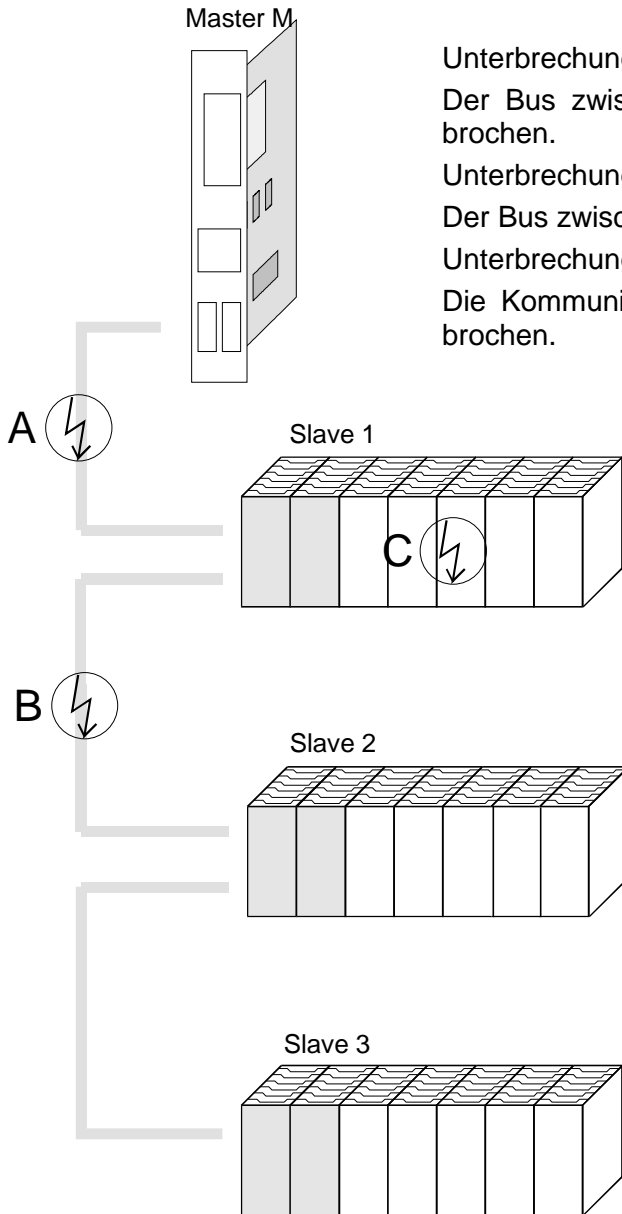
Nun erfolgt das Einlesen des Peripherieaufbaus. Zuerst wird die Anzahl der gesteckten Module eingelesen, anschließend werden die Module anhand ihrer Typkennungen identifiziert. Mit dem registrierten Peripherieaufbau werden Steckplatzkennungen für die Module generiert und über den Rückwandbus in die Module geschrieben. Es entsteht eine interne Aufbauliste, die von außen nicht zugänglich ist. Mit Hilfe dieser Steckplatzkennungen wird eine direkt adressierte Kommunikation ermöglicht. Für den Fall eines Fehlers geht der Buskoppler in den Zustand "STOP". Nach fehlerfreiem Hochlauf geht der Buskoppler in den Zustand "READY" über.

Der Buskoppler kann nach der Fehlerbeseitigung nur durch erneutes Einschalten in den normalen Betriebszustand gebracht werden.



**Einsatz der Diagnose-LEDs an einem Beispiel**

Das folgende Beispiel zeigt die Reaktion der LEDs bei unterschiedlichen Netzwerkunterbrechungen.



Unterbrechung Position A

Der Bus zwischen dem Master und Slave1 ist unterbrochen.

Unterbrechung Position B

Der Bus zwischen Slave1 und Slave2 ist unterbrochen

Unterbrechung Position C

Die Kommunikation über den Rückwandbus ist unterbrochen.

Slave 1		Unterbrechung Position		
LED	A	B	C	
ER	aus	aus	an	
BA	aus	aus	an	
RC	aus	an	an	
RD	an	an	aus	

Slave 2		Unterbrechung Position		
LED	A	B	C	
ER	aus	aus	aus	
BA	aus	aus	an	
RC	aus	aus	an	
RD	an	an	aus	

Slave 3		Unterbrechung Position		
LED	A	B	C	
ER	aus	aus	aus	
BA	aus	aus	an	
RC	aus	aus	an	
RD	an	an	an	

**Konfiguration des Masters**

Der Interbus erstellt, wie schon oben erläutert, einen Datenbereich mit Ein- und Ausgangsbytes. Die Zuordnung zwischen den mit dem Buskoppler verbundenen Modulen und den Bits und Bytes des Prozessabbilds wird durch den Buskoppler durchgeführt.

Der Interbus-Master tauscht mit jedem Interbus-Koppler einen zusammenhängenden Eingangs- und Ausgangsdatenblock aus. Die Zuordnung der Bytes aus diesem Datenblock zu den Adressen des Prozessabbilds wird über Datenbausteine der SPS oder durch eine Konfigurationssoftware durchgeführt.

Master-Software	Konfigurations-Software	Hersteller
SPS-Anschaltungen Version <4	SYS SWT	Phoenix Contact
SPS-Anschaltungen Version <4	IBM CMD	Phoenix Contact
PC-Anschaltungen Version <3	SYS SWT	Phoenix Contact
allgemein	SYS SWT	Phoenix Contact

## Technische Daten

### Interbus-Koppler IM 253IBS

Elektrische Daten	VIPA 253-1IB00
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme	max. 300mA
Ausgangsstrom Rückwandbus	max. 3,5A
Potenzialtrennung	≥ AC 500V, nach DIN 19258
Statusanzeigen	über LEDs auf der Frontseite
Anschlüsse / Schnittstellen	9pol. Sub-D (Stecker)      ankommender Fernbus 9pol. Sub-D (Buchse)      weiterführender Fernbus
Interbus Schnittstelle	
Ankopplung	Fernbus, 9pol. Sub-D nach DIN 19258
Netzwerk Topologie	Ring mit integrierter Rückleitung
Medium	Abgeschirmtes verdrilltes Twisted Pair
Übertragungsrate	500kBit/s
Gesamtlänge	12,8km
zwischen zwei Stationen	400m
digitale Ein-/Ausgänge	max.160 Ein- und 160 Ausgangsbits
max. Teilnehmeranzahl	256
Kombination mit Peripheriemodulen	
max. Modulanzahl	16
max. digital E/A	16 (Prozessdatenbreite 20 E / 20 A)
max. analog E/A	4 (Prozessdatenbreite 10 E / 10 A) keine Parametrierung möglich
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g

## Teil 4 CANopen

### Überblick

Inhalt dieses Kapitels ist die Beschreibung der CANopen-Slaves von VIPA. Nach einer Systemvorstellung folgt die Beschreibung der Module.

Neben einem Schnelleinstieg in die Projektierung für "Experten" finden Sie hier auch eine Einführung in die Telegrammstruktur und die Funktionscodes von CANopen.

Mit der Beschreibung des Emergency Objekts und NMT und den Technischen Daten endet das Kapitel.

Nachfolgend sind beschrieben:

- CAN-Bus-Grundlagen
- CANopen-Slaves von VIPA
- Einstellung von Baudrate und Modul-ID
- Einsatz des CANopen-Slaves im CAN-Bus mit Telegrammbeschreibung
- Beschreibung der CAN-spezifischen Objekte
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 4 CANopen</b> .....	<b>4-1</b>
Systemübersicht .....	4-2
Grundlagen.....	4-3
IM 253CAN - CANopen Slave - Aufbau .....	4-5
IM 253CAN, DO 24xDC 24V - Aufbau .....	4-9
CANopen Schnelleinstieg .....	4-13
Einstellung von Baudrate und Modul-ID.....	4-17
Telegrammaufbau .....	4-18
PDO - Prozessdaten-Objekt .....	4-20
SDO - Servicedaten-Objekt .....	4-24
Objekt-Verzeichnis.....	4-26
Emergency Object .....	4-67
NMT - Netzwerk Management .....	4-68
Technische Daten.....	4-70

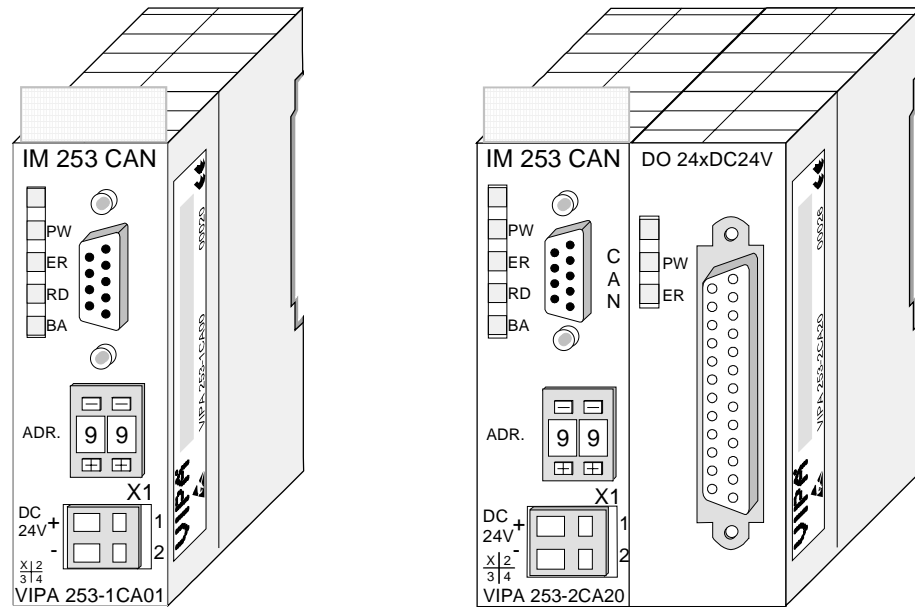
## Systemübersicht

Zur Zeit sind zwei CAN-Bus-Koppler von VIPA verfügbar:

- CANopen-Slave IM 253CAN
- CANopen-Slave IM 253CAN mit DO 24xDC 24V

### CANopen-Slave IM 253CAN

- 10 Receive- und 10 Transmit-PDOs (PDO-Linking, PDO-Mapping)
- 253-2CA20: 1 Receive-PDO (PDO-Linking, PDO-Mapping: fix)
- 2 SDOs
- Unterstützung aller Übertragungsraten



### Bestelldaten IM 253CAN

Bestellnummer	Beschreibung
VIPA 253-1CA01	CAN-Bus CANopen Slave
VIPA 253-2CA20	CAN-Bus CANopen Slave mit DO 24xDC 24V

## Grundlagen

### Allgemeines

Der CAN-Bus (**C**ontrol **A**rea **N**etwork) ist ein international offener Feldbus-Standard für Gebäude-, Fertigungs- und Prozessautomatisierung und wurde ursprünglich für die Automobiltechnik entwickelt.

Aufgrund der umfassenden Fehlererkennungs-Maßnahmen gilt der CAN-Bus als das sicherste Bussystem mit einer Restfehlerwahrscheinlichkeit von weniger als  $4,7 \times 10^{-11}$ . Fehlerhafte Meldungen werden signalisiert und automatisch neu übertragen.

Im Gegensatz zu Profibus und INTERBUS-S sind beim CAN-Bus auch verschiedene Schicht-7-Anwenderprofile unter dem CAL-Schicht-7-Protokoll definiert (CAL=**C**AN application layer). Ein solches Anwenderprofil ist CANopen, dessen Standardisierung der CiA (**C**AN in **A**utomation) e.V. übernimmt.

### CANopen

CANopen ist das Anwenderprofil für den Bereich industrieller Echtzeitsysteme und wird zur Zeit von vielen Herstellern implementiert. CANopen wurde als Profil DS-301 von der CAN-Nutzerorganisation (C.i.A) veröffentlicht. Das Kommunikationsprofil DS-301 dient zur Standardisierung der Geräte. Somit werden die Produkte verschiedener Hersteller austauschbar. Weiter sind zur Gewährleistung der Austauschbarkeit in dem Geräteprofil DS-401 die gerätespezifischen Daten und die Prozessdaten standardisiert. DS-401 standardisiert die digitalen und analogen Ein-/Ausgabe-Module.

CANopen besteht aus dem Kommunikationsprofil (communication profile) das festlegt, welche Objekte für die Übertragung bestimmter Daten zu verwenden sind, und den Geräteprofilen (device profiles), die die Art der Daten spezifizieren, die mit den Objekten übertragen werden.

Das CANopen-Kommunikationsprofil basiert auf einem Objektverzeichnis ähnlich dem des Profibus. Im Kommunikationsprofil DS-301 sind zwei Objektarten sowie einige Spezialobjekte definiert:

- Prozessdatenobjekte (PDO)  
PDOs dienen der Übertragung von Echtzeitdaten
- Servicedatenobjekte (SDO)  
SDOs ermöglichen den lesenden und schreibenden Zugriff auf das Objektverzeichnis

**Übertragungs-  
medium**

CAN basiert auf einer linienförmigen Topologie. Sie haben die Möglichkeit mittels Routerknoten eine Netzstruktur aufzubauen. Die Anzahl der Teilnehmer pro Netz wird nur durch die Leistungsfähigkeit des eingesetzten Bustreiberbausteins begrenzt.

Die maximale Netzausdehnung ist durch Signallaufzeiten begrenzt. Bei 1MBaud ist z.B. eine Netzausdehnung von 40m und bei 80kBaud von 1000m möglich.

CAN-Bus verwendet als Übertragungsmedium eine abgeschirmte Dreidrahtleitung (Fünfdraht optional).

Der CAN-Bus arbeitet mit Spannungsdifferenzen. Er ist daher unempfindlicher gegenüber Störeinflüssen als eine Spannungs- oder Stromschnittstelle. Das Netz sollte als Linie konfiguriert sein, mit einem 120Ω Abschlusswiderstand am Ende.

Auf dem VIPA CAN-Bus-Koppler befindet sich ein 9poliger Stecker. Über diesen Stecker koppeln Sie den CAN-Bus-Koppler als Slave direkt in das CAN-Bus-Netz ein.

Alle Teilnehmer im Netz kommunizieren mit der gleichen Baudrate.

Die Bus Struktur erlaubt das rückwirkungsfreie Ein- und Auskoppeln von Stationen oder die schrittweise Inbetriebnahme des Systems. Spätere Erweiterungen haben keinen Einfluss auf Stationen, die bereits in Betrieb sind. Es wird automatisch erkannt, ob ein Teilnehmer ausgefallen oder neu am Netz ist.

**Buszugriffs-  
verfahren**

Man unterscheidet bei Buszugriffsverfahren generell zwischen kontrolliertem (deterministischem) und unkontrolliertem (zufälligen) Buszugriff.

CAN arbeitet nach dem Verfahren Carrier-Sense Multiple Access (CSMA), d.h. jeder Teilnehmer ist bezüglich des Buszugriffs gleichberechtigt und kann auf den Bus zugreifen, sobald dieser frei ist (zufälliger Buszugriff).

Der Nachrichtenaustausch ist nachrichtenbezogen und nicht teilnehmerbezogen. Jede Nachricht ist mit einem priorisierenden Identifier eindeutig gekennzeichnet. Es kann immer nur ein Teilnehmer für seine Nachricht den Bus belegen.

Die Buszugriffssteuerung bei CAN geschieht mit Hilfe der zerstörungsfreien, bitweisen Arbitrierung. Hierbei bedeutet zerstörungsfrei, dass der Gewinner der Arbitrierung sein Telegramm nicht erneut senden muss. Beim gleichzeitigen Mehrfachzugriff von Teilnehmern auf den Bus wird automatisch der wichtigste Teilnehmer ausgewählt. Erkennt ein sendebereiter Teilnehmer, dass der Bus belegt ist, so wird sein Sendewunsch bis zum Ende der aktuellen Übertragung verzögert.

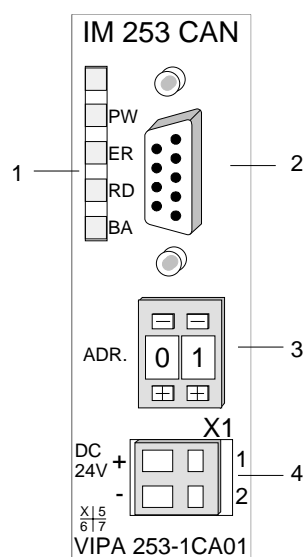


## IM 253CAN - CANopen Slave - Aufbau

### Eigenschaften

- 10 Rx und 10 Tx PDO
- 2 SDOs
- Unterstützung aller Übertragungsraten
- PDO-Linking
- PDO-Mapping

### Aufbau



- [1] LED Statusanzeigen
- [2] CAN-Bus-Stecker
- [3] Adress- bzw. Baudraten-einsteller
- [4] Anschluss für externe 24V Spannungsversorgung

### Komponenten

#### LEDs

Das Modul besitzt vier LEDs, die der Diagnose dienen. Die Verwendung und die jeweiligen Farben dieser Diagnose-LEDs finden Sie in den nachfolgenden Tabellen.

Bezeichnung	Farbe	Bedeutung
PW	Gelb	Signalisiert eine anliegende Betriebsspannung.
ER	Rot	Leuchtet bei Fehler in der Rückwandbusübertragung.
RD	Grün	Blinkt mit 1Hz bei positivem Selbsttest und erfolgreicher Initialisierung Leuchtet bei Datenübertragung über den VBUS.
BA	Gelb	Aus bei positivem Selbsttest und erfolgreicher Initialisierung. 1Hz Blinken im Zustand "Pre-Operational". Leuchtet im Zustand "Operational". 10Hz Blinken im Zustand "Prepared".

**Statusanzeige durch LED-Kombination**

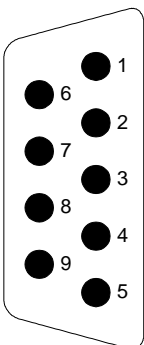
Durch Kombination der LEDs werden verschiedene Zustände angezeigt:

- PW ein
  - ER ein
  - RD ein
  - BA ein
- Fehler in RAM- oder EEPROM-Initialisierung
- 
- PW ein
  - ER blinkt 1Hz
  - RD blinkt 1Hz
  - BA blinkt 1Hz
- Baudrateneinstellung aktiviert
- 
- PW ein
  - ER blinkt 10Hz
  - RD blinkt 10Hz
  - BA blinkt 10Hz
- Fehler in der CAN-Baudrateneinstellung
- 
- PW ein
  - ER aus
  - RD blinkt 1Hz
  - BA aus
- Modul-ID-Einstellung aktiviert

**9poliger SubD-Stecker**

Der CAN-Bus-Koppler von VIPA wird über einen 9poligen Stecker in das CAN-Bus-System eingebunden.

Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:



Pin	Belegung
1	nicht belegt
2	CAN low
3	CAN Ground
4	nicht belegt
5	nicht belegt
6	optional Ground
7	CAN high
8	nicht belegt
9	optional pos. Supply

**Adresseinsteller für Baudrate und Modul-ID**

Über diesen Adresseinsteller stellen Sie die CAN-Baudrate sowie die Modul-ID ein.

Näheres hierzu finden Sie unter "Einstellung von Baudrate und Modul-ID" in diesem Kapitel.

**Spannungsversorgung**

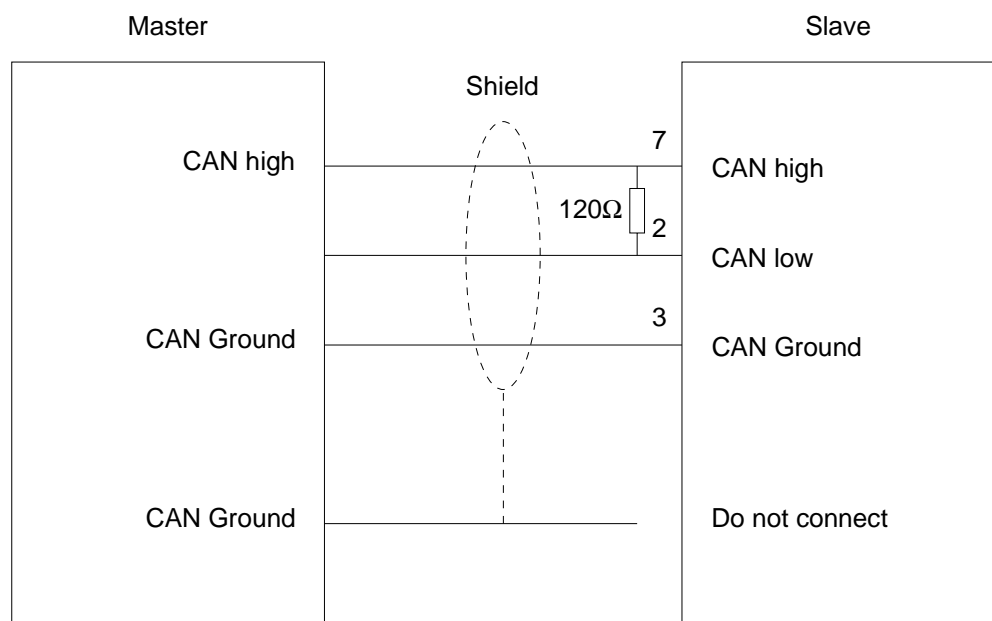
Der CAN-Bus-Koppler besitzt ein eingebautes Netzteil. Das Netzteil ist mit 24V Gleichspannung zu versorgen. Über die Versorgungsspannung werden neben der Buskopplerelektronik auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus mit maximal 3,5A versorgen kann.

Das Netzteil ist gegen Verpolung und Überstrom geschützt.

CAN-Bus und Rückwandbus sind galvanisch voneinander getrennt.

**Verkabelung unter CAN-Bus**

CAN-Bus verwendet als Übertragungsmedium eine abgeschirmte Dreidrahtleitung.

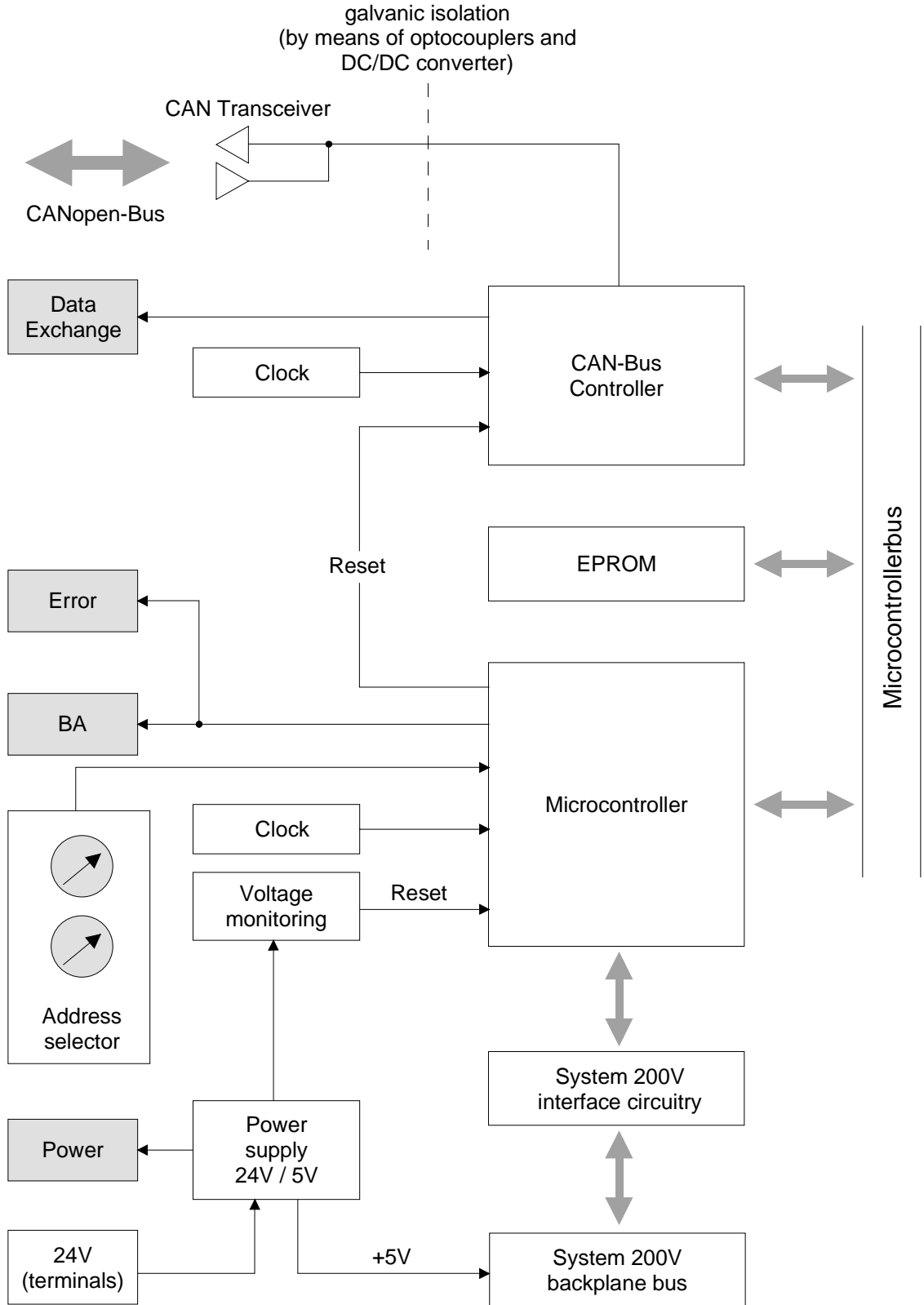
**Leitungsabschluss**

In Systemen mit mehr als zwei Stationen werden alle Teilnehmer parallel verdrahtet. Hierzu ist das Buskabel unterbrechungsfrei durchzuschleifen.

**Hinweis!**

An den Leitungsenden muss das Buskabel immer mit einem Abschlusswiderstand von 120Ω abgeschlossen werden, um Reflexionen und damit Übertragungsprobleme zu vermeiden!

**Blockschaltbild** Das nachfolgende Blockschaltbild zeigt den prinzipiellen Hardwareaufbau des Buskopplers und die Kommunikation, die intern stattfindet:

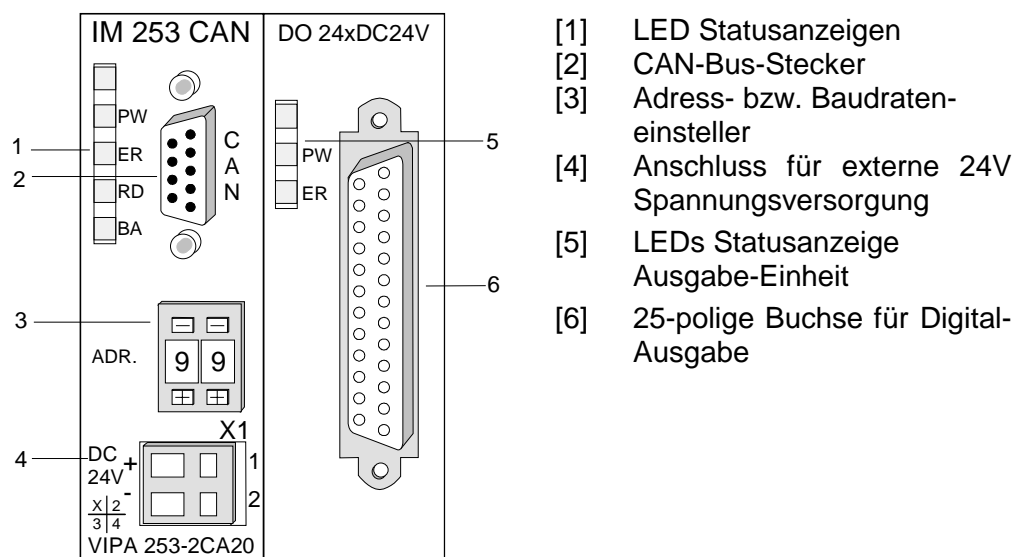


## IM 253CAN, DO 24xDC 24V - Aufbau

### Eigenschaften

- CANopen-Slave mit 24 digitalen Ausgängen "on-board"
- Projektierung über Standard-Tools (z.B. SyCon von Hilscher)
- 1 Rx PDO
- 2 SDOs
- Unterstützung aller Übertragungsraten
- PDO-Linking
- PDO-Mapping: fix

### Aufbau



### Komponenten

















#### LEDs

Das Modul besitzt vier LEDs, die der Busdiagnose dienen. Die Verwendung und die jeweiligen Farben dieser Diagnose-LEDs finden Sie in den nachfolgenden Tabellen.

Bezeichnung	Farbe	Bedeutung
PW	Gelb	Signalisiert eine anliegende Betriebsspannung.
ER	Rot	Leuchtet bei Fehler in der Rückwandbusübertragung.
RD	Grün	Blinkt mit 1Hz bei positivem Selbsttest und erfolgreicher Initialisierung Leuchtet bei Datenübertragung über den VBUS.
BA	Gelb	Aus bei positivem Selbsttest und erfolgreicher Initialisierung. 1Hz Blinken im Zustand "Pre-Operational". Leuchtet im Zustand "Operational". 10Hz Blinken im Zustand "Prepared".

**Statusanzeige durch LED-Kombination**

Durch Kombination der LEDs werden verschiedene Zustände angezeigt:

- |   |        |  |
|---|--------|--|
|  | PW ein | Fehler in RAM- oder EEPROM-Initialisierung |
|  | ER ein |  |
|  | RD ein |  |
|  | BA ein |  |
  
- |   |               |                                |
|---|---------------|--------------------------------|
|  | PW ein        | Baudrateneinstellung aktiviert |
|  | ER blinkt 1Hz |                                |
|  | RD blinkt 1Hz |                                |
|  | BA blinkt 1Hz |                                |
  
- |   |                |  |
|---|----------------|--|
|  | PW ein         | Fehler in der CAN-Baudrateneinstellung |
|  | ER blinkt 10Hz |  |
|  | RD blinkt 10Hz |  |
|  | BA blinkt 10Hz |  |
  
- |   |               |                                |
|---|---------------|--------------------------------|
|    | PW ein        | Modul-ID-Einstellung aktiviert |
|    | ER aus        |                                |
|   | RD blinkt 1Hz |                                |
|  | BA aus        |                                |

**LEDs Digital-Ausgabe-Teil**

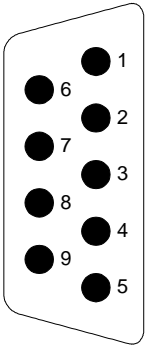
Auf dem digitalen Ausgabe-Teil befinden sich 2 LEDs, die folgende Funktion haben:

Bezeichnung	Farbe	Bedeutung
PW	Gelb	Signalisiert eine anliegende Betriebsspannung über den Profibus-Teil (Power).
ER	Rot	Leuchtet bei Kurzschluss, Überlast und Überhitzung

**9poliger SubD-Stecker**

Der CAN-Bus-Koppler von VIPA wird über einen 9poligen Stecker in das CAN-Bus-System eingebunden.

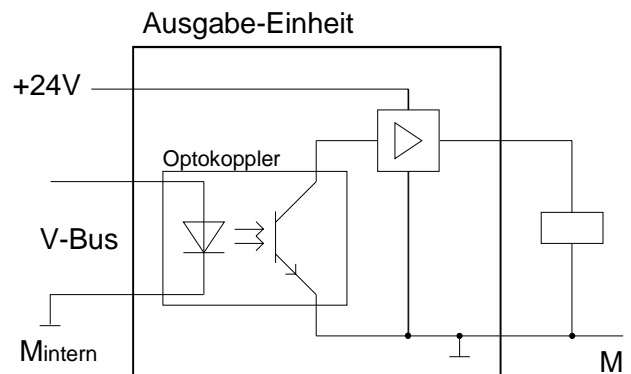
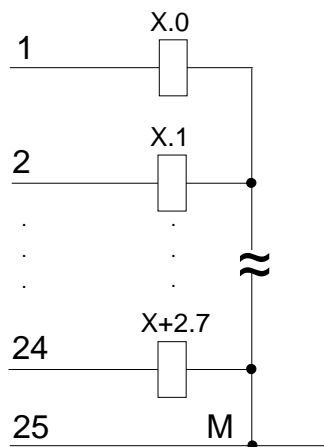
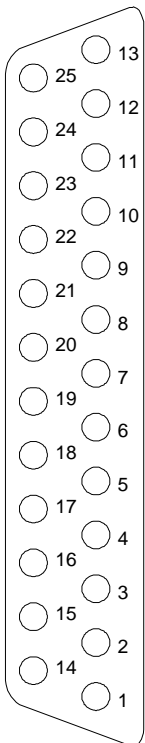
Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:



Pin	Belegung
1	nicht belegt
2	CAN low
3	CAN Ground
4	nicht belegt
5	nicht belegt
6	optional Ground
7	CAN high
8	nicht belegt
9	optional pos. Supply

**Ausgabe-Einheit Anschluss- und Prinzipschaltbild**

Die DC 24V-Spannungsversorgung des Ausgabe-Teils erfolgt intern über die Spannungsversorgung des Slave-Teils.



**Adresseinsteller für Baudrate und Modul-ID**

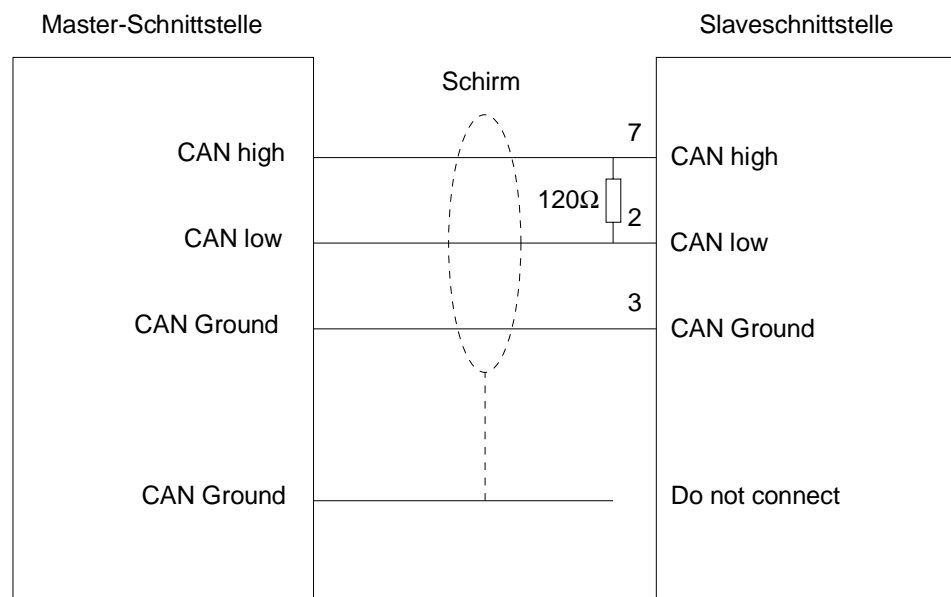
Über diesen Adresseinsteller stellen Sie die CAN-Baudrate sowie die Modul-ID ein.  
 Näheres hierzu finden Sie unter "Einstellung von Baudrate und Modul-ID" in diesem Kapitel.

**Spannungsversorgung**

Der CAN-Bus-Koppler besitzt ein eingebautes Netzteil. Das Netzteil ist mit 24V Gleichspannung zu versorgen. Über die Versorgungsspannung werden neben der Buskopplerelektronik auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus mit maximal 3,5A versorgen kann.  
 Das Netzteil ist gegen Verpolung und Überstrom geschützt.  
 CAN-Bus und Rückwandbus sind galvanisch voneinander getrennt.

**Verkabelung unter CAN-Bus**

CAN-Bus verwendet als Übertragungsmedium eine abgeschirmte Dreidrahtleitung.



**Leitungsabschluss**

In Systemen mit mehr als zwei Stationen werden alle Teilnehmer parallel verdrahtet. Hierzu ist das Buskabel unterbrechungsfrei durchzuschleifen.



**Hinweis!**

An den Leitungsenden muss das Buskabel immer mit einem Abschlusswiderstand von 120Ω abgeschlossen werden, um Reflexionen und damit Übertragungsprobleme zu vermeiden!



## CANopen Schnelleinstieg

### Übersicht

Dieser Abschnitt richtet sich an erfahrene CANopen-Anwender, die CAN bereits kennen. Hier soll kurz gezeigt werden, welche Nachrichten für den Einsatz des System 200V unter CAN in der Ausgangskonfiguration erforderlich sind.



### Hinweis!

Bitte beachten Sie, dass in diesem Handbuch hexadezimale Zahlen in der für Programmierer üblichen "0x"-Schreibweise dargestellt werden.

z.B.: **0x15AE** = 15AEh

### Einstellung von Baudrate und Modul-ID

Über den Adress-Einsteller sind an den Bus-Kopplern eine einheitliche Übertragungsrate sowie eine unterschiedliche Knotenadresse (Node-ID) einzustellen.

Nach Einschalten der Spannungsversorgung haben Sie die Möglichkeit über 00 am Adresseinsteller innerhalb von 10s die Baudrate und die Modul-ID zu programmieren.

Näheres hierzu siehe weiter unten unter "Einstellung von Baudrate und Modul-ID".

### CAN-Identifizier

Die CAN-Identifizier für die Ein-/Ausgabe-Daten des System 200V werden aus den Knotenadressen (1...99) abgeleitet:

Datenart	Default CAN-Identifizier	Datenart	Default CAN-Identifizier
digitale Eingänge 1 ... 64Bit	0x180 + Knotenadresse	digitale Ausgänge 1 ... 64Bit	0x200 + Knotenadresse
analoge Eingänge 1 ... 4 Wörter	0x280 + Knotenadresse	analoge Ausgänge 1 ... 4 Wörter/Kanäle	0x300 + Knotenadresse
weitere digitale oder analoge Eingänge	0x380 + Knotenadresse	weitere digitale oder analoge Ausgänge	0x400 + Knotenadresse
	0x480 + Knotenadresse		0x500 + Knotenadresse
	0x680 + Knotenadresse		0x780 + Knotenadresse
	0x1C0 + Knotenadresse		0x240 + Knotenadresse
	0x2C0 + Knotenadresse		0x340 + Knotenadresse
	0x3C0 + Knotenadresse		0x440 + Knotenadresse
	0x4C0 + Knotenadresse		0x540 + Knotenadresse
	0x6C0 + Knotenadresse		0x7C0 + Knotenadresse

**Digitale Ein-/Ausgänge**

Die CAN-Nachrichten mit digitalen Eingangsdaten stellen sich wie folgt dar:  
*Identifizier 0x180+Knotenadresse + bis zu 8Byte Nutzdaten*

<b>Identifizier</b> 11Bit	<b>DI 0</b> 8Bit	<b>DI 1</b> 8Bit	<b>DI 2</b> 8Bit	...	<b>DI 7</b> 8Bit
---------------------------	------------------	------------------	------------------	-----	------------------

Die CAN-Nachrichten mit digitalen Ausgangsdaten stellen sich wie folgt dar:  
*Identifizier 0x200+Knotenadresse + bis zu 8Byte Nutzdaten*

<b>Identifizier</b> 11Bit	<b>DO 0</b> 8Bit	<b>DO 1</b> 8Bit	<b>DO 3</b> 8Bit	...	<b>DO 7</b> Bit
---------------------------	------------------	------------------	------------------	-----	-----------------

**Analoge Ein-/Ausgänge**

Die CAN-Nachrichten mit analog. Eingangsdaten stellen sich wie folgt dar:  
*Identifizier 0x280+Knotenadresse + bis zu 4Wörter Nutzdaten*

<b>Identifizier</b> 11Bit	<b>AI 0</b> 1Wort	<b>AI 1</b> 1Wort	<b>AI 2</b> 1Wort	<b>AI 3</b> 1Wort
---------------------------	-------------------	-------------------	-------------------	-------------------

Die CAN-Nachrichten mit analog. Ausgangsdaten stellen sich wie folgt dar:  
*Identifizier 0x300+Knotenadresse + bis zu 4Wörter Nutzdaten*

<b>Identifizier</b> 11Bit	<b>AI 0</b> 1Wort	<b>AI 1</b> 1Wort	<b>AI 2</b> 1Wort	<b>AI 3</b> 1Wort
---------------------------	-------------------	-------------------	-------------------	-------------------

**Node Guarding**

Da das System 200V per Default-Einstellung im ereignisgesteuerten Modus arbeitet (kein zyklischer Datenaustausch) wird der Ausfall eines Knotens nicht unbedingt erkannt. Abhilfe schafft hier die Überwachung der Knoten durch zyklische Statusabfrage (Node Guarding).

Hierzu wird zyklisch ein Statustelegamm über Remote-Transmit-Request (RTR) angefordert: Das Telegramm besteht nur aus einem 11Bit Identifizier:

*Identifizier 0x700+Knotenadresse*

<b>Identifizier</b> 11Bit
---------------------------

Der System 200V-Knoten antwortet mit einem Telegramm, das ein Statusbyte enthält:

*Identifizier 0x700+Knotenadresse + Statusbyte*

<b>Identifizier</b> 11Bit	<b>Status</b> 8Bit
---------------------------	--------------------

Bit 0 ... 6: Knotenstatus

0x7F: Pre-Operational

0x05: Operational

0x04: Stopped bzw. Prepared

Bit 7: Toggle-Bit, kippt nach jedem Senden

Damit der Buskoppler einen Ausfall des Netzwerk-Masters erkennt (Watchdog-Funktion), müssen noch die Guard-Time (Objekt 0x100C) und der Life-Time-Factor (Objekt 0x100D) auf Werte  $\neq 0$  gesetzt werden.

(Reaktionszeit bei Ausfall: Guard-Time x Life Time Factor).

**Heartbeat**

Neben dem Node Guarding unterstützt der System 200V CANopen Koppler den Heartbeat Mode.

Wird im Index 0x1017 (Heartbeat Producer Time) ein Wert eingetragen, so wird mit Ablauf des Heartbeat-Timers der Gerätezustand (Operational, Pre-Operational, ...) des Buskopplers mittels COB-Identifizier (0x700+Modul-Id) übertragen:

*Identifizier 0x700+Knotenadresse + Statusbyte*

<b>Identifizier</b> 11Bit	<b>Status</b> 8Bit
---------------------------	--------------------

Der Heartbeat Mode startet automatisch sobald im Index 0x1017 ein Wert größer 0 eingetragen ist.

**Emergency Object**

Um anderen Teilnehmern am CANopen-Bus interne Gerätefehler mit hoher Priorität mitteilen zu können, verfügt der VIPA CAN-Bus-Koppler über das Emergency Object.

Für das Emergency-Telegramm ist der nach dem Boot-Up im Objektverzeichnis in der Variablen 0x1014 fest eingestellte **COB-Identifizier** in Hexadezimaldarstellung: **0x80 + Modul-ID**.

Das Emergency-Telegramm ist stets 8Byte lang. Es besteht aus:

*Identifizier 0x80 + Knotenadresse + 8 Nutzdatenbyte*

<b>Identifizier</b> 11Bit	<b>EC0</b>	<b>EC1</b>	<b>Ereg</b>	<b>Inf0</b>	<b>Inf1</b>	<b>Inf2</b>	<b>Inf3</b>	<b>Inf4</b>
---------------------------	------------	------------	-------------	-------------	-------------	-------------	-------------	-------------

Error Code	Meaning	Info 0	Info 1	Info 2	Info 3	Info 4
0x0000	Reset Emergency	0x00	0x00	0x00	0x00	0x00
0x1000	Module Configuration has changed and Index 0x1010 is equal to 'save'	0x06	0x00	0x00	0x00	0x00
0x1000	Module Configuration has changed	0x05	0x00	0x00	0x00	0x00
0x1000	Error during initialization of backplane modules	0x01	0x00	0x00	0x00	0x00
0x1000	Error during module configuration check	0x02	Module Number	0x00	0x00	0x00
0x1000	Error during read/write module	0x03	Module Number	0x00	0x00	0x00
0x1000	Module parameterization error	0x30	Module Number	0x00	0x00	0x00
0x1000	Diagnostic alarm from an analog module	0x40 + Module Number	diagnostic byte 1	diagnostic byte 2	diagnostic byte 3	diagnostic byte 4
0x1000	Process alarm from an analog module	0x80 + Module Number	diagnostic byte 1	diagnostic byte 2	diagnostic byte 3	diagnostic byte 4

*Fortsetzung ...*

... Fortsetzung Emergency Objekt

Error Code	Meaning	Info 0	Info 1	Info 2	Info 3	Info4
0x1000	PDO Control	0xFF	0x10	PDO Number	LowByte Timer Value	HighByte Timer Value
0x5000	Module					
0x6300	SDO PDO-Mapping	LowByte MapIndex	HighByte MapIndex	No. Of Map Entries	0x00	0x00
0x8100	Heartbeat Consumer	Node ID	LowByte Timer Value	HighByte Timer Value	0x00	0x00
0x8100	SDO Block Transfer	0xF1	LowByte Index	HighByte Index	SubIndex	0x00
0x8130	Node Guarding Error	LowByte GuardTime	HighByte GuardTime	LifeTime	0x00	0x00
0x8210	PDO not processed due to length error	PDO Number	Wrong length	PDO length	0x00	0x00
0x8220	PDO length exceeded	PDO Number	Wrong length	PDO length	0x00	0x00



### Hinweis!

Mit den beschriebenen Telegrammen sind Sie nun in der Lage, das System 200V zu starten und zu stoppen, Eingänge zu lesen, Ausgänge zu schreiben und die Module zu überwachen.

Nachfolgend sind alle Funktionen nochmals detailliert beschrieben.

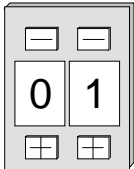
## Einstellung von Baudrate und Modul-ID

### Übersicht

Sie haben die Möglichkeit über 00 am Adresseinsteller nach Einschalten der Spannungsversorgung innerhalb von 10s die Baudrate und die Modul-ID zu programmieren.

Die eingestellten Werte werden in einem EEPROM dauerhaft gespeichert und können jederzeit durch erneute Programmierung geändert werden.

### Einstellung der Baudrate über Adresseinsteller



- Stellen Sie am Adresseinsteller die Adresse 00 ein
- Schalten Sie die Spannungsversorgung für den CAN-Bus-Koppler ein  
Die LEDs ER, RD, und BA blinken mit 1Hz. Nun können Sie innerhalb von 5s über den Adresseinsteller die CAN-Baudrate programmieren:

Adresseinsteller	CAN-Baudrate	garantierte max. Buslänge
"00"	1MBaud	25m
"01"	500kBaud	100m
"02"	250kBaud	250m
"03"	125kBaud	500m
"04"	100kBaud	600m
"05"	50kBaud	1000m
"06"	20kBaud	2500m
"07"	10kBaud	5000m
"08"	800kBaud	50m

Nach diesen 5 Sekunden wird die eingestellte CAN-Baudrate im EEPROM gespeichert.

### Einstellung der Modul-ID

Die LEDs ER und BA gehen aus und die grüne RD-LED blinkt weiterhin. Sie haben jetzt weitere 5s zur Einstellung der Modul-ID.

- Stellen Sie die Modul-ID im Bereich 01...99 am Adresseinsteller ein. Jede Modul-ID darf nur einmal am Bus vergeben sein. Die Modul-ID muss vor dem Einschalten des Buskopplers eingestellt werden.

Nach 5s werden die Einstellungen übernommen und der Buskoppler geht in den Normalbetrieb (Zustand "Pre-Operational").

### Einstellung der Baudrate über SDO-Write

Die CAN-Baudrate kann auch über ein SDO-Write auf das Objekt "0x2001" neu programmiert werden. Dieser Wert wird dann nach einem RESET des Buskopplers als CAN-Baudrate übernommen. Dies schafft eine sehr praktische Möglichkeit alle Buskoppler einer Anlage von einem zentralen CAN-Terminal auf eine neue CAN-Baudrate zu programmieren. Nach einem RESET der Anlage wird die neu programmierte Baudrate von den Buskopplern übernommen.

## Telegrammaufbau

### Identifizier

Alle CANopen Telegramme besitzen nach CiA DS-301 folgenden Aufbau:

#### Identifizier

Byte	Bit 7 ... Bit 0
1	Bit 3 ... Bit 0: Höchstwertige 4 Bits der Modul-ID Bit 7 ... Bit 4: CANopen Funktionscode
2	Bit 3 ... Bit 0: Datenlänge (DLC) Bit 4: RTR-Bit: 0: keine Daten (Anforderungstelegramm) 1: Daten vorhanden Bit 7 ... Bit 5: Niederwertige 3 Bits der Modul-ID

### Data

#### Data

Byte	Bit 7 ... Bit 0
3 ... 10	Daten

Der Unterschied zu einem Schicht-2-Telegramm besteht in einer zusätzlichen Unterteilung des 2 Byte Identifiers in einen Funktionsteil und eine Modul-ID. Im Funktionsteil wird die Art des Telegramms (Objekt) festgelegt und mit der Modul-ID wird der Empfänger adressiert.

Der Datenaustausch bei CANopen-Geräten erfolgt in Form von Objekten. Im CANopen-Kommunikationsprofil sind zwei Objektarten sowie einige Spezialobjekte definiert.

Der VIPA CAN-Bus-Koppler IM 253 CAN unterstützt folgende Objekte:

- 10 Transmit PDOs (PDO Linking, PDO Mapping)
- 10 Receive PDOs (PDO Linking, PDO Mapping)
- 2 Standard SDOs
- 1 Emergency Objekt
- 1 Netzwerkmanagement Objekt NMT
- Node Guarding
- Heartbeat

Der VIPA CAN-Bus-Koppler IM 253 CAN mit DO 24xDC24V unterstützt folgende Objekte:

- 1 Receive PDO (PDO-Linking, PDO-Mapping: fix)
- 2 Standard SDOs
- 1 Emergency Objekt
- 1 Netzwerkmanagement Objekt NMT
- Node Guarding
- Heartbeat

**CANopen Funktionscodes** Nachfolgend sind die unter CANopen definierten Objekte mit Funktionscode aufgelistet, die vom VIPA CAN-Bus-Koppler unterstützt werden:

Objekt	Function Code (4 Bits)	Empfänger	Definition	Funktion
NMT	0000	Broadcast	CiA DS-301	Netzwerkmanagement
EMERGENCY	0001	Master	CiA DS-301	Fehlertelegramm
PDO1S2M	0011	Master, Slave (RTR)	CiA DS-301	Digital Eing. Daten 1
PDO1M2S	0100	Slave	CiA DS-301	Digital Ausg. Daten 1
SDO1S2M	1011	Master	CiA DS-301	Konfigurationsdaten
SDO1M2S	1011	Slave	CiA DS-301	Konfigurationsdaten
Node Guarding	1110	Master, Slave (RTR)	CiA DS-301	Modulüberwachung
Heartbeat	1110	Master, Slave	Aplikationsspez.	Modulüberwachung

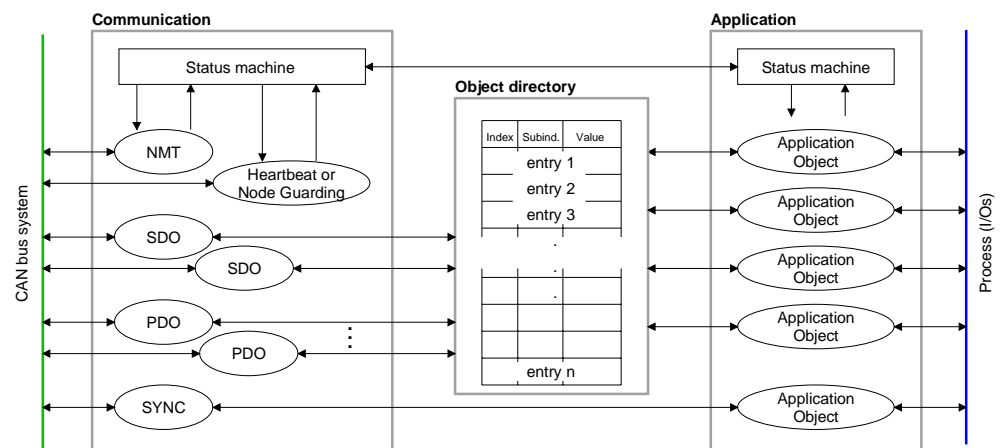


**Hinweis!**

Der genaue Aufbau und Dateninhalt aller Objekte ist im "CiA Communication Profile DS-301 Version 3.0" sowie im "CiA Device Profile for I/O-Modules DS-401 Version 1.4" detailliert beschrieben.

**Struktur des Gerätemodells**

Ein CANopen Gerät kann wie folgt strukturiert werden:



*Communication*

Stellt die Kommunikationsdatenobjekte und die zugehörige Funktionalität zum Datenaustausch über das CANopen Netzwerk zur Verfügung.

*Application*

Die Applikationsdatenobjekte enthalten z.B. Ein- und Ausgangsdaten. Eine Applikationsstatusmaschine überführt die Ausgänge im Fehlerfall in einen sicheren Zustand.

Das Objektverzeichnis ist wie eine zweidimensionale Tabelle organisiert. Die Daten werden über Index und Subindex adressiert.

*Object directory*

Dieses enthält alle Datenobjekte (Applikationsdaten + Parameter), die von außen zugänglich sind und die das Verhalten von Kommunikation, Applikation und Statusmaschinen beeinflussen.

## PDO - Prozessdaten-Objekt

### PDO

Bei vielen Feldbussystemen wird ständig das gesamte Prozessabbild übertragen - meist mehr oder weniger zyklisch. CANopen ist nicht auf dieses Kommunikationsprinzip beschränkt, da CAN durch die Multi-Master Buszugriffsregelung andere Möglichkeiten bietet.

Bei CANopen werden die Prozessdaten in Segmente zu maximal 8Byte aufgeteilt. Diese Segmente heißen **Prozessdaten-Objekte** (PDOs). Die PDOs entsprechen jeweils einem CAN-Telegramm und werden über dessen spezifischen CAN-Identifizier zugeordnet und in ihrer Priorität bestimmt.

Für den Prozessdatenaustausch stehen bei Einsatz des VIPA CAN-Bus-Kopplers IM 253CAN insgesamt 20 PDOs zur Verfügung. Jedes PDO besteht dabei aus maximal 8 Datenbytes. PDOs werden unbestätigt übertragen, da das CAN-Protokoll die Übertragung sicherstellt.

Für Eingangsdaten stehen 10 Tx Transmit-PDOs und für Ausgangsdaten 10 Rx Receive-PDOs zur Verfügung. Die PDOs werden aus Sicht des Buskopplers bezeichnet:

Receive-PDOs (RxPDOs) werden vom Koppler empfangen und enthalten Ausgangsdaten.

Transmit-PDOs (TxPDOs) werden vom Koppler gesendet und enthalten Eingangsdaten.

Die Belegung dieser PDOs mit Ein- bzw. Ausgangsdaten erfolgt automatisch.

### Variables PDO Mapping

CANopen legt die Datenbelegung für die ersten beiden PDOs im Geräteprofil fest. Die Belegung der PDOs ist in den Mapping-Tabellen im Objektverzeichnis hinterlegt. Diese Mapping-Tabellen bilden den Querverweis zwischen den Applikationsdaten im Objektverzeichnis und der Reihenfolge in den PDOs.

Die vom Koppler automatisch erzeugte Belegung der PDOs sind in der Regel ausreichend. Für spezielle Anwendungen kann diese Belegung geändert werden. Hierzu sind die Mapping-Tabellen entsprechend zu konfigurieren. Zunächst wird eine 0 auf Subindex 0 geschrieben (deaktiviert aktuelle Mapping-Konfiguration). Daraufhin tragen Sie die gewünschten Applikationsobjekte in Subindex 1...8 ein. Abschließend wird die Anzahl der nun gültigen Einträge in Subindex 0 parametrisiert und der Koppler überprüft die Einträge auf Konsistenz.



### Hinweis!

Bei Einsatz des IM 253 mit DO 24xDC24V steht nur ein Receive-PDO zur Verfügung, das PDO-Mapping ist hier fix.



**PDO Identifier  
COB-ID**

Der wichtigste Kommunikationsparameter eines PDOs ist der CAN-Identifier (auch Communication Object Identifier, COB-ID genannt). Er dient zur Identifizierung der Daten und bestimmt deren Priorität beim Buszugriff.

Für jedes CAN-Datentelegramm darf es nur einen Sendeknoten (Producer) geben. Da CAN jedoch alle Nachrichten im Broadcast-Verfahren sendet, kann ein Telegramm von beliebig vielen Knoten empfangen werden (Consumer). Ein Knoten kann also seine Eingangsinformation mehreren Busteilnehmern gleichzeitig zur Verfügung stellen - auch ohne Weiterleitung durch einen logischen Bus-Master.

Im System 200V sind für Sende- und Empfangs-PDOs Default-Identifier in Abhängigkeit von der Knotenadresse vorgesehen.

Die nach dem Boot-Up fest eingestellten COB-Identifier für die Empfangs- und Sende-PDO-Transfers sind nachfolgend aufgelistet.

Der Transmissionstyp ist im Objektverzeichnis (Indizes 0x1400-0x1409 und 0x1800-0x1809, Subindex 0x02) fest auf asynchron, Event gesteuert (= 0xFF) eingestellt. Über den EVENT-Timer (Value \* 1ms) können die PDOs zyklisch übertragen werden.

Send:                    0x180 + Modul-ID: PDO1S2M digital                    (nach DS-301)  
                           0x280 + Modul-ID: PDO2S2M analog  
                           0x380 + Modul-ID: PDO3S2M digital oder analog  
                           0x480 + Modul-ID: PDO4S2M  
                           0x680 + Modul-ID: PDO5S2M  
                           0x1C0 + Modul-ID: PDO6S2M  
                           0x2C0 + Modul-ID: PDO7S2M  
                           0x3C0 + Modul-ID: PDO8S2M  
                           0x4C0 + Modul-ID: PDO9S2M  
                           0x6C0 + Modul-ID: PDO10S2M

Receive:                0x200 + Modul-ID: PDO1M2S digital                    (nach DS-301)  
                           0x300 + Modul-ID: PDO2M2S analog  
                           0x400 + Modul-ID: PDO3M2S digital oder analog  
                           0x500 + Modul-ID: PDO4M2S  
                           0x780 + Modul-ID: PDO5M2S  
                           0x240 + Modul-ID: PDO6M2S  
                           0x340 + Modul-ID: PDO7M2S  
                           0x440 + Modul-ID: PDO8M2S  
                           0x540 + Modul-ID: PDO9M2S  
                           0x7C0 + Modul-ID: PDO10M2S

<b>PDO-Linking</b>	<p>Wenn das Consumer-Producer-Modell der CANopen-PDOs zum direkten Datenaustausch zwischen Knoten (ohne Master) genutzt werden soll, so muss die Identifier-Verteilung entsprechend angepasst werden, damit der TxPDO-Identifier des Producers mit dem RxPDO-Identifier des Consumers übereinstimmt:</p> <p>Dieses Verfahren nennt man PDO-Linking. Es ermöglicht beispielsweise den einfachen Aufbau von elektronischen Getrieben, bei denen mehrere Slave-Achsen gleichzeitig auf den Ist-Wert im TxPDO der Master-Achse hören.</p>
<b>PDO Kommunikationsarten</b>	<p>CANopen bietet folgende Möglichkeiten der Prozessdatenübertragung:</p> <ul style="list-style-type: none"><li>• Ereignisgesteuert</li><li>• Gepollt</li><li>• Synchronisiert</li></ul>
<b>Ereignisgesteuert</b>	<p>Das "Ereignis" ist die Änderung eines Eingangswertes, die Daten werden sofort nach dieser Änderung verschickt. Durch die Ereignissteuerung wird die Busbandbreite optimal ausgenutzt, da nicht ständig das Prozessabbild, sondern nur die Änderung desselben übertragen wird. Gleichzeitig wird eine kurze Reaktionszeit erreicht, da bei Änderung eines Eingangswertes nicht erst auf die nächste Abfrage durch einen Master gewartet werden muss.</p>
<b>Gepollt</b>	<p>Die PDOs können auch durch Datenanforderungstelegramme (Remote Frames) gepollt werden. Auf diese Art kann etwa das Eingangsprozessabbild bei ereignisgesteuerten Eingängen auch ohne Eingangsänderung auf den Bus gebracht werden, beispielsweise bei einem zur Laufzeit ins Netz aufgenommenen Monitor- oder Diagnosegerät.</p> <p>Die VIPA CANopen Buskoppler unterstützen die Abfrage von PDOs über Remote Frames - da dies hardwarebedingt aber nicht bei allen CANopen Geräten vorausgesetzt werden kann, ist diese Kommunikationsart nur bedingt zu empfehlen.</p>
<b>Synchronisiert</b>	<p>Nicht nur bei Antriebsanwendungen ist es sinnvoll, das Ermitteln der Eingangsinformation sowie das Setzen der Ausgänge zu synchronisieren. CANopen stellt hierzu das SYNC-Objekt zur Verfügung, ein CAN-Telegramm hoher Priorität ohne Nutzdaten, dessen Empfang von den synchronisierten Knoten als Trigger für das Lesen der Eingänge bzw. für das Setzen der Ausgänge verwendet wird.</p>

**PDO  
Übertragungsart**

Der Parameter "PDO Übertragungsart" legt fest, wie das Versenden des PDOs ausgelöst wird bzw. wie empfangene PDOs behandelt werden:

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

**Synchron**

Die Übertragungsart 0 ist nur für RxPDOs sinnvoll: Das PDO wird erst nach Empfang des nächsten SYNC-Telegramms ausgewertet.

Bei Übertragungsart 1-240 wird das PDO zyklisch gesendet bzw. erwartet: nach jedem "n-ten" SYNC ( $n=1\dots240$ ). Da die Übertragungsart nicht nur im Netz, sondern auch auf einem Koppler kombiniert werden darf, kann so z.B. ein schneller Zyklus für digitale Eingänge vereinbart werden ( $n=1$ ), während die Daten der Analog-Eingänge in einem langsameren Zyklus übertragen werden (z.B.  $n=10$ ). Die Zykluszeit (SYNC-Rate) kann überwacht werden (Objekt 0x1006), der Koppler schaltet bei SYNC-Ausfall dann seine Ausgänge in den Fehlerzustand.

**Asynchron**

Die Übertragungsarten 254 + 255 sind asynchron oder auch ereignisgesteuert. Bei Übertragungsart 254 ist das Ereignis herstellerspezifisch bei 255 im Geräteprofil definiert.

Bei der Wahl der ereignisgesteuerten PDO-Kommunikation ist zu berücksichtigen, dass u.U. viele Ereignisse gleichzeitig auftreten können und sich dann entsprechende Verzögerungszeiten einstellen können, bis ein relativ niederpriories PDO verschickt werden kann.

Auch muss verhindert werden, dass ein sich ständig ändernder Eingang mit hoher PDO-Priorität den Bus blockiert ("babbling idiot").

**Inhibit-Zeit**

Über den Parameter "Inhibit-Zeit" kann ein "Sende-Filter" aktiviert werden, der die Reaktionszeit bei der relativ ersten Eingangsänderung nicht verlängert, aber bei unmittelbar darauffolgenden Änderungen aktiv ist.

Die Inhibit-Zeit (Sendeverzögerungszeit) beschreibt die Zeitspanne, die zwischen dem Versenden zweier gleicher Telegramme mindestens abgewartet werden muss.

Wenn die Inhibit-Zeit genutzt wird, können Sie die maximale Busbelastung und damit die Latenzzeit im "worst case"-Fall ermitteln.

## SDO - Servicedaten-Objekt

### SDO

Für Zugriffe auf das Objektverzeichnis wird das Service-Daten-Objekt (SDO) verwendet. Mit dem SDO können Sie lesend oder schreibend auf das Objektverzeichnis zugreifen. Im CAL-Schicht-7-Protokoll finden Sie die Spezifikation des Multiplexed-Domain-Transfer-Protocol, das von den SDOs genutzt wird. Mit diesem Protokoll können Sie Daten beliebiger Länge übertragen. Hierbei werden Nachrichten gegebenenfalls auf mehrere CAN-Nachrichten mit gleichem Identifier aufgeteilt (Segmentierung).

In der ersten CAN-Nachricht des SDOs sind 4 der 8 Bytes mit Protokollinformationen belegt. Für Zugriffe auf Objektverzeichniseinträge mit bis zu vier Bytes Länge genügt eine einzige CAN-Nachricht. Bei Datenlängen größer als 4 Bytes erfolgt eine segmentierte Übertragung. Die nachfolgenden Segmente des SDOs enthalten bis zu 7 Bytes Nutzdaten. Das letzte Byte enthält eine Endekennung. Ein SDO wird bestätigt übertragen, d.h. jeder Empfang einer Nachricht wird quittiert.

Die für Lese- und Schreibzugriff vorgesehenen COB-Identifizier sind:

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



### Hinweis!

Eine nähere Beschreibung der SDO-Telegramme finden sie in der vom CiA verfassten DS-301 Norm.

Nachfolgend sollen lediglich die Fehlermeldungen aufgeführt werden, die im Falle einer fehlerhaften Parameterkommunikation erzeugt werden.

**SDO Error-Codes**

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

## Objekt-Verzeichnis

### Struktur

Im CANopen-Objektverzeichnis werden alle für den Buskoppler relevanten CANopen Objekte eingetragen. Jeder Eintrag im Objektverzeichnis ist durch einen 16Bit-Index gekennzeichnet.

Falls ein Objekt aus mehreren Komponenten besteht (z.B. Objekttyp Array oder Record), sind die Komponenten über einen 8Bit-Subindex gekennzeichnet.

Der Objektname beschreibt die Funktion eines Objekts. Das Datentyp-Attribut spezifiziert den Datentyp des Eintrags.

Über das Zugriffsattribut ist spezifiziert, ob ein Eintrag nur gelesen werden kann, nur geschrieben werden oder gelesen und geschrieben werden darf.

Das Objektverzeichnis ist in folgende 3 Bereiche aufgeteilt:

### Kommunikationsspezifischer Profilbereich (0x1000 – 0x1FFF)

Dieser Bereich beinhaltet die Beschreibung aller spezifischen Parameter für die Kommunikation.

0x1000 – 0x1011 allgemeine kommunikationsspezifische Parameter (z.B. der Gerätenamen)

0x1400 – 0x140F Kommunikationsparameter (z.B. Identifier) der Receive-PDOs

0x1600 – 0x160F Mappingparameter der Receive-PDOs  
Die Mappingparameter enthalten die Querverweise auf die Applikationsobjekte, die in die PDOs gemappt sind und die Datenbreite des entsprechenden Objektes

0x1800 – 0x180F Kommunikations- und Mappingparameter der  
0x1A00 – 0x1A0F Transmit-PDOs

### Herstellerspezifischer Profilbereich (0x2000 – 0x5FFF)

Hier finden Sie die herstellerspezifischen Einträge wie z.B. PDO-Control, CAN-Baudrate (Baudrate nach RESET) usw.

### Standardisierter Geräteprofilbereich (0x6000 – 0x9FFF)

In diesem Bereich liegen die Objekte für das Geräteprofil nach DS-401.



### Hinweis!

Da die CiA Normen ausschließlich in englischer Sprache vorliegen, wurden die Tabelleneinträge der Objekte zum eindeutigen Verständnis in englischer Sprache übernommen.

Eine nähere Beschreibung der Tabelleneinträge in Deutsch finden Sie jeweils unterhalb der Tabellen.

**Objektverzeichnis  
Übersicht**

Index		Content of Object
0x1000		Device type
0x1001		Error register
0x1003		Error store
0x1004		Number of PDOs
0x1005		SYNC identifier
0x1006		SYNC interval
0x1008		Device name
0x1009		Hardware version
0x100A		Software version
0x100B		Node number
0x100C		Guard time
0x100D		Life time factor
0x100E		Node Guarding Identifier
0x1010	X	Save parameter
0x1011	X	Load parameter
0x1014		Emergency COB-ID
0x1016	X	Heartbeat consumer time
0x1017	X	Heartbeat producer time
0x1018		Device identification
0x1027		Module list
0x1029		Error behavior
0x1400 - 0x1409	X	Communication parameter für Receive-PDOs (RxPDO, Master to Slave)
0x1600 - 0x1609	X	Mappingparameter for Receive-PDOs (RxPDO)
0x1800 - 0x1809	X	Communication parameter for Transmit-PDOs (TxPDO, Slave to Master)
0x1A00 - 0x1A09	X	Mappingparameter für Transmit-PDOs (TxPDO)
0x2001		CAN-Baudrate
0x2100		Kill EEPROM
0x2101		SJA1000
0x2400	X	PDO Control
0x3001 - 0x3010	X	Module Parameterization
0x3401	X	Module Parameterization
0x6000		Digital-Input-8-Bit Array (see DS 401)
0x6002	X	Polarity Digital-Input-8-Bit Array (see DS 401)
0x6100		Digital-Input-16-Bit Array (see DS 401)
0x6102		Polarity Digital-Input-16-Bit Array (v DS 401)
0x6120		Digital-Input-32Bit Array (see DS 401)
0x6122		Polarity Digital-Input-32-Bit Array (see DS 401)
0x6200		Digital-Output-8-Bit Array (see DS 401)
0x6202	X	Polarity Digital-Output-8-Bit Array (see DS 401)
0x6206	X	Fault Mode Digital-Output-8-Bit Array (see DS 401)
0x6207	X	Fault State Digital-Output-8-Bit Array (see DS 401)
0x6300		Digital-Output-16-Bit Array (see DS 401)

*Fortsetzung ...*

... Fortsetzung  
**Objektverzeichnis**  
**Übersicht**

Index		Content of Object
0x6302		Polarity Digital-Output-16-Bit Array (see DS 401)
0x6306		Fault Mode Digital-Output-16-Bit Array (see DS 401)
0x6307		Fault State Digital-Output-16-Bit Array (see DS 401)
0x6320		Digital-Output-32-Bit Array (see DS 401)
0x6322		Polarity Digital-Output-32-Bit Array (see DS 401)
0x6326		Fault Mode Digital-Output-32-Bit Array (see DS 401)
0x6327		Fault State Digital-Output-32-Bit Array (see DS 401)
0x6401		Analog-Input Array (see DS 401)
0x6411		Analog-Output Array (see DS 401)
0x6421	X	Analog-Input Interrupt Trigger Array (see DS 401)
0x6422		Analog-Input Interrupt Source Array (see DS 401)
0x6423	X	Analog-Input Interrupt Enable (see DS 401)
0x6424	X	Analog-Input Interrupt Upper Limit Array (see DS 401)
0x6425	X	Analog-Input Interrupt Lower Limit Array (see DS 401)
0x6426	X	Analog-Input Interrupt Delta Limit Array (see DS 401)
0x6443	X	Fault Mode Analog-Output Array (see DS 401)
0x6444	X	Fault State Analog-Output Array (see DS 401)

X = save into EEPROM



## Device Type

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

Der 32Bit-Wert ist in zwei 16Bit-Felder unterteilt:

MSB	LSB
<b>Additional information Device</b>	<b>profile number</b>
0000 0000 0000 wxyz (bit)	401dec=0x0191

Die "Additional Information" enthält Angaben über die Signalarten des I/O-Gerätes:

z=1 bedeutet digitale Eingänge

y=1 digitale Ausgänge

x=1 analoge Eingänge

w=1 analoge Ausgänge

## Error register

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

Bit7							Bit0
ManSpec	reserved	reserved	Comm.	reserved	reserved	reserved	Generic

ManSpec.: Herstellerspezifischer Fehler, wird in Objekt 0x1003 genauer spezifiziert.

Comm.: Kommunikationsfehler (Overrun CAN)

Generic: Ein nicht näher spezifizierter Fehler ist aufgetreten (Flag ist bei jeder Fehlermeldung gesetzt)

**Error store**

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

Das "Predefined Error Field" ist in zwei 16Bit-Felder unterteilt:

MSB	LSB
Additional information	Error code

Der "Additional Code" enthält den Error Trigger (siehe Emergency Objekt) und damit eine detaillierte Fehlerbeschreibung.

Neue Fehler werden jeweils an Subindex 1 gespeichert, alle anderen Subindices werden entsprechend inkrementiert.

Durch Schreiben einer "0" auf Subindex 0 wird der gesamte Fehlerspeicher gelöscht. Wenn kein Fehler seit dem PowerOn aufgetreten ist, dann besteht Objekt 0x1003 nur aus Subindex 0 mit eingetragener "0".

Durch einen Reset oder Power Cycle wird der Fehlerspeicher gelöscht.

**Number of PDOs**

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

Der 32Bit-Wert ist in zwei 16Bit-Felder unterteilt:

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

### SYNC identifier

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

Die unteren 11Bit des 32Bit Wertes enthalten den Identifier (0x80=128dez), das MSBit gibt Auskunft, ob das Gerät das SYNC-Telegramm empfängt (1) oder nicht (0).

Achtung: Im Gegensatz zu den PDO-Identifiern signalisiert das gesetzte MSB, dass dieser Identifier für den Knoten relevant ist.

### SYNC interval

Index	Sub index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1006	0	Communication cycle period	Unsigned32	rw	N	0x00000000	Maximum length of the SYNC interval in $\mu$ s.

Wenn hier ein Wert ungleich Null eingetragen wird, so geht der Koppler in den Fehlerzustand, wenn beim synchronen PDO-Betrieb innerhalb der "Watchdog-Zeit" kein SYNC-Telegramm empfangen wurde.

### Synchronous Window Length

Index	Sub index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1007	0	Synchronous window length	Unsigned32	rw	N	0x00000000	Contains the length of time window for synchronous PDOs in $\mu$ s.

### Device name

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

VIPA IM 253 1CA01 = VIPA CANopen-Slave IM253-1CA01

Da der zurückgelieferte Wert größer als 4Byte ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

---

**Hardware version**

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

VIPA IM 253 1CA01 = 1.00

Da der zurückgelieferte Wert größer als 4Byte ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

---

**Software version**

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

VIPA IM 253 1CA01 = 3.xx

Da der zurückgelieferte Wert größer als 4Byte ist, wird das segmentierte SDO-Protokoll zur Übertragung verwendet.

---

**Node number**

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

Die Knotennummer wird aus Kompatibilitätsgründen unterstützt.

---

**Guard time**

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

### Life time factor

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

Wenn innerhalb der Life Time kein Guarding-Telegramm empfangen wurde, geht der Knoten in den Fehlerzustand. Wenn "Life Time Factor" und/oder "Guard Time" = 0 sind, so führt der Knoten kein Lifeguarding durch, kann aber dennoch vom Master überwacht werden (Node Guarding).

### Guarding identifier

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

### Save parameters

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

Durch Schreiben der Signatur "save" im ASCII-Code (hex-Code: 0x65766173) auf Subindex 1 werden die aktuellen Parameter nicht-flüchtig gespeichert. (Bytefolge auf dem Bus incl. SDO-Protokoll: 0x23 0x10 0x10 0x01 0x73 0x61 0x76 0x65).

Ein erfolgreicher Speichervorgang wird durch das entsprechende TxSDO (0x60 im ersten Byte) bestätigt.



#### Hinweis!

Da der Buskoppler während des Speichervorgangs keine CAN-Telegramme senden und empfangen kann, kann nur gespeichert werden, wenn der Knoten im Zustand Pre-Operational ist.

Es wird empfohlen, vor dem Abspeichern das gesamte Netz in den Zustand Pre-Operational zu versetzen. Dadurch wird ein Puffer-Überlauf vermieden.

### Load default values

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

Durch Schreiben der Signatur "load" im ASCII-Code (hex-Code: 0x6461666C) auf Subindex 1 werden alle Parameter **beim nächsten Booten (Reset)** auf Default-Werte (Auslieferungszustand) zurückgesetzt. (Bytefolge auf dem Bus incl. SDO-Protokoll: 0x23 0x11 0x10 0x01 0x6C 0x6F 0x61 0x64).

Hierdurch werden die Default-Identifizierer für die PDOs wieder aktiv.

### Emergency COB-ID

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

### Consumer Heartbeat Time

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1016	0	Consumer heartbeat time	Unsigned8	ro	N	0x05	Number of entries
	1		Unsigned32	rw	N	0x00000000	Consumer heartbeat time

Struktur des "Consumer Heartbeat Time" Eintrags:

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

Sobald Sie versuchen, für die gleiche Node-ID eine "consumer heartbeat time" ungleich 0 zu konfigurieren, bricht der Knoten den SDO-Download ab und bringt den Fehlercode 0604 0043hex.

---

**Producer  
Heartbeat Time**

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

---

**Identity Object**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1018	0	Identity Object	Unsigned8	ro	N	0x04	Contains general Informations about the device (number of entries)
	1	Vendor ID	Unsigned32	ro	N	0xAFFEAF00	Vendor ID
	2	Product Code	Unsigned32	ro	N	0x2531CA01	Product Code
	3	Revision Number	Unsigned32	ro	N		Revision Number
	4	Serial Number	Unsigned32	ro	N		Serial Number

---

**Modular Devices**

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

## Modultypen

I/O-Modultyp	Identification (hex)	Nr. of Digital-Input-Bytes	Nr. of Analog-Input-Bytes	Nr. of Digital-Output-Bytes	Nr. of Analog-Output-Bytes
DI 8xDC24V	0x9FC1	1	-	-	-
DI 16xDC24V	0x9FC2	2	-	-	-
DI 14xDC24V/2C	0x08C0	2	4	-	6
DI 32xDC24V	0x9FC4	4	-	-	-
DO 8xDC24V	0xAFC8	-	-	1	-
DO 16xDC24V	0xAFD0	-	-	2	-
DO 32xDC24V	0xAFD8	-	-	4	-
DIO 8xDC24V	0xBFC9	1	-	1	-
DIO 16xDC24V	0xBFD2	2	-	2	-
AI2x12Bit	0x15C3	-	4	-	-
AI4x12Bit	0x15C4	-	8	-	-
AI8x12Bit	0x11C5	-	16	-	-
AO4x12Bit	0xA5E0	-	-	-	8
AI2/AO2x12Bit	0x45DB	-	4	-	4
CP 240	0x1CC1	16	-	16	-
FM 250	0xB5F4	-	10	-	10
FM 250-SSI	0xB5DB	-	4	-	4
FM 253, FM 254	0x18CB	-	12	-	12

## Error Behavior

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

Sobald im "operational" Status ein Gerätefehler entdeckt wird, sollte das Modul automatisch in den "pre-operational" Status übergehen.

Wenn beispielsweise "Error behavior" implementiert ist, kann das Modul so konfiguriert sein, dass es im Fehlerfall in den "stopped"-Status übergeht.

Folgende Fehlerklassen können angezeigt werden:

- 0 = pre-operational
- 1 = no state change
- 2 = stopped
- 3 = reset after 2 seconds



### Communication parameter RxPDO1

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

Subindex 1 (COB-ID): Die unteren 11Bit des 32Bit-Wertes (Bits 0-10) enthalten den CAN-Identifizier, das MSBit (Bit 31) gibt Auskunft, ob das PDO aktiv ist (1) oder nicht (0), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1).

Der Subindex 2 enthält die Übertragungsart.

### Communication parameter RxPDO2

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

### Communication parameter RxPDO3

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

**Communication parameter RxPDO4**

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

**Communication parameter RxPDO5**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1404	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000780 + NODE_ID	COB-ID RxPDO5
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

**Communication parameter RxPDO6**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1405	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000240 + NODE_ID	COB-ID RxPDO6
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

---

**Communication  
parameter RxPDO7**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1406	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000340 + NODE_ID	COB-ID RxPDO7
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

---

**Communication  
parameter RxPDO8**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1407	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000440 + NODE_ID	COB-ID RxPDO8
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

---

**Communication  
parameter RxPDO9**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1408	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000540 + NODE_ID	COB-ID RxPDO9
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO10

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1409	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC00007C0 + NODE_ID	COB-ID RxPD10
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Mapping RxPDO1

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

Das erste Empfangs-PDO (RxPDO1) ist per Default für digitale Ausgänge vorgesehen. Je nach Anzahl der bestückten Ausgänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt.

Da die digitalen Ausgänge byteweise organisiert sind, kann die Länge des PDOs in Bytes direkt dem Subindex 0 entnommen werden.

Wenn das Mapping verändert wird, so muss der Eintrag in Subindex 0 entsprechend angepasst werden.

### Mapping RxPDO2

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

Das 2. Empfangs-PDO (RxPDO2) ist per Default für analoge Ausgänge vorgesehen. Abhängig von der angeschlossenen Zahl von Ausgängen wird die notwendige Länge des PDOs automatisch festgelegt und die entsprechenden Objekte werden gemappt. Da die analogen Ausgänge wortweise organisiert sind, kann die Länge des PDO in Worten direkt aus dem Subindex 0 gelesen werden. Wird das Mapping verändert, muss auch der Eintrag im Subindex entsprechend geändert werden.

### Mapping RxPDO3- RxPDO10

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1602 - 0x1609	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the 3 rd to 10 th receive PDO; sub-index 0: number of mapped objects (2 byte index, 1 byte sub-index, 1 byte bit-width)
	1	1 st mapped object	Unsigned32	rw	N	0x00000000	
	2	2 nd mapped object	Unsigned32	rw	N	0x00000000	
	...	...	...	...	...	...	
	8	8 th mapped	Unsigned32	rw	N	0x00000000	

Die Empfangs-PDOs 3 bis 10 (RxPDO3) erhalten automatisch über den Koppler ein Default-Mapping, abhängig von den angeschlossenen Terminals. Der Vorgang wird unter "PDO-Mapping" näher beschrieben.

### Communication parameter TxPDO1

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

Subindex 1 (COB-ID): Die unteren 11Bit des 32Bit Wertes (Bits 0-10) enthalten den CAN-Identifizier, das MSBit (Bit 31) gibt Auskunft, ob das PDO aktiv ist (1) oder nicht (0), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1). Der Subindex 2 enthält die Übertragungsart, Subindex 3 die Wiederholungsverzögerung zwischen zwei gleichen PDOs. Wenn ein "Event Timer" mit einem Wert ungleich 0 existiert, wird nach Ablauf dieses Timers das PDO übertragen.

Existiert ein "Inhibit Timer", wird das Ereignis um diese Zeit verzögert.

### Communication parameter TxPDO2

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

### Communication parameter TxPDO3

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

### Communication parameter TxPDO4

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

### Communication parameter TxPDO5

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1804	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 5th transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000680 + NODE_ID	COB-ID TxPDO5
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO6

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1805	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 6th transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x800001C0 + NODE_ID	COB-ID TxPDO6
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO7

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1806	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 7th transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x800002C0 + NODE_ID	COB-ID TxPDO7
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO8

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1807	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 8th transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x800003C0 + NODE_ID	COB-ID TxPDO8
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]



### Communication parameter TxPDO9

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1808	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 9 <sup>th</sup> transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x800004C0 + NODE_ID	COB-ID TxPDO9
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO10

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1809	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 10 <sup>th</sup> transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x800006C0 + NODE_ID	COB-ID TxPDO10
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Mapping TxPDO1

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

*Fortsetzung ...*

### ... Fortsetzung Mapping TxPDO1

Das erste Sende-PDO (TxPDO1) ist per Default für digitale Eingänge vorgesehen. Je nach Anzahl der bestückten Eingänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt. Da die digitalen Eingänge byteweise organisiert sind, kann die Länge des PDOs in Bytes direkt dem Subindex 0 entnommen werden.

Wenn das Mapping verändert wird, muss der Eintrag in Subindex 0 entsprechend angepasst werden.

### Mapping TxPDO2

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

Das zweite Sende-PDO (TxPDO2) ist per Default für analoge Eingänge vorgesehen. Je nach Anzahl der bestückten Eingänge wird automatisch die erforderliche Länge des PDOs bestimmt und die entsprechenden Objekte gemappt. Da die digitalen Eingänge wortweise organisiert sind, kann die Länge des PDOs in Worten direkt dem Subindex 0 entnommen werden.

Wenn das Mapping verändert wird, muss der Eintrag in Subindex 0 entsprechend angepasst werden.

### Mapping TxPDO3- TxPDO10

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x1A02 - 0x1A09	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the 3 rd to 10 th transmit PDO; sub-index 0: number of mapped objects
	1	1 st mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2 nd mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8 th mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)

Die Sende-PDOs 3 bis 10 (TxPDO3-10) erhalten automatisch über den Koppler ein Default-Mapping, abhängig von den angeschlossenen Terminals. Der Vorgang wird unter "PDO-Mapping" näher beschrieben.

### CAN-Baudrate

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x2001	0	CAN-Baudrate	Unsigned8	rw	N	0x01	Setting CAN-Baudrate

Dieser Indexeintrag schreibt eine neue Baudrate in das EEPROM. Beim nächsten Bootvorgang (Reset) startet der CAN-Koppler mit der neuen Baudrate.

Wert	CAN-Baudrate
"00"	1 MBaud
"01"	500 kBaud
"02"	250 kBaud
"03"	125 kBaud
"04"	100 kBaud
"05"	50 kBaud
"06"	20 kBaud
"07"	10 kBaud
"08"	800 kBaud

**KILL EEPROM**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x2100	0	KILL EEPROM	Boolean	wo	N		KILL EEPROM

Das KILL EEPROM wird aus Gründen der Kompatibilität unterstützt.

Das Schreiben in den Index 0x2100 löscht alle gespeicherten Identifier aus dem EEPROM.

Der CANopen-Koppler startet **beim nächsten Hochfahren (reset)** mit der Default-Konfiguration.

**SJA1000****Message Filter**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x2101	0	Number of Elements	Unsigned8	ro	N	0x02	SJA1000 Message Filter
	1	Acceptance mask	Unsigned8	ro	N		Acceptance mask
	2	Acceptance code	Unsigned8	ro	N		Acceptance code

Mit Hilfe des Acceptance-Filters ist der CAN-Controller in der Lage, empfangene Nachrichten nur dann an den RXFIFO weiterzuleiten, wenn die Identifier-Bits der empfangenen Nachricht den vorher im Acceptance-Filter definierten entsprechen. Der Acceptance-Filter wird über das Acceptance-Coderegister und das Acceptance-Maskregister definiert.

Diese Filter werden nach dem Hochfahren und nach einem Kommunikationsreset aktualisiert.

**Acceptance-Mask:** Das Acceptance-Maskregister legt fest, welche der entsprechenden Bits des Acceptance-Codes relevant ( $AM.X = 0$ ) und welche 'don't care' ( $AM.X = 1$ ) für das Filtern sind.

**Acceptance-Code:** Die Acceptance-Code-Bits (AC.7 bis AC.0) und die 8 wichtigsten Bits des Nachrichtenidentifiers (ID.10 bis ID.3) müssen an den Stellen stehen, die durch die Acceptance-Mask-Bits als relevant gekennzeichnet wurden (AM.7 bis AM.0). Wenn die folgenden Bedingungen erfüllt werden, werden die Nachrichten akzeptiert:

$$0(\text{ID.10 bis ID.3}) \equiv (\text{AC.7 bis AC.0}) \vee (\text{AM.7 bis AM.0}) \equiv 11111111$$

**PDO-Control**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x2400	0	Number of Elements	Unsigned8	ro	N	0x0A	Time control for RxPDOs
	1	RxPDO1	Unsigned16	rw	N	0x0000	Timer value [ms]
	2	RxPDO2	Unsigned16	rw	N	0x0000	Timer value [ms]
	...	...	...	...	...	...	...
	10	RxPDO10	Unsigned16	rw	N	0x0000	Timer value [ms]

Sobald der Timerwert ungleich 0 ist, startet die Kontrolle. Mit jedem empfangenen RxPDO wird der Timer wieder zurückgesetzt.

Sobald der Timer abgelaufen ist, geht der CAN-Koppler in den Zustand "pre-operational" über und schickt ein Emergency-Telegramm.

**Module  
Parameterization**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x3001 - 0x3010	0	Number of Elements	Unsigned8	ro	N	0x04 or 0x00	Number of entries 0x04 : module available 0x00 : no module available
	1	Prm 1 to 4	Unsigned32	rw	N	depending on the components fitted	Parameter bytes 1 to 4
	2	Prm 5 to 8	Unsigned32	rw	N	depending on the components fitted	Parameter bytes 5 to 8
	3	Prm 9 to 12	Unsigned32	rw	N	depending on the components fitted	Parameter bytes 9 to 12
	4	Prm 13 to 16	Unsigned32	rw	N	depending on the components fitted	Parameter bytes 13 to 16

Über die Indizes 0x3001 bis 0x3010 können die Analogmodule, Zähler und Kommunikationsmodule parametrisiert werden.

**Default configuration**

AI4	0x00, 0x00, 0x28, 0x28, 0x28, 0x28, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
AI8	0x00, 0x00, 0x26, 0x26, 0x26, 0x26, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
AO4	0x00, 0x00, 0x09, 0x09, 0x09, 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
AI/AO	0x00, 0x00, 0x09, 0x09, 0x09, 0x09, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
CP 240	0x00, 0x00, 0x00, 0x00, 0x00, 0x13, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
FM 250	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
FM 254	0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

**Beispiel 1                      Set AI4 to mode 0x2C**

**Read default configuration**

```

Read SubIndex 0  M2S: 0x40 0x01 0x30 0x00 0x00 0x00 0x00 0x00
                  S2M: 0x4F 0x01 0x30 0x00 0x04 0x00 0x00 0x00
Read SubIndex 1  M2S: 0x40 0x01 0x30 0x01 0x00 0x00 0x00 0x00
                  S2M: 0x43 0x01 0x30 0x01 0x00 0x00 0x28 0x28
Read SubIndex 2  M2S: 0x40 0x01 0x30 0x02 0x00 0x00 0x00 0x00
                  S2M: 0x43 0x01 0x30 0x02 0x28 0x28 0x00 0x00
Read SubIndex 3  M2S: 0x40 0x01 0x30 0x03 0x00 0x00 0x00 0x00
                  S2M: 0x43 0x01 0x30 0x03 0x00 0x00 0x00 0x00
Read SubIndex 4  M2S: 0x40 0x01 0x30 0x04 0x00 0x00 0x00 0x00
                  S2M: 0x43 0x01 0x30 0x04 0x00 0x00 0x00 0x00
    
```

**Write new configuration**

```

Write SubIndex 1 M2S: 0x23 0x01 0x30 0x01 0x00 0x00 0x2C 0x2C
                  S2M: 0x60 0x01 0x30 0x01 0x00 0x00 0x00 0x00
Write SubIndex 2 M2S: 0x23 0x01 0x30 0x02 0x2C 0x2C 0x00 0x00
                  S2M: 0x60 0x01 0x30 0x02 0x00 0x00 0x00 0x00
    
```

**Read new configuration**

```

Read SubIndex 0  M2S: 0x40 0x01 0x30 0x00 0x00 0x00 0x00 0x00
                  S2M: 0x4F 0x01 0x30 0x00 0x04 0x00 0x00 0x00
Read SubIndex 1  M2S: 0x40 0x01 0x30 0x01 0x00 0x00 0x00 0x00
                  S2M: 0x43 0x01 0x30 0x01 0x00 0x00 0x2C 0x2C
Read SubIndex 2  M2S: 0x40 0x01 0x30 0x02 0x00 0x00 0x00 0x00
                  S2M: 0x43 0x01 0x30 0x02 0x2C 0x2C 0x00 0x00
Read SubIndex 3  M2S: 0x40 0x01 0x30 0x03 0x00 0x00 0x00 0x00
                  S2M: 0x43 0x01 0x30 0x03 0x00 0x00 0x00 0x00
Read SubIndex 4  M2S: 0x40 0x01 0x30 0x04 0x00 0x00 0x00 0x00
                  S2M: 0x43 0x01 0x30 0x04 0x00 0x00 0x00 0x00
    
```

**Beispiel 2****Set FM250 to Counter Mode 0x08 and 0x0B****Read default configuration**

Read SubIndex 0 M2S: 0x40 0x02 0x30 0x00 0x00 0x00 0x00 0x00  
S2M: 0x4F 0x02 0x30 0x00 0x04 0x00 0x00 0x00

Read SubIndex 1 M2S: 0x40 0x02 0x30 0x01 0x00 0x00 0x00 0x00  
S2M: 0x43 0x02 0x30 0x01 0x00 0x00 0x00 0x00

**Write new configuration**

Write SubIndex 1 M2S: 0x23 0x02 0x30 0x01 0x08 0x0B 0x00 0x00  
S2M: 0x60 0x02 0x30 0x01 0x00 0x00 0x00 0x00

**Read new configuration**

Read SubIndex 0 M2S: 0x40 0x02 0x30 0x00 0x00 0x00 0x00 0x00  
S2M: 0x4F 0x02 0x30 0x00 0x04 0x00 0x00 0x00

Read SubIndex 1 M2S: 0x40 0x02 0x30 0x01 0x00 0x00 0x00 0x00  
S2M: 0x43 0x02 0x30 0x01 0x08 0x0B 0x00 0x00

**Module  
Parameterization**

Index	Sub-index	Name	Typ	Attr.	Map.	Default value	Meaning
0x3401	0x00	Number of Elements	Unsigned8	ro	N	depending on the components fitted	Number of Entries
	0x01	1 st mapped object	Unsigned32	rw	N		
	...	...	...	...	...		
	0x40	8 th mapped object	Unsigned32	rw	N		

Der Index 0x3401 wird aus Kompatibilitätsgründen unterstützt.

Benutzen Sie Index 3001 bis 3010 für neue Projekte.

Alternativ können Sie analoge Parameter auch über folgende Indizes schreiben/lesen.

Subindex 0...0x40 (256 bytes):

Subindex 0: Anzahl der Subindizes

Subindex 1: Parameterbyte 0 ... 3

...

Subindex 0x20: Parameterbyte 124 ... 127

Jeder Subindex besteht aus 2 Datenworten. Geben Sie hier Ihre Parameterbytes an. Jedes analoge Eingangs- oder Ausgangsmodul hat 16Byte Parameterdaten, d.h. sie belegen 4 Subindizes, z.B.:

1. Analogmodul Subindizes 1 bis 4,
2. Analogmodul Subindizes 5 bis 8,
3. Analogmodul Subindizes 9 bis 12.

**8bit Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6000	0x00	8-bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8-bit input blocks
	0x01	1 st input block	Unsigned8	ro	Y		
	...	...	...	...	...		
	0x48	27 st input block	Unsigned8	ro	Y		



### 8bit Polarity Digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6002	0x00	8-bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8-bit input blocks
	0x01	1 st input block	Unsigned8	rw	N	0x00	1 st polarity digital input block
	...	...	...	...	...	...	...
	0x48	72 nd input block	Unsigned8	rw	N	0x00	72 nd polarity digital input block

Individuelle Invertierung der Eingangskanäle

1 = Eingang invertiert

0 = Eingang nicht invertiert

### 16bit Digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6100	0x00	16-bit digital input block	Unsigned8	ro	N	depending on the fitted components	Number of available digital 16-bit input blocks
	0x01	1 st input block	Unsigned16	ro	N		1 st digital input block
	...	...	...	...	...	...	...
	0x24	36 nd input block	Unsigned16	ro	N		36 nd digital input block

**16bit Polarity  
Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6102	0x00	16-bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 16-bit input blocks
	0x01	1 st input block	Unsigned16	rw	N	0x0000	1 st polarity digital input block
	...	...	...	...	...	...	...
	0x24	36 input block	Unsigned16	rw	N	0x0000	36 polarity digital input block

Individuelle Invertierung der Eingangspolarität

1 = Input invertiert

0 = Input nicht invertiert

**32bit Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6120	0x00	32-bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 32-bit input blocks
	0x01	1 st input block	Unsigned32	ro	N		1 st digital input block
	...	...	...	...	...	...	...
	0x12	18 input block	Unsigned32	ro	N		18 digital input block

### 32bit Polarity Digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6122	0x00	8-bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 32-bit input blocks
	0x01	1 st input block	Unsigned32	rw	N	0x00000000	1 st polarity digital input block
	...	...	...	...	...	...	...
	0x12	18 input block	Unsigned32	rw	N	0x00000000	18 polarity digital input block

Individuelle Invertierung der Eingangspolarität

1 = Input invertiert

0 = Input nicht invertiert

### 8bit Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6200	0x00	8-bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8-bit output blocks
	0x01	1 st output block	Unsigned8	rw	Y		1 st digital output block
	...	...	...	...	...	...	...
	0x48	72 nd output block	Unsigned8	rw	Y		72 nd digital output block

### 8bit Change Polarity Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6202	0x00	8-bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8-bit output blocks
	0x01	1 st output block	Unsigned8	rw	N	0x00	1 st polarity digital output block
	...	...	...	...	...	...	...
	0x48	72 nd output block	Unsigned8	rw	N	0x00	72 nd polarity digital output block

Individuelle Invertierung der Ausgangskanäle

1 = Ausgang invertiert

0 = Ausgang nicht invertiert

### 8bit Error Mode Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6206	0x00	8-bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8-bit output blocks
	0x01	1 st output block	Unsigned8	rw	N	0xFF	1 st error mode digital output block
	...	...	...	...	...	...	...
	0x48	72 nd output block	Unsigned8	rw	N	0xFF	72 nd error mode digital output block

Mit diesem Objekt können Sie bestimmen, ob im Fehlerfall ein Ausgabe-Kanal einen bestimmt Wert annimmt, den Sie im Objekt 0x6207 vorgeben.

1 = den Wert in Objekt 0x6207 übernehmen

0 = Ausgabewert im Fehlerfall fixieren

### 8bit Error Value Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6207	0x00	8-bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8-bit output blocks
	0x01	1 st output block	Unsigned8	rw	N	0x00	1 st error value digital output block
	...	...	...	...	...	...	...
	0x48	72 nd output block	Unsigned8	rw	N	0x00	72 nd error value digital output block

Vorausgesetzt der Error Mode ist aktiviert, wird im Fehlerfall der hier vorgegebene Wert übernommen.

1 = Im Fehlerfall Ausgabewert auf 0 sobald Objekt 0x6206 aktiviert ist.

0 = Im Fehlerfall Ausgabewert auf 1 sobald Objekt 0x6206 aktiviert ist.

### 16bit Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6300	0x00	16-bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16-bit output blocks
	0x01	1 st output block	Unsigned16	rw	N		1 st digital output block
	...	...	...	...	...	...	...
	0x24	36 th output block	Unsigned16	rw	N		36 th digital output block

**16bit Change  
Polarity Digital  
outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6302	0x00	16-bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16-bit output blocks
	0x01	1 st output block	Unsigned16	rw	N	0x0000	1 st polarity digital output block
	... 0x24	... 36 th output block	... Unsigned16	... rw	... N	... 0x0000	... 36 th polarity output block

Die Ausgangspolarität kann individuell invertiert werden.

1 = Output invertiert

0 = Output nicht invertiert

**16bit Error Mode  
Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6306	0x00	16-bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16-bit output blocks
	0x01	1 st output block	Unsigned16	rw	N	0xFFFF	1 st error mode digital output block
	... 0x24	... 36 th output block	... Unsigned16	... rw	... N	... 0xFFFF	... 36 th error mode digital output block

Dieses Objekt zeigt an, ob ein Ausgang im Falle eines internen Gerätefehlers einen vordefinierten Fehlerwert annimmt (s. Objekt 6207).

1 = Ausgangswert nimmt vordefinierten Wert aus Objekt 6207

0 = Ausgangswert bleibt im Falle eines Fehlers erhalten

### 16bit Error Value Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6307	0x00	16-bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16-bit output blocks
	0x01	1 st output block	Unsigned16	rw	N	0x0000	1 st error value digital output block
	...	...	...	...	...	...	...
	0x24	36 th output block	Unsigned16	rw	N	0x0000	36 th error value digital output block

Vorausgesetzt der entsprechende ErrorMode ist aktiviert, setzen Gerätefehler den Ausgang auf den Wert, der durch dieses Objekt definiert wird.

1 = Der Ausgang wird im Fehlerfall auf '0' gesetzt (Objekt 6206 aktiv)

0 = Der Ausgang wird im Fehlerfall auf '1' gesetzt (Objekt 6206 aktiv)

### 32bit Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6320	0x00	32-bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 32-bit output blocks
	0x01	1 st output block	Unsigned32	rw	N		1 st digital output block
	...	...	...	...	...	...	...
	0x12	18 output block	Unsigned32	rw	N		18 digital output block

### 32bit Change Polarity Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6322	0x00	32-bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 32-bit output blocks
	0x01	1 st output block	Unsigned32	rw	N	0x00000000	1 st polarity digital output block
	...	...	...	...	...	...	...
	0x12	18 output block	Unsigned32	rw	N	0x00000000	18 polarity output block

Die Ausgangspolarität kann individuell invertiert werden.

1 = Output invertiert

0 = Output nicht invertiert

### 32bit Error Mode Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6326	0x00	32-bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 32-bit output blocks
	0x01	1 st output block	Unsigned32	rw	N	0xFFFFFFFF	1 st error mode digital output block
	...	...	...	...	...	...	...
	0x48	18 output block	Unsigned32	rw	N	0xFFFFFFFF	18 error mode digital output block

Dieses Objekt zeigt an, ob ein Ausgang auf einen vordefinierten Fehlerwert gesetzt wird (s. Objekt 6207), falls ein interner Gerätefehler auftritt.

1 = Ausgangswert übernimmt den in Objekt 6207 definierten Wert an

0 = Ausgangswert wird im Falle eines Fehlers erhalten



### 32bit Error Value Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6237	0x00	32-bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 32-bit output blocks
	0x01	1 st output block	Unsigned32	rw	N		1 st error value digital output block
	...	...	...	...	...		...
	0x12	18 output block	Unsigned32	rw	N		18 error value digital output block

Vorausgesetzt der entsprechende ErrorMode ist aktiviert, setzen Gerätefehler den Ausgang auf den Wert, der durch dieses Objekt definiert wird.

1 = Der Ausgang wird im Fehlerfall auf '0' gesetzt (Objekt 6206 aktiv)

0 = Der Ausgang wird im Fehlerfall auf '1' gesetzt (Objekt 6206 aktiv)

### Analog inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6401	0x00	2 byte input block	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	1 st input channel	Unsigned16	ro	Y		1 st analog input channel
	...	...	...	...	...		...
	0x24	24 th input channel	Unsigned16	ro	Y		24 th analog input channel

### Analog outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6411	0x00	2 byte output block	Unsigned8	ro	N	depending on the components fitted	Number of available analog outputs
	0x01	1 st output channel	Unsigned16	ro	Y		1 st analog output channel
	...	...	...	...	...		...
	0x24	24 th output channel	Unsigned16	ro	Y		24 th analog output channel

**Analog Input  
Interrupt Trigger  
selection**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6421	0x00	Number of Inputs	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	Trigger 1 st input channel	Unsigned8	rw	N	0x07	Input interrupt trigger for 1 st analog input channel
	...	...	...	...	...	...	...
	0x24	Trigger 24 th input channel	Unsigned8	rw	N	0x07	Input interrupt trigger for 24 th analog input channel

Dieses Objekt legt fest, welches Ereignis eine Unterbrechung eines bestimmten Kanals auslösen soll. Die gesetzten Bits der untenstehenden Liste verweisen auf den Unterbrechungstrigger.

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

**Analog Input  
Interrupt Source**

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

Dieses Objekt legt fest, welcher Kanal die Unterbrechung verursacht hat. Gesetzte Bits verweisen auf die Nummer des Kanals, der die Unterbrechung verursacht hat. Die Bits werden automatisch zurückgesetzt, nachdem sie von einem SDO gelesen oder durch ein PDO versandt wurden.

- 1 = Unterbrechung verursacht
- 0 = keine Unterbrechung verursacht

### Event driven analog inputs

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

Obwohl die analogen Eingänge im TxPDO2 in Übereinstimmung mit CANopen per default auf den Übertragungstyp 255 (ereignisgesteuert) gesetzt werden, wird das "Ereignis" (die Änderung eines Eingangswertes) durch die Ereigniskontrolle in Objekt 0x6423 unterdrückt, um den Bus nicht mit analogen Signalen zu überschwemmen.

Vor der Aktivierung ist es sinnvoll, das Übertragungsverhalten der analogen PDOs zu parametrieren, indem man eine Inhibit-Zeit festlegt (Objekt 0x1800ff, Subindex 3) und/oder eine Grenzwertüberwachung (Objekte 0x6424 + 0x6425) und/oder eine Deltafunktion (Objekt 0x6426).

### Upper limit value analog inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6424	0x00	Number of Inputs	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	Upper limit 1 st input channel	Unsigned32	rw	N	0x00000000	Upper limit value for 1 st analog input channel
	...	...	...	...	...	...	...
	0x24	Upper limit 24 th input channel	Unsigned32	rw	N	0x00000000	Upper limit value for 24 th analog input channel

Werte ungleich 0 aktivieren den Obergrenzenwert für diesen Kanal. Ein PDO wird dann übertragen, wenn diese Obergrenze überschritten wird. Zusätzlich muss die Ereignissteuerung aktiviert sein (Objekt 0x6423). Das Datenformat korrespondiert zu dem der analogen Eingänge.

**Lower limit value analog inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6425	0x00	Number of Inputs	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	Lower limit 1 st input channel	Unsigned32	rw	N	0x00000000	Lower limit value for 1 st analog input channel
	...	...	...	...	...	...	...
	0x24	Lower limit 24 th input channel	Unsigned32	rw	N	0x00000000	Lower limit value for 24 th analog input channel

Werte ungleich 0 aktivieren den Untergrenzenwert für diesen Kanal. Ein PDO wird dann übertragen, wenn diese Untergrenze unterschritten wird. Zusätzlich muss die Ereignissteuerung aktiviert sein (Objekt 0x6423). Das Datenformat korrespondiert zu dem der analogen Eingänge.

**Delta function**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6426	0x00	Number of Inputs	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	Delta value 1 st input channel	Unsigned32	rw	N	0x00000002	Delta value for 1 st analog input channel
	...	...	...	...	...	...	...
	0x24	Delta value 24 th input channel	Unsigned32	rw	N	0x00000002	Delta value for 24 th analog input channel

Werte ungleich 0 aktivieren die Deltafunktion für diesen Kanal. Ein PDO wird dann übertragen, wenn sich der Wert seit der letzten Übertragung um mehr als den Deltawert verändert hat. Zusätzlich muss die Ereignissteuerung aktiviert sein (Objekt 0x6423). Das Datenformat korrespondiert zu dem der analogen Eingänge (Der Deltawert kann nur positive Werte annehmen).

### Analog Output Error Mode

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6443	0x00	Analog output block	Unsigned8	ro	N	Depending on the components fitted	Number of available analog outputs
	0x01	1 st analog output block	Unsigned8	rw	N	0xFF	1 st error mode analog output block
	...	...	...	...	...	...	...
	0x24	36 th analog output block	Unsigned8	rw	N	0xFF	36 th error mode analog output block

Dieses Objekt legt fest, ob ein Ausgang im Falle eines internen Gerätefehlers auf einen bestimmten Fehlerwert gesetzt wird (s. Objekt 0x6444).

0 = Aktueller Wert

1 = fällt auf Fehlerwert 0x6444

### Analog Output Error Value

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6444	0x00	16-bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available analog output blocks
	0x01	1 st analog block	Unsigned16	rw	N	0x0000	1 st analog output block
	...	...	...	...	...	...	...
	0x24	36 th analog block	Unsigned16	rw	N	0x0000	36 th analog output block

Unter der Bedingung, dass der zugehörige Fehler aktiviert ist (0x6443), setzen Gerätefehler die Ausgänge auf den Wert, der hier konfiguriert wird.

**SDO Abort Codes**

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

## Emergency Object

### Übersicht

Um anderen Teilnehmern am CANopen-Bus interne Gerätefehler oder CAN-Busfehler mitteilen zu können, verfügt der CANopen Buskoppler über das Emergency-Object. Es ist mit einer hohen Priorität versehen und liefert wertvolle Informationen über den Zustand des Gerätes und des Netzes.



### Hinweis!

Es wird dringend empfohlen, das Emergency Object auszuwerten - es stellt eine wertvolle Informationsquelle dar!

### Telegramm- Aufbau

Das Emergency-Telegramm ist immer 8Byte lang. Es enthält zunächst den 2Byte Error Code, dann das 1Byte Error Register und schließlich den 5Byte großen Additional Code.

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

### Fehlermeldungen

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

## NMT - Netzwerk Management

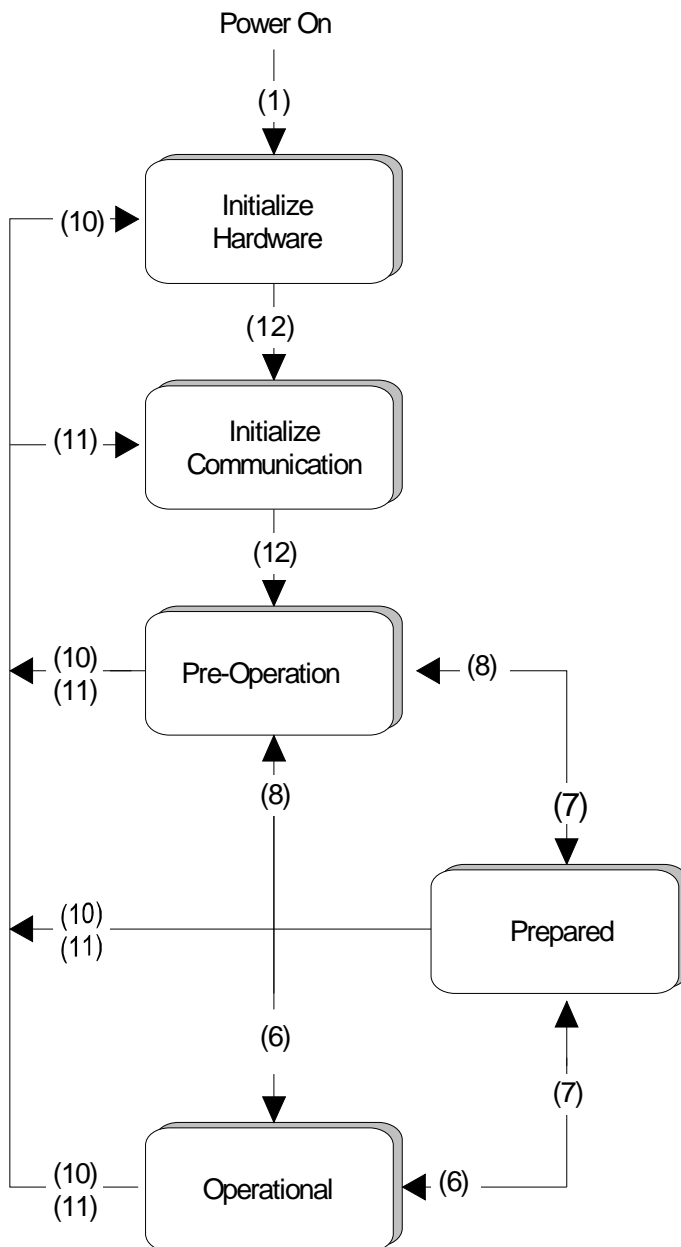
Das Netzwerkmanagement (NMT) spezifiziert globale Dienste für Netzwerküberwachung und -Management. Dazu gehört neben dem An- und Abmelden einzelner Teilnehmer auch die Überwachung der Teilnehmer während des Betriebs- und die Behandlung von Ausnahmezuständen.

NMT-Service-Telegramme haben den COB-Identifizier 0x0000. Eine additive Modul-ID ist nicht erforderlich. Die Länge beträgt immer 2 Datenbytes.

Das 1. Datenbyte enthält den NMT-Command Specifier: **CS**.

Das 2. Datenbyte enthält die Modul-ID (0x00 für ein Broadcast Command).

Die nachfolgende Abbildung gibt einen Überblick über alle CANopen Statusübergänge und die dazugehörigen NMT-Command Specifier "CS":



- (1): Der Initialisierungs-Status wird beim Einschalten selbsttätig erreicht.
- (6): "Start\_Remote\_Node" (CS:0x01)  
Startet Modul, gibt Ausgänge frei und startet Übertragung von PDOs.
- (7): "Stop\_Remote\_Node" (CS:0x02)  
Ausgänge gehen in den Fehlerzustand und SDO und PDO werden abgeschaltet.
- (8): "Enter\_Pre-operational\_State" (CS:0x80)  
Stoppt PDO-Übertragung, SDO weiter aktiv.
- (10): "Reset\_Node" (CS:0x81)  
Führt Reset durch. Alle Objekte werden auf Power-On Defaults zurückgesetzt.
- (11): "Reset\_Communication" (CS:0x82)  
Führt Reset der Kommunikationsfunktionen durch. Objekte 0x1000 - 0x1FFF werden auf Power-On Defaults zurückgesetzt.
- (12): Nach der Initialisierung wird der Status Pre-Operational automatisch erreicht - dabei wird die Boot-Up-Nachricht abgeschickt.



---

**Node Guarding**

Der Buskoppler unterstützt das von CANopen definierte Node Guarding um die Überwachung der Busteilnehmer zu gewährleisten.

Der Guarding-Betrieb des Moduls startet mit dem ersten, vom Master empfangenen Guarding-Anforderungstelegramm (RTR). Der zugehörige COB-Identifizier ist im Objektverzeichnis in der Variablen 0x100E fest auf 0x700 + Modul-ID eingestellt. Wird während des Guardingbetriebs innerhalb der "Guard-Time" (Objekt 0x100C) kein Guarding-Anforderungstelegramm mehr vom Master empfangen, so geht das Modul davon aus, dass der Master nicht mehr korrekt arbeitet. Nach der Zeit, die durch das Produkt aus "Guard-Time" (0x100C) und "Life-Time-Factor" (0x100D) eingestellt ist, versetzt sich das Modul automatisch in den Zustand "Pre-Operational".

Wird entweder die "Guard-Time" (Objekt 0x100C) oder der "Life-Time-Factor" (0x100D) mittels SDO-Download vom Master auf Null eingestellt, so findet keine Überprüfung auf Ablauf der Guardingzeit statt, und das Modul bleibt im aktuellen Zustand.

---

**Heartbeat**

Neben dem Node Guarding unterstützt der VIPA CAN-Koppler den Heartbeat Mode.

Wird im Index 0x1017 (Heartbeat Producer Time) ein Wert eingetragen, so wird mit Ablauf des Heartbeat-Timers der Gerätezustand (Operational, Pre-Operational, ...) des Buskopplers mittels COB-Identifizier (0x700 + Modul-ID) übertragen.

Der Heartbeat Mode startet automatisch sobald im Index 0x1017 ein Wert größer 0 eingetragen ist.

## Technische Daten

### CANopen-Koppler IM 253CAN

Elektrische Daten	VIPA 253-1CA01
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme	max. 700mA
Ausgangsstrom Rückwandbus	max. 3,5A
Potenzialtrennung	≥ AC 500V
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse/Schnittstellen	9poliger SubD (Stecker) CAN-Bus-Ankopplung
CAN-Bus Schnittstelle	
Ankopplung	9poliger SubD-Stecker
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an einem Ende, Stichleitungen sind möglich.
Medium	Abgeschirmtes dreiadriges Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen.
Übertragungsrate	10kBaud bis 1MBaud
max. Gesamtlänge	ohne Repeater 1000m bei 50kBaud
digitale Ein-/Ausgänge	Je Koppler maximal 32 E/A-Module frei kombinierbar.
max. Teilnehmeranzahl	127 Stationen (je nach Masteranschlutung)
Kombination mit Peripheriemodulen	
max. Modulanzahl	32 (abhängig von der Stromaufnahme)
max. Eingänge/Ausgänge	je 80Byte (80Byte = 10 PDOs à 8Byte)
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g

**CANopen-Koppler**  
**IM 253CAN,**  
**DO 24xDC 24V**

Elektrische Daten	VIPA 253-2CA20
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme an L+	max. 800mA
Ausgangsstrom Rückwandbus	max. 3,5A
Potenzialtrennung	≥ AC 500V
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse/Schnittstellen	9poliger SubD (Stecker) CAN-Bus-Ankopplung
CAN-Bus Schnittstelle	
Ankopplung	9poliger SubD-Stecker
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an einem Ende, Stichleitungen sind möglich.
Medium	Abgeschirmtes dreiadriges Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen.
Übertragungsrate	10kBaude bis 1Mbaude
max. Gesamtlänge	ohne Repeater 1000m bei 50kBaude
max. Teilnehmeranzahl	127 Stationen (je nach Masteranschaltung)
Ausgabe-Einheit	
Anzahl der Ausgänge	24
Nennlastspannung	DC 24V (18...35V) intern über CAN-Koppler versorgt
Ausgangsstrom je Kanal	0,5A (Summenstrom max. 4A)
Statusanzeige	Power (PW) Sicherung intakt, Error (ER) Kurzschluss, Überlast
Programmierdaten	
Ausgabedaten	3Byte
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x76
Gewicht	150g



## Teil 5 DeviceNet

### Überblick

Inhalt dieses Kapitels ist die Beschreibung des DeviceNet-Slaves von VIPA. Nach einer Systemvorstellung folgt die Beschreibung des Moduls. Ein weiterer Bestandteil dieses Kapitels ist die Projektierung unter Einsatz des *DeviceNet-Manager* der Firma Allen - Bradley. Hier wird anhand von Beispielen die Projektierung des DeviceNet-Kopplers und die Parametrierung der System 200V Module beschrieben.

Mit einer Übersicht der Diagnosemeldungen, der Vorgehensweise, wie Sie Ihren DeviceNet-Koppler an Profibus anbinden können und den Technischen Daten endet das Kapitel.

Nachfolgend sind beschrieben:

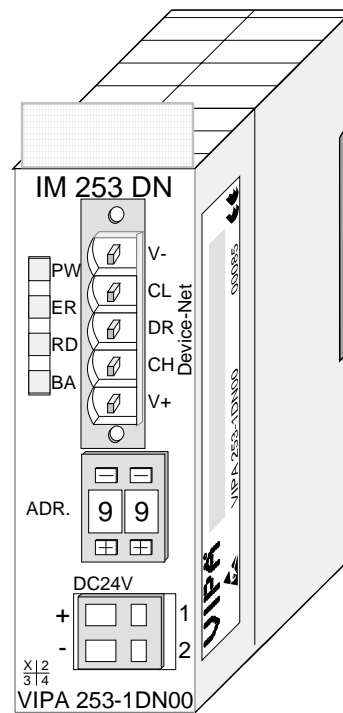
- DeviceNet-Grundlagen
- Hardwarebeschreibung des DeviceNet-Kopplers IM 253DN von VIPA
- Projektierung im *DeviceNet-Manager* mit Beispielen
- Diagnose
- Anbindungsmöglichkeit an Profibus
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 5 DeviceNet .....</b>	<b>5-1</b>
Systemübersicht .....	5-2
Grundlagen.....	5-3
IM 253DN - DeviceNet-Koppler - Aufbau .....	5-5
Projektierung unter Einsatz des DeviceNet-Managers .....	5-8
Einstellung von Baudrate und DeviceNet-Adresse .....	5-9
Test am DeviceNet-Bus .....	5-10
Module im DeviceNet-Manager parametrieren.....	5-11
I/O-Adressierung des DeviceNet-Scanners .....	5-16
Diagnose .....	5-17
Schnittstelle zum Profibus.....	5-22
Technische Daten.....	5-23

## Systemübersicht

Mit dem DeviceNet-Koppler von VIPA können Sie bis zu 32 Module (je 40Byte) Ihrer System 200V Peripherie an DeviceNet ankoppeln. Folgende DeviceNet-Komponenten sind zur Zeit von VIPA verfügbar.



**Bestelldaten  
DeviceNet**

Typ	Bestellnummer	Beschreibung
IM 253DN	VIPA 253-1DN00	DeviceNet-Koppler

## Grundlagen

### Allgemeines

DeviceNet ist ein offenes Low-End-Netzwerk, das auf der CAN-Bus-Physik basiert. Zusätzlich wird über den Bus die DC 24V Stromversorgung mitgeführt.

Über DeviceNet können Sie direkte Verbindungen zwischen einfachen Industriegeräten wie Sensoren und Schaltern und technisch hochentwickelten Geräten wie Frequenzumformer und Bar-Code-Lesegeräten zu ihrem Steuerungssystem herstellen.

Diese direkte Anbindung ermöglicht eine bessere Kommunikation zwischen den Geräten, sowie wichtige Diagnosemöglichkeiten auf Geräteebene.

### DeviceNet

DeviceNet ist ein offener Gerätenetzwerk-Standard, der das Anwenderprofil für den Bereich industrieller Echtzeitsysteme erfüllt.

Die Spezifikation und das Protokoll sind offen. Die Spezifikation besitzt und verwaltet die unabhängige Anbieterorganisation "Open DeviceNet Vendor Association" ODVA.

Hier werden auch standardisierte Geräteprofile erstellt, die eine logische Austauschbarkeit unter einfachen Geräten desselben Gerätetyps ermöglichen.

Im Gegensatz zum klassischen Quelle-Ziel-Modell verwendet DeviceNet das moderne Produzenten/Konsumenten-Modell, das Datenpakete mit Identifier-Feldern zur Identifizierung der beigefügten Daten erfordert.

Dies erlaubt mehrere Prioritätsebenen, eine effizientere Übertragung von E/A-Daten und mehrere Datenkonsumenten.

Ein sendewilliges Gerät *produziert* die Daten mit einem Identifier auf dem Netzwerk. Alle Geräte, die Daten benötigen, hören auf Meldungen. Erkennen Geräte einen geeigneten Identifier, agieren Sie und *konsumieren* somit die Daten.

Über DeviceNet werden zwei Arten von Meldungen transportiert:

- *E/A-Meldungen*

Meldungen für zeitkritische und steuerungorientierte Daten, die in einzelnen oder mehrfachen Verbindungen ausgetauscht werden und Identifier mit hoher Priorität verwenden.

- *Explizite Meldungen*

Hiermit werden Mehrzweck-Punkt-zu-Punkt-Kommunikationspfade zwischen zwei Geräten aufgebaut. Diese kommen bei der Konfiguration der Netzkoppler und bei Diagnosen zum Einsatz. Hierfür werden in der Regel Identifier mit niedriger Priorität verwendet.

Bei Meldungen, die länger als 8Byte sind tritt der Fragmentierungsdienst in Kraft. Regeln für Master/Slave-, Peer-to-Peer- und Multi-Master-Anschaltungen werden ebenfalls bereitgestellt.

**Übertragungs-  
medium**

DeviceNet verwendet eine Stammleitungs-/Stichleitungs-Topologie mit bis zu 64 Netzknoten. Die maximale Länge beträgt entweder 500m bei 125kBaud, 250m bei 250kBaud oder 100m bei 500kBaud.

Die Stichleitungen können bis zu 6m lang sein, wobei der Gesamtumfang aller Stichleitungen von der Baudrate abhängt.

Netzknoten können ohne Unterbrechung des Netzwerks entfernt oder hinzugefügt werden. Es wird automatisch erkannt, ob ein Teilnehmer ausgefallen oder neu am Netz ist.

DeviceNet verwendet als Übertragungsmedium eine abgeschirmte Fünfdrahtleitung.

DeviceNet arbeitet mit Spannungsdifferenzen und ist daher unempfindlicher gegenüber Störeinflüssen als eine Spannungs- oder Stromschnittstelle.

Signale und Stromversorgung laufen über das Netzkabel. Dies ermöglicht den Anschluss von netzwerkversorgten und von Komponenten mit eigener Stromversorgung. Auch lassen sich auf diese Weise redundante Stromversorgungen in das Netzwerk einkoppeln, die bei Bedarf die Stromversorgung sicherstellen.

**Buszugriffs-  
verfahren**

DeviceNet arbeitet nach dem Verfahren Carrier-Sense Multiple Access (CSMA), d.h. jeder Teilnehmer ist bezüglich des Buszugriffs gleichberechtigt und kann auf den Bus zugreifen, sobald dieser frei ist (zufälliger Buszugriff).

Der Nachrichtenaustausch ist nachrichtenbezogen und nicht teilnehmerbezogen. Jede Nachricht ist mit einem priorisierenden Identifier eindeutig gekennzeichnet. Es kann immer nur ein Teilnehmer für seine Nachricht den Bus belegen.

Die Buszugriffssteuerung bei DeviceNet geschieht mit Hilfe der zerstörungsfreien, bitweisen Arbitrierung. Hierbei bedeutet zerstörungsfrei, dass der Gewinner der Arbitrierung sein Telegramm nicht erneut senden muss. Beim gleichzeitigen Mehrfachzugriff von Teilnehmern auf den Bus wird automatisch der wichtigste Teilnehmer ausgewählt. Erkennt ein sendebereiter Teilnehmer, dass der Bus belegt ist, so wird sein Sendewunsch bis zum Ende der aktuellen Übertragung verzögert.

**Adressierung**

Alle Teilnehmer am Bus müssen eindeutig über eine ID-Adresse identifizierbar sein. Jedes DeviceNet-Gerät besitzt eine Möglichkeit zur Adresseinstellung.

**EDS-Datei**

Zur Konfiguration einer Slave-Anschaltung in Ihrem eigenen Projektierwerkzeug bekommen Sie die Leistungsmerkmale der DeviceNet-Geräte in Form einer EDS-Datei (**E**lectronic **D**ata **S**heet) mitgeliefert.



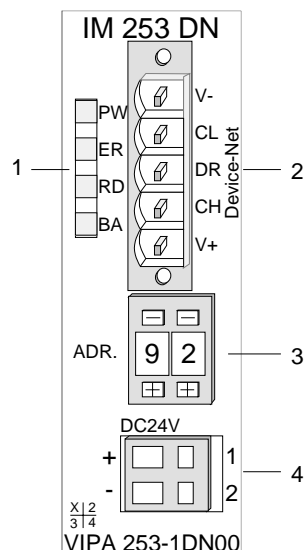
## IM 253DN - DeviceNet-Koppler - Aufbau

Der DeviceNet-Koppler IM 253DN ermöglicht die einfache Anbindung von dezentralen Peripheriemodulen über das DeviceNet-Protokoll.

### Eigenschaften

- Group 2 only Device  
- benutzt Predefined Connection Set
- Poll only Device  
- keine Betriebsart BIT STROBE  
- keine Betriebsart CHANGE OF STATE
- Unterstützung aller Baudraten: 125, 250 und 500kBaud
- Adresseinstellung über Schalter
- Einstellung der Übertragungsrate durch speziellen POWER ON Vorgang (Start mit Adresse 90...92)
- LED Statusanzeigen
- max. 32 Peripheriebaugruppen steckbar
- davon maximal 8 parametrierbare Module
- Modulkonfiguration mit DeviceNet Manager
- Profibus-DeviceNet-Übergang durch Kombination mit IM 208DP möglich

### Frontansicht 253-1DN00



- [1] LED Statusanzeige
- [2] Anschluss DeviceNet
- [3] Adresseinsteller
- [4] Anschluss 24V Versorgungsspannung

## Komponenten

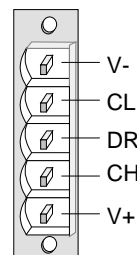
### LEDs

Zur schnellen Diagnose des aktuellen Modul-Status befinden sich auf der Frontseite 4 LEDs. Eine detaillierte Beschreibung der Fehlerdiagnose über LED und Rückwandbus finden Sie im Unterkapitel "Diagnose".

Bez.	Farbe	Bedeutung
PW	gelb	Power-LED: Betriebsspannung ein
ER	rot	Fehler im DeviceNet oder am Rückwandbus
RD	grün	Status Rückwandbus
BA	gelb	Status DeviceNet

### Anschluss DeviceNet

Der Anschluss an DeviceNet erfolgt über eine 5polige Buchse vom Typ Open Style Connector. Die Belegung der Kontakte ist auch auf der Front am Modulgehäuse aufgedruckt.

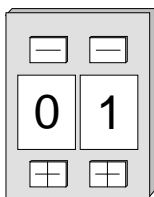


- [V-] GND Betriebsspannung
- [CL] CAN low
- [DR] DRAIN
- [CH] CAN HIGH
- [V+] DC 24V Betriebsspannung

### Adresseinsteller

Der Adresseinsteller dient:

- der Festlegung einer eindeutigen DeviceNet-Adresse
- der Programmierung der Übertragungsrate



#### Adressen:

0...63: DeviceNet Adresse

90, 91, 92: Übertragungsrate auf 125, 250, 500kbaud setzen

### Spannungsversorgung

Der Buskoppler-Koppler besitzt ein eingebautes Netzteil. Das Netzteil ist mit 24V Gleichspannung zu versorgen. Über die Versorgungsspannung werden neben der Buskopplerelektronik auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus mit maximal 3,5A versorgen kann.

Das Netzteil ist gegen Verpolung und Überstrom geschützt.

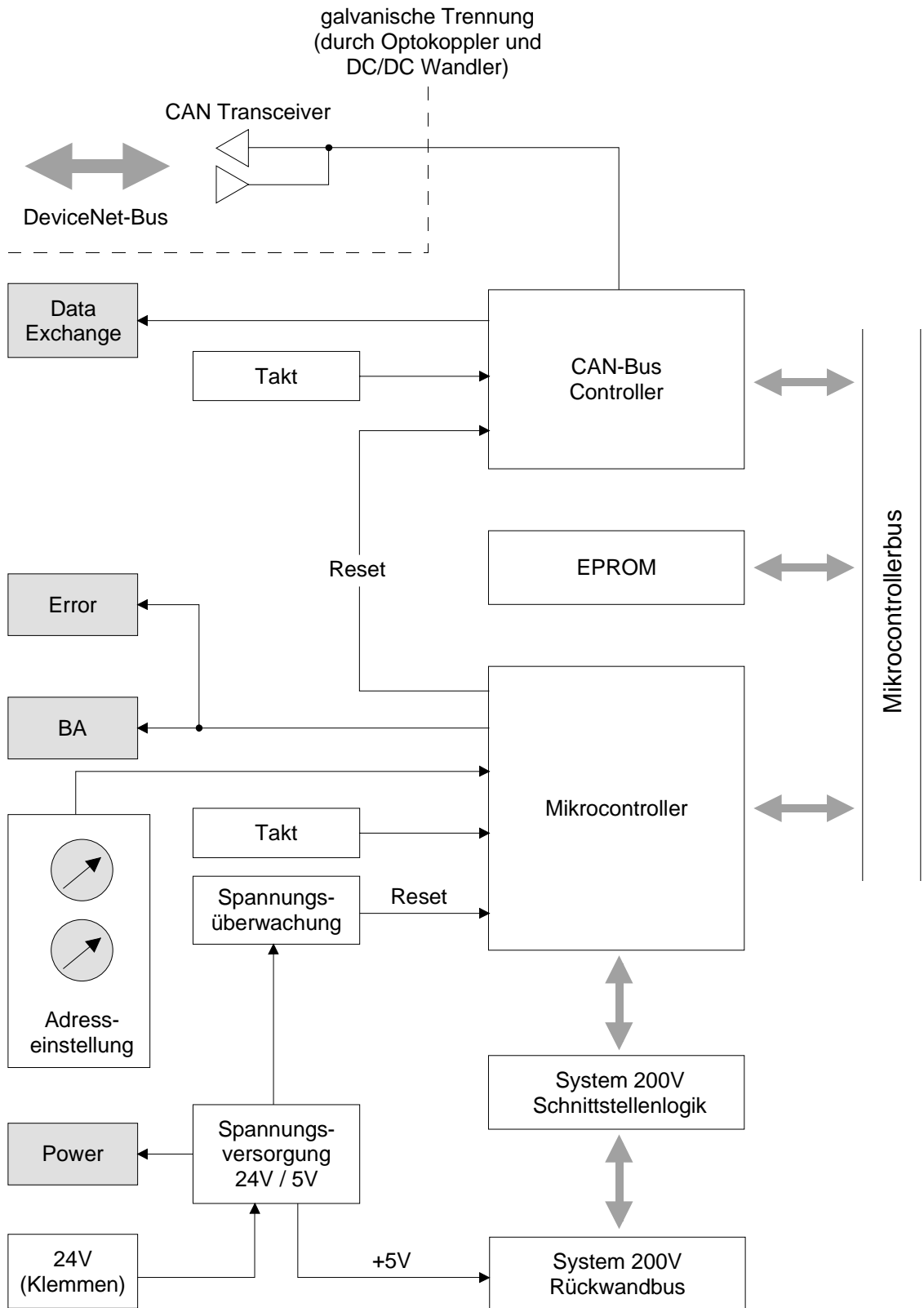
DeviceNet und Rückwandbus sind galvanisch voneinander getrennt.



#### Hinweis!

Der DeviceNet-Koppler bezieht keinen Strom aus der im DeviceNet mitgeführten Versorgungsspannung.

**Blockschaltbild** Das nachfolgende Blockschaltbild zeigt den prinzipiellen Hardwareaufbau des Buskopplers und die Kommunikation, die intern stattfindet:



## Projektierung unter Einsatz des DeviceNet-Managers

### Übersicht

Die eigentliche Projektierung eines DeviceNet erfolgt mit der Software *DeviceNet-Manager* der Firma Allen - Bradley.

Die Projektierung besteht aus folgenden Schritten:

- Konfiguration des *DeviceNet-Managers*
- Übertragungsrate und DeviceNet-Adresse am Modul einstellen
- Test am DeviceNet
- Module parametrieren
- I/O-Adressierung des DeviceNet-Scanners (Master)

---

### Konfiguration des DeviceNet-Managers

Durch die Konfiguration werden die modulspezifischen Daten des VIPA DeviceNet-Kopplers dem DeviceNet Manager verfügbar gemacht.

Folgende Schritte sind hierzu erforderlich:

- Legen Sie die mitgelieferte Diskette in Ihren PC ein.
- Kopieren Sie die Datei **IM253DN.BMP** auf Ihren PC in das Verzeichnis **/DNETMGR/RES** des *DeviceNet-Managers*
- Die EDS-Datei befindet sich auf der Diskette in einem Unterverzeichnis von 501.VND. Kopieren Sie die Datei **1.EDS** in das Verzeichnis **/DNETMGR/EDS/501.VND/0.TYP/1.COD**

Sie können aber auch die ganze Struktur

```
501.vnd
  |-- 0.typ
    |-- 1.cod
      |-- 1.eds
        |-- device.bmp
```

in das Verzeichnis DNETMGR/EDS kopieren.

## Einstellung von Baudrate und DeviceNet-Adresse

Sie haben die Möglichkeit bei ausgeschalteter Spannungsversorgung die Baudrate bzw. die DeviceNet-Adresse einzustellen und diese durch Einschalten der Spannungsversorgung an das Modul zu übergeben.

### Übertragungsrate einstellen

Alle am Bus angeschlossenen Teilnehmer kommunizieren mit der gleichen Übertragungsrate. Sie können über den Adresseinsteller eine gewünschte Übertragungsrate vorgeben.

- Schalten Sie die Spannungsversorgung aus.
- Stellen Sie die gewünschte Baudrate am Adresseinsteller ein.

Einstellung	Baudrate in kBaud
90	125
91	250
92	500

- Schalten Sie die Spannungsversorgung ein.

*Die eingestellte Übertragungsrate wird im EEPROM gespeichert.*

*Ihr DeviceNet-Koppler ist nun auf die gewünschte Baudrate eingestellt.*

### LED-Anzeige RD-LED ER-LED

Bei erfolgreicher Speicherung leuchtet die RD-LED (grün).

Bei falsch eingestellter Datenübertragungsrate leuchtet die ER-LED.

### DeviceNet-Adresse einstellen

Alle am Bus angeschlossenen Teilnehmer müssen eindeutig über eine DeviceNet-Adresse identifizierbar sein. Die Adresse können Sie im spannungslosen Zustand am Adresseinsteller einstellen.

- Schalten Sie die Spannungsversorgung aus.
- Stellen Sie am Adresseinsteller die gewünschte Adresse ein.  
**Bitte beachten Sie, dass die Adresse nur einmal im System vorhanden ist und zwischen 0 und 63 liegt!**

- Schalten Sie die Spannungsversorgung ein.

*Die eingestellte Adresse wird übernommen und im RAM abgelegt.*



#### Hinweis!

Änderungen in der Adressierung werden erst nach POWER ON oder einem automatischen Reset wirksam. Änderungen im normalen Betrieb werden nicht erkannt.

### LED-Anzeige ER-LED

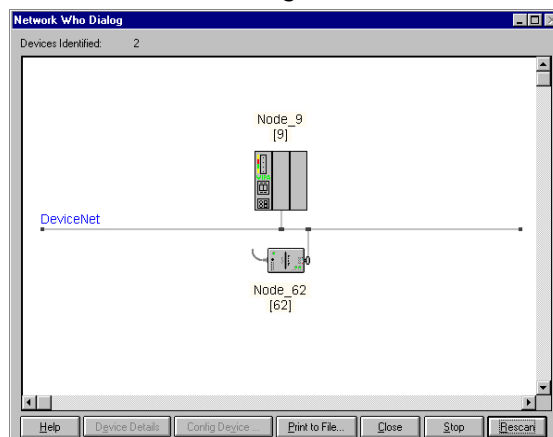
Bei einer falschen oder bereits vorhandenen Adresse leuchtet nach Power On die ER-LED (rot).

## Test am DeviceNet-Bus

### Vorgehen

- PC mit *DeviceNet-Manager* und VIPA DeviceNet-Koppler an das DeviceNet anschließen.
- Übertragungsrate und DeviceNet-Adresse am Koppler einstellen
- Spannungsversorgung des Buskopplers einschalten.
- *DeviceNet-Manager* starten.
- Im Manager die gleiche Datenrate einstellen wie beim Buskoppler
- Im Manager die Funktion NETWORK WHO starten

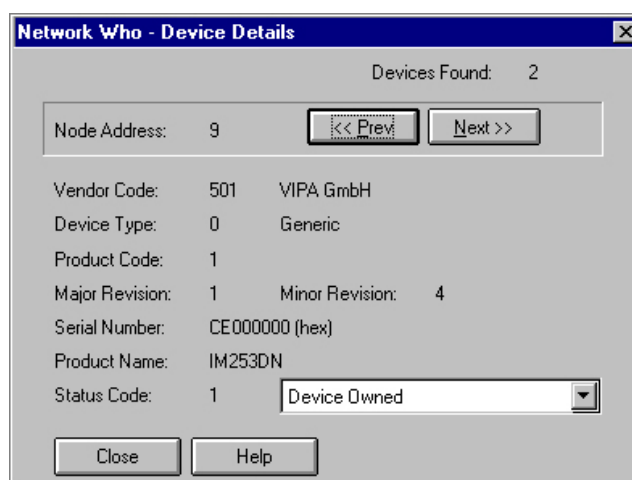
*Es öffnet sich das folgende Netzwerkfenster:*



### Device Details

- Buskoppler mit der rechten Maustaste anklicken.
- Im Kontextmenü die Funktion DEVICE DETAILS wählen.

*Am Bildschirm öffnet sich das Fenster DEVICE DETAILS*



*In diesem Fenster können Sie von jedem am DeviceNet befindlichen Koppler die DeviceNet-Adresse (Node Address), den Hersteller-Code (Vendor Code) hier 501 für VIPA GmbH und weitere interne Informationen abrufen.*

## Module im DeviceNet-Manager parametrieren

Das System 200V umfasst auch parametrierbare Module wie z.B. die Analogmodule. Werden solche Module am DeviceNet-Koppler betrieben, müssen die Parameterdaten im DeviceNet-Koppler gespeichert werden.

### Parametrierung in Gruppen

Folgendes sollten Sie bei der Parametrierung beachten:

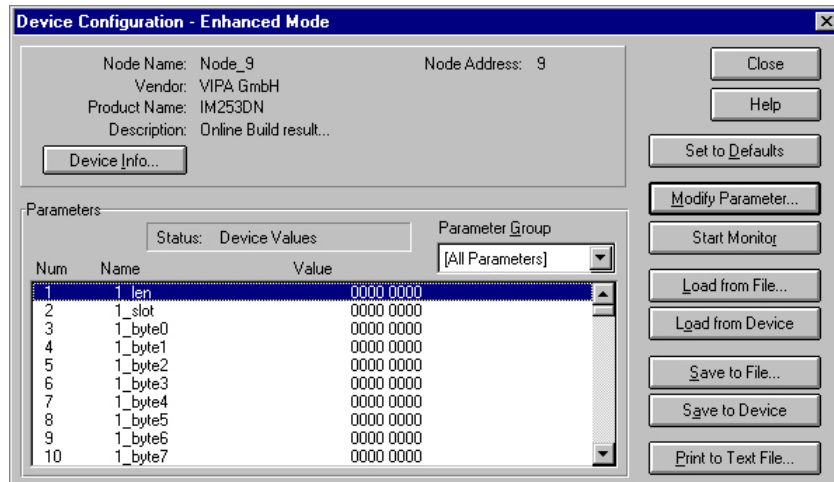
- In DeviceNet werden Parameterdaten in Form von Gruppen verwaltet.
- Maximal kann jeder DeviceNet-Koppler 144Byte Parameterdaten verarbeiten und speichern.
- Die 144Byte sind aufgeteilt in 8 Gruppen zu je 18Byte.
- Jede Gruppe darf die Parameterdaten für 1 Modul beinhalten.
- Die Gruppen sind durch eine Prefix-Nr. (1...8) im Parameter-Namen gekennzeichnet.
- Die Angabe über die Anzahl der Parameterbytes erfolgt im "Len"-Parameter (1. Parameter) einer Gruppe. Die Anzahl der Parameterbytes finden Sie in der Dokumentation zu den Peripheriemodulen in den Technischen Daten.
- Die Gruppen-Zuordnung zu einem Modul ist unabhängig von Steckplatz und gesteckter Reihenfolge.
- Die Steckplatzzuordnung erfolgt durch den "Slot"-Parameter einer Gruppe (2. Parameter)
- Durch Doppelklick auf einen Parameter können Werte als Bit-Muster eingegeben werden
- Freie Gruppen erkennen Sie am "Value" 0000 0000.

### Vorgehen

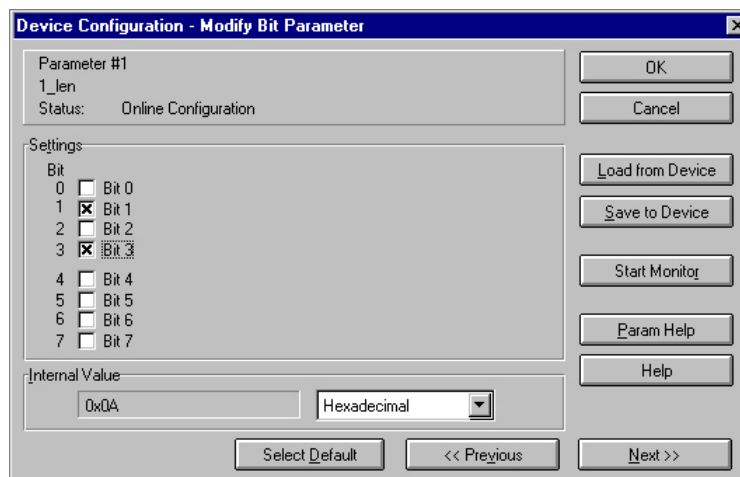
Voraussetzung: Ihr IM 253DeviceNet-Koppler befindet sich aktiv am Bus.

Nachfolgend ist beschrieben, wie im *DeviceNet-Manager* die Parametersätze angelegt werden.

- Führen Sie im *DeviceNet-Manager* die Funktion WHO aus.  
*Es öffnet sich ein Netzwerkfenster, das unter anderem auch Ihren Koppler zeigt.*
- Doppelklicken Sie auf das Symbol des Buskopplers, dessen Parameterdaten Sie ändern möchten.  
*Die Parameterdaten werden aus dem Koppler geladen und in folgendem Fenster dargestellt:*



- Suchen Sie in der Parameterliste eine freie Gruppe (Value=0000 0000). Durch Einstellung von "All Parameters" im Auswahlfeld *Parameter Group* können Sie alle 8 Gruppen in der Parameterliste ausgeben.
- Doppelklicken Sie auf den "Len"-Parameter.  
*Es öffnet sich das folgende Dialogfenster:*

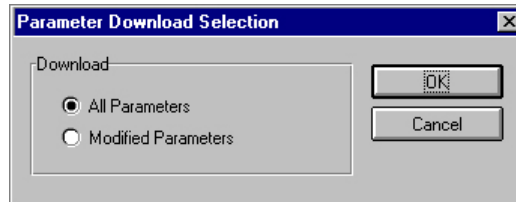


- Tragen Sie hier bit-codiert die Anzahl der Parameterbytes ein, die das zu parametrierende Modul besitzt. Die Anzahl entnehmen Sie bitte der Dokumentation des Peripheriemoduls. Dabei entsprechende Bits durch Anklicken setzen (Checkbox markiert) oder zurücksetzen.
- Zum Schließen der Maske klicken Sie auf [OK]. Über die Schaltfläche [Next>>] wird der nächste Parameter (Slot) der gleichen Gruppe angezeigt.
- Geben Sie nun bit-codiert auf die gleiche Weise die Steckplatz-Nr. des zu parametrierenden Moduls an.  
Über die Schaltfläche [Param Help] können Sie den Eingabebereich abrufen.
- Über [Next >>] können Sie jetzt nacheinander die Parameterbytes Ihres Moduls eingeben.



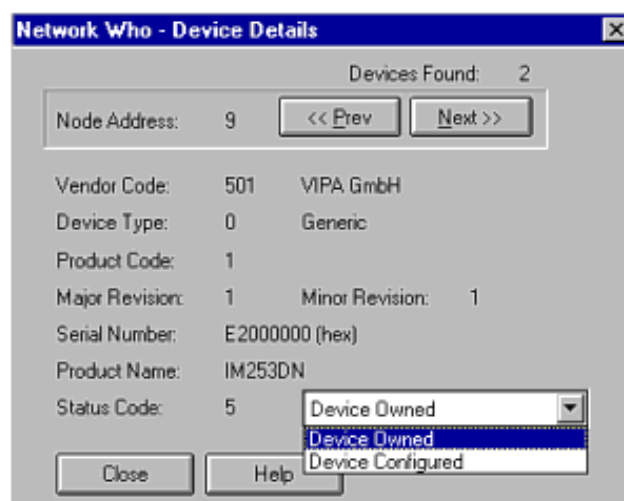
- Zur Parametrierung weiterer Module wählen Sie eine andere freie Gruppe und verfahren Sie auf die gleiche Weise.
- Sind alle Parameter in den einzelnen Gruppen abgelegt, können Sie über die Schaltfläche [Save to Device] die Parameter an den DeviceNet-Koppler übertragen und dort speichern.

Mit dem Klick auf [Save to Device] öffnet sich folgendes Auswahlfenster:



Hier können Sie wählen ob alle Parameter oder nur die geänderten Parameter übertragen werden sollen.

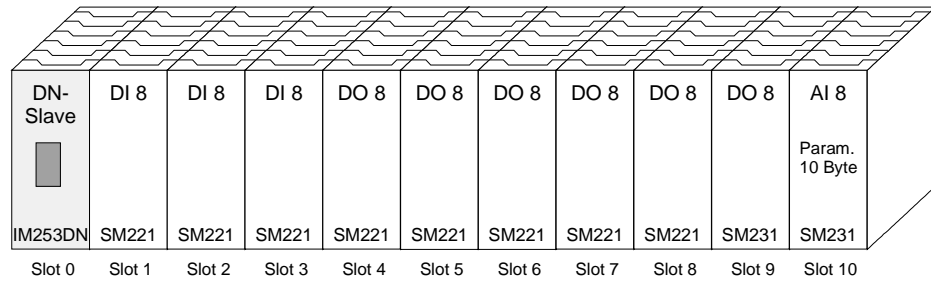
- Während der Datenübertragung erhalten Sie als Status-Text die Meldung "Status: downloading". Sobald die Übertragung beendet ist, wechselt der Status-Text in "Status: Device Values"
- Bei Abfrage der "Device Details" sieht man nun, dass der Status zusätzlich das Bit CONFIGURED enthält.



Nach Eingabe der Parameterwerte und anschließendem Download in den DeviceNet-Koppler sind die über den Rückwandbus angebotenen Peripheriemodule entsprechend parametrierbar.

**Beispiel**

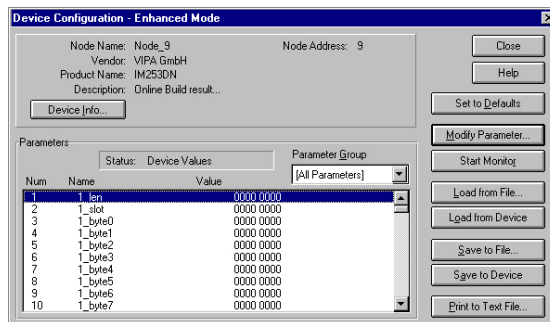
Nachfolgend soll kurz anhand eines Beispiels die Parametrierung am System 200V gezeigt werden. Das System hat folgenden Aufbau:



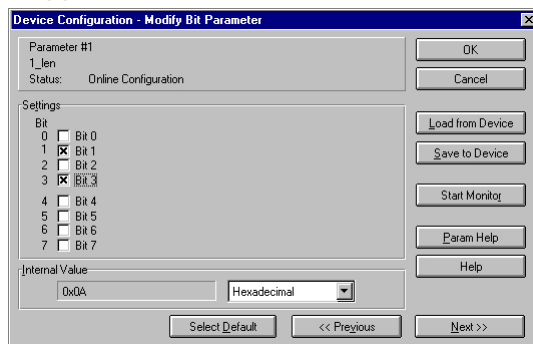
Das Beispiel zeigt einen DeviceNet-Koppler mit 10 Modulen, wobei die Module auf den Steckplätzen 1 bis 9 nicht parametrierbar sind. Nachfolgend ist die Parametrierung des Analog-Moduls auf Steckplatz 10 beschrieben:

- Voraussetzung:
- Das Beispiel ist aufgebaut und aktiv am Bus.
  - *DeviceNet-Manager* von Allen - Bradley ist installiert.

- Führen Sie im *DeviceNet-Manager* die Funktion WHO aus und öffnen Sie durch Doppelklick auf den DeviceNet-Koppler das Parameterfenster.



- Suchen Sie in der Parameterliste eine freie Gruppe (Value=0000 0000)
- Doppelklicken Sie auf den "Len"-Parameter.



Das Analog-Modul besitzt 10 Byte Parametrierdaten. Geben Sie diesen Wert bit-codiert ein.

- Klicken Sie auf [Next>>] und geben Sie als "Slot" den Steckplatz 10 an.
- Über [Next >>] können Sie jetzt nacheinander die Parameterbytes Ihres Moduls eingeben.

Das Analog-Eingabe-Modul besitzt folgende Parameter:

Byte	Bit 7 ... Bit 0	Default
0	Diagnosealarm-Byte: Bit 0 ... 5: reserviert Bit 6: 0: Diagnosealarm gesperrt 1: Diagnosealarm freigegeben Bit 7: reserviert	00h
1	reserviert	00h
2	Funktions-Nr. Kanal 0 (siehe Modulbeschreibung)	2Dh
3	Funktions-Nr. Kanal 1 (siehe Modulbeschreibung)	2Dh
4	Funktions-Nr. Kanal 2 (siehe Modulbeschreibung)	2Dh
5	Funktions-Nr. Kanal 3 (siehe Modulbeschreibung)	2Dh
6	Option-Byte Kanal 0	00h
7	Option-Byte Kanal 1	00h
8	Option-Byte Kanal 2	00h
9	Option-Byte Kanal 3	00h

- Sind alle Parameter in der Gruppe abgelegt, können Sie über die Schaltfläche [Save to Device] die Parameter an den DeviceNet-Koppler übertragen und dort speichern.
- Während der Datenübertragung erhalten Sie als "Status"-Text die Meldung "Status: downloading". Sobald die Übertragung beendet ist, wechselt der "Status"-Text in "Status: Device Values"



### Hinweis!

Nachträgliche Änderungen an der Parametrierung sind jederzeit möglich. Klicken Sie hierzu auf [Load from Device], führen Sie Ihre Änderungen durch und speichern Sie mit [Save to Device] ihre Änderungen.

## I/O-Adressierung des DeviceNet-Scanners

Der DeviceNet-Koppler ermittelt automatisch die am Rückwandbus gesteckten Module und generiert hieraus die Anzahl der Ein- und Ausgangsbytes.

Bei der Projektierung der Ein-/Ausgabe-Module müssen Sie diese zwei Werte ermitteln und im DeviceNet-Scanner (Master) angeben:

- produced connection size (Anzahl Eingangsbyte)
- consumed connection size (Anzahl Ausgangsbytes)

Die Adressierung ergibt sich aus der Reihenfolge der Module (Steckplatz 1 bis 32) und der im DeviceNet-Scanner für den Buskoppler eingestellten Basisadresse.

### DeviceNet-Scanner konfigurieren

- Im DeviceNet-Scanner die Verbindungsart POLL IO einstellen.
- Parameter einstellen:  
 "Receive data size" = Anzahl Eingangsbyte  
 "Transmit data size" = Anzahl Ausgangsbyte
- Basisadresse (Mapping) von Receive Data und Transmit Data entsprechend den individuellen Gegebenheiten einstellen.
- DeviceNet-Koppler IM 253DN in der Scanliste aktivieren.
- DeviceNet-Scanner starten.

Nach der Konfiguration des DeviceNet-Scanners können die Ein- und Ausgabe-Module unter den parametrierten Adressen angesprochen werden.

### Beispiel

Am Rückwandbus sind die folgenden 6 Module gesteckt:

Steckplatz	Gestecktes Modul	Eingabe-Daten	Ausgabe-Daten
Slot 0	DeviceNet-Koppler	-	-
Slot 1	Digital Out SM 222		1Byte
Slot 2	Digital Out SM 222		1Byte
Slot 3	Digital In SM 221	1Byte	
Slot 4	Analog In SM 231	4Words	
Slot 5	Analog Out SM 232		4Words
Summe:		1+4*2=9Byte	1+1+4*2=10Byte

Daraus ergeben sich:

- produced connection size: 9Byte (Summe Eingabe-Bytes)
- consumed connection size: 10Byte (Summe Ausgabe-Bytes)

## Diagnose

### Überblick

Die eingebauten LEDs zur Statusanzeige erlauben eine umfassende Diagnose sowohl beim POWER ON - Vorgang, als auch während des Betriebs. Entscheidend für die Diagnose ist die Kombination der verschiedenen LEDs und der aktuelle Betriebsmodus.

Es bedeuten:

LED	Bedeutung
<input type="checkbox"/> aus	LED leuchtet nicht
<input type="checkbox"/> ein	LED leuchtet dauernd
<input checked="" type="checkbox"/> blinkt	LED blinkt

Entsprechend der Stellung des Adresseinstellers werden folgende Betriebsmodi unterschieden:

- DeviceNet-Modus (Adresseinsteller in Stellung 0...63)
- Parametrier-Modus (Adresseinsteller in Stellung 90...92)

---

### DeviceNet-Modus

#### POWER ON ohne DeviceNet

LED	Bedeutung
<input checked="" type="checkbox"/> PW ein <input type="checkbox"/> ER aus <input checked="" type="checkbox"/> RD blinkt <input type="checkbox"/> BA aus	Nach POWER ON leuchtet die PW-LED und zeigt eine korrekte Spannungsversorgung an. Die RD-LED blinkt, weil die im EEPROM gespeicherten Konfigurationsdaten erfolgreich in die Peripheriemodule geladen wurden
<input checked="" type="checkbox"/> PW ein <input checked="" type="checkbox"/> ER ein <input type="checkbox"/> RD aus <input type="checkbox"/> BA aus	Nach POWER ON leuchtet die PW-LED. Die ER-LED leuchtet, weil der Rückwandbus gestört ist oder die Konfigurationsdaten nicht in die Peripheriemodule geladen werden konnten.

**POWER ON mit DeviceNet ohne Master**

LED	Bedeutung
<input checked="" type="checkbox"/> PW ein <input type="checkbox"/> ER aus <input checked="" type="checkbox"/> RD blinkt <input checked="" type="checkbox"/> BA blinkt	Nach POWER ON leuchtet die PW-LED. Die RD-LED blinkt, weil: <ul style="list-style-type: none"> <li>• der Rückwandbus in Ordnung ist</li> <li>• die im EEPROM gespeicherten Konfigurationsdaten erfolgreich in die parametrierbaren Peripheriemodule geladen wurden.</li> </ul> Die BA-LED blinkt, weil: <ul style="list-style-type: none"> <li>• wenigstens ein weiteres Gerät aktiv am DeviceNet ist,</li> <li>• und die am Koppler eingestellte Adresse eindeutig ist.</li> </ul>
<input checked="" type="checkbox"/> PW ein <input checked="" type="checkbox"/> ER ein <input type="checkbox"/> RD aus <input type="checkbox"/> BA aus	Nach POWER ON leuchtet die PW-LED. Die ER-LED leuchtet, weil die am DeviceNet-Koppler eingestellte <ul style="list-style-type: none"> <li>• Adresse ungültig oder bereits von einem anderen Gerät belegt ist</li> <li>• Datenübertragungsrate falsch ist.</li> </ul>
<input checked="" type="checkbox"/> PW ein <input checked="" type="checkbox"/> ER ein <input checked="" type="checkbox"/> RD blinkt <input checked="" type="checkbox"/> BA blinkt	Nach POWER ON leuchtet die PW-LED. Die ER-LED leuchtet, wenn die Konfigurationsdaten nicht in die parametrierbaren Peripheriemodule geladen wurden. Die RD-LED blinkt, da <ul style="list-style-type: none"> <li>• der Rückwandbus in Ordnung ist</li> <li>• die Konfigurationsdaten nicht in die parametrierbaren Peripheriemodule geladen wurden.</li> </ul> Die BA-LED blinkt, da <ul style="list-style-type: none"> <li>• wenigstens ein weiteres Gerät aktiv am DeviceNet ist,</li> <li>• und die am Koppler eingestellte Adresse eindeutig ist.</li> </ul>

**POWER ON mit DeviceNet und Master**

LED	Bedeutung
<input type="checkbox"/> PW ein <input type="checkbox"/> ER ein <input checked="" type="checkbox"/> RD blinkt <input type="checkbox"/> BA ein	<p>Nach POWER ON leuchtet die PW-LED.</p> <p>Die ER-LED leuchtet, da die Konfigurationsdaten nicht in die parametrierbaren Peripheriemodule geladen wurden.</p> <p>Die RD-LED blinkt, da</p> <ul style="list-style-type: none"> <li>• der Rückwandbus in Ordnung ist</li> <li>• die Konfigurationsdaten nicht in die parametrierbaren Peripheriemodule geladen wurden.</li> </ul> <p>Die BA-LED leuchtet, da</p> <ul style="list-style-type: none"> <li>• der Koppler IM 253DN eine DeviceNet-Verbindung zu einem Master aufgebaut hat.</li> </ul> <p>Hinweise!</p> <p>Der Koppler IM 253DN führt nach 30s einen Reset durch.</p> <p>Ein Fehler bei POWER ON mit DeviceNet und Master hat die gleiche LED-Anzeige wie ein Hardware-Fehler.</p> <p>Die Unterscheidung ist möglich:</p> <ul style="list-style-type: none"> <li>• durch Trennen der DeviceNet-Verbindung → ER-LED und RD blinken!</li> <li>• mit Network Who im DeviceNet-Manager → Bei Hardware-Fehler erscheint der IM253DN nicht im Netzwerk</li> </ul> <p>Bei einem Hardware-Fehler setzen Sie sich bitte mit der VIPA-Hotline in Verbindung!</p>

**Fehlerfreier Betriebszustand mit DeviceNet und Master**

LED	Bedeutung
<input type="checkbox"/> PW ein <input type="checkbox"/> ER aus <input checked="" type="checkbox"/> RD ein <input type="checkbox"/> BA ein	<p>Nach POWER ON leuchtet die PW-LED. Die RD-LED leuchtet, weil die Verbindung über den Rückwandbus zu den Peripheriemodulen möglich ist.</p> <p>Die BA-LED leuchtet, weil der Koppler IM 253DN eine DeviceNet-Verbindung zu einem Master aufgebaut hat.</p>

**Fehler im Betrieb mit DeviceNet und Master**

LED	Bedeutung
<input type="checkbox"/> PW ein <input checked="" type="checkbox"/> ER ein <input type="checkbox"/> RD aus <input checked="" type="checkbox"/> BA ein	Nach POWER ON leuchtet die PW-LED. Die ER-LED leuchtet, da am Rückwandbus ein Fehler erkannt wurde. Die BA-LED leuchtet, weil der Koppler IM 253DN eine DeviceNet-Verbindung zu einem Master aufgebaut hat. Hinweis! Der Koppler IM 253DN führt nach 30s einen Reset durch.

**Übergang vom Betriebs- in den Modulfehler-Status**

LED	Bedeutung
<input checked="" type="checkbox"/> PW ein <input checked="" type="checkbox"/> ER ein <input type="checkbox"/> RD aus <input type="checkbox"/> BA aus	Die ER-LED leuchtet 1 Sekunde lang, weil ein Modulfehler erkannt wurde. Anschließend führt der Koppler IM 253DN einen Reset durch. Nach dem Reset startet der Koppler neu und zeigt den Fehler durch entsprechende LED-Anzeige an.

**Anzeige bei Neustart nach Reset**

LED	Bedeutung
<input checked="" type="checkbox"/> PW ein <input checked="" type="checkbox"/> ER ein <input checked="" type="checkbox"/> RD blinkt <input checked="" type="checkbox"/> BA ein	Die ER-LED leuchtet dauernd und die RD-LED blinkt, weil die Anzahl der I/O-Daten durch den Modulausfall verändert ist. Die Konfigurationsdaten konnten nicht übertragen werden. An allen Allen - Bradley Scannern erscheint die Meldung #77.
<input checked="" type="checkbox"/> PW ein <input type="checkbox"/> ER aus <input checked="" type="checkbox"/> RD ein <input checked="" type="checkbox"/> BA ein	Die ER-LED leuchtet nicht und die RD-LED leuchtet dauernd, weil die Anzahl der I/O-Daten durch den Modulausfall verändert ist. Die Verbindung zu den I/O-Modulen wurde aufgebaut. An allen Allen - Bradley Scannern erscheint die Meldung #77.



**Übergang vom Betriebs- in den Verbindungsfehler-Status**

LED	Bedeutung
<input type="checkbox"/> PW ein	Die ER-LED blinkt, weil die Zeitüberwachung der I/O-Verbindung einen Fehler erkannt hat. Die RD-LED blinkt, weil die I/O-Verbindung nicht mehr existiert. Alle Ein- und Ausgänge werden auf Null gesetzt. Die BA-LED leuchtet, weil die Verbindung zum Master noch besteht.
<input checked="" type="checkbox"/> ER blinkt	
<input checked="" type="checkbox"/> RD blinkt	
<input type="checkbox"/> BA ein	

**Parametrier-Modus****POWER ON im Parametrier-Modus**

LED	Bedeutung
<input type="checkbox"/> PW ein	Nach POWER ON leuchtet die PW-LED und zeigt eine korrekte Spannungsversorgung an. Die RD-LED leuchtet nach kurzer Zeit auf, weil die Baudrate in das EEPROM übernommen wurde.
<input type="checkbox"/> ER aus	
<input checked="" type="checkbox"/> RD ein	
<input type="checkbox"/> BA aus	

**Geräte-Fehler**

LED	Bedeutung
<input type="checkbox"/> PW ein	Am Koppler ist eine ungültige Adresse eingestellt. Gültige Einstellung wählen: <ul style="list-style-type: none"> <li>• 0...63 als DeviceNet-Adresse</li> <li>• 90...92 für die Einstellung der Baudrate</li> </ul>
<input checked="" type="checkbox"/> ER ein	
<input type="checkbox"/> RD aus	
<input type="checkbox"/> BA aus	
<input type="checkbox"/> PW ein	Wenn der Koppler nicht mit dem DeviceNet verbunden ist, wurde ein Fehler im internen EEPROM oder RAM erkannt. Bei einer Verbindung mit dem DeviceNet kann auch ein Fehler beim Übertragen der Konfigurationsdaten in die Peripheriemodule vorliegen.  Hinweis! Ein Fehler bei POWER ON mit DeviceNet und Master hat die gleiche LED-Anzeige wie ein Hardware-Fehler. Die Unterscheidung ist möglich: <ul style="list-style-type: none"> <li>• durch Trennen der DeviceNet-Verbindung → ER-LED und RD blinken!</li> <li>• mit Network WHO im DeviceNet-Manager → Bei Hardware-Fehler erscheint der IM 253DN nicht im Netzwerk!</li> </ul> Bei einem Hardware-Fehler setzen Sie sich bitte mit der VIPA-Hotline in Verbindung!
<input checked="" type="checkbox"/> ER ein	
<input checked="" type="checkbox"/> RD ein	
<input type="checkbox"/> BA ein	

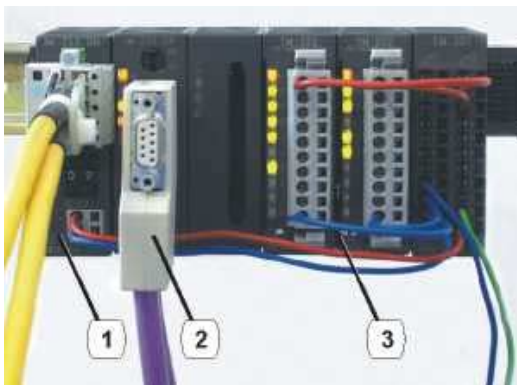
## Schnittstelle zum Profibus

### Beschreibung

Das modulare System 200V ermöglicht es Ihnen auf einfachste Art und Weise eine DeviceNet–Profibus-Bridge aufzubauen. Der Profibus-Master wird einfach direkt am Rückwandbus gemeinsam mit dem DeviceNet-Koppler betrieben.

Über die Verbindung von DeviceNet zu Profibus-DP können 256Byte Eingangs- und 256Byte Ausgangsdaten ausgetauscht werden.

Wird die maximale Datenanzahl nicht voll ausgenutzt, ist es außerdem möglich, zusätzlich zum Profibus-Master auch Peripheriemodule zu stecken.



- [1] DeviceNet-Koppler IM 253DN
- [2] Profibus-Master IM 208DP(0)
- [3] Zusätzliche Peripheriemodule

### Beispiel

Sie wollen eine Verbindung zwischen DeviceNet und Profibus-DP schaffen. Am Rückwandbus sind die folgenden 4 Module gesteckt:

Steckplatz (hex)	Gestecktes Modul	I/O-Daten und Adressen
Slot 0	DeviceNet-Koppler IM 253DN	-
Slot 1	Profibus-Master IM 208DP	Input ab Adresse 0 Output ab Adresse 2
Slot 2	Digital Out SM 222	1Byte, Adresse 0
Slot 3	Digital Out SM 222	1Byte, Adresse 1

### Vorgehen

- Bauen Sie Ihr System auf, indem Sie den Profibus-Master IM 208DP rechts neben DeviceNet-Koppler stecken, gefolgt von den 2 Ausgabemodulen (siehe Abbildung).
- Bitte achten Sie darauf, die Adressen der direkt gesteckten Peripheriemodule in Ihrem Profibus-Projektierwerkzeug zu reservieren. Näheres hierzu finden Sie in der Dokumentation zu Ihrem Profibus-Master.

Die über Profibus-DP angebotenen Peripheriebaugruppen und die Ausgabebaugruppen tauschen ihre Daten mit dem Profibus-Master aus. Dieser wiederum kommuniziert über den Rückwandbus mit dem DeviceNet-Koppler.

## Technische Daten

### DeviceNet-Koppler IM 253DN

Elektrische Daten	VIPA 253-1DN00
Spannungsversorgung Stromaufnahme	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil Buskoppler: 50mA inkl. Versorgung der Peripheriemodule: max. 800mA
Ausgangsstrom Rückwandbus Potenzialtrennung zwischen DeviceNet und Rückwandbus	max. 3,5A 500V eff.
Funktionsspezifische Daten	
Statusanzeige Physikalischer Anschluss DeviceNet Netzwerk-Topologie Übertragungsmedium Übertragungsrate Gesamtlänge des Buses Anzahl der Teilnehmer	über LED auf der Frontseite 5poliger Stecker Open Style Connector Linearer Bus, Stichleitungen bis 6m Länge möglich Abgeschirmtes, 5adriges Kabel 125, 250, 500kBaud bis 500m max. 64
Kombination mit Peripheriemodulen	
Modulanzahl Eingänge Ausgänge	max. 32 max. 256Byte max. 256Byte
Mechanische Daten	
Abmessungen (BxHxT) Gewicht	25,4x76x76mm 80g



## Teil 6 SERCOS

### Überblick

Inhalt dieses Kapitels ist die Beschreibung des SERCOS-Kopplers von VIPA. Nach einer Systemvorstellung folgt die Beschreibung des Moduls. Ein weiterer Bestandteil dieses Kapitels ist die Projektierung. Hier wird anhand von Beispielen die Projektierung des SERCOS-Kopplers und die Parametrierung der System 200V-Module beschrieben.

Mit einer Übersicht der Diagnosemeldungen und den Technischen Daten endet das Kapitel.

Nachfolgend sind beschrieben:

- SERCOS-Grundlagen
- Hardwarebeschreibung des SERCOS-Kopplers IM 253SC von VIPA
- Beschreibung der Identifier mit Zuordnungsbeispiel
- Beispiel zur Parametrierung
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 6 SERCOS</b> .....	<b>6-1</b>
Systemübersicht .....	6-2
Grundlagen.....	6-3
IM 253Sercos - SERCOS-Koppler - Aufbau.....	6-5
Grundparametrierung über Adresseinsteller .....	6-8
SERCOS Identifier .....	6-10
Beispiel zur automatischen ID-Zuweisung .....	6-13
Technische Daten.....	6-22



### Hinweis!

Für den Einsatz des in diesem Kapitel beschriebenen SERCOS-Kopplers werden fundierte Kenntnisse im Umgang mit SERCOS vorausgesetzt.

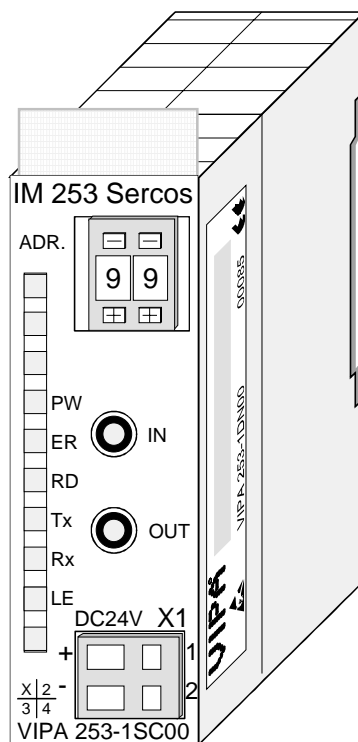
Sie finden hier ausschließlich die VIPA-spezifischen Eigenschaften erklärt.

Die Beschreibung der Eigenschaften, die dem SERCOS-Standard entsprechen, wie etwa die Identifier S-0 und S-1, finden Sie beispielsweise in der SERCOS-Spezifikation des SERCOS-Interface-Arbeitskreis.

## Systemübersicht

Mit dem SERCOS-Koppler von VIPA können Sie bis zu 32 Module Ihrer System 200V Peripherie an SERCOS anknüpfen.

Folgende SERCOS-Komponenten sind zur Zeit von VIPA verfügbar.



### Bestelldaten SERCOS

Typ	Bestellnummer	Beschreibung
IM 253SC	VIPA 253-1SC00	SERCOS-Koppler

## Grundlagen

### SERCOS

SERCOS steht für **S**erial **R**eal Time **C**ommunication **S**ystem und hat sich im Bereich der numerischen Steuerungen weltweit etabliert. Über die klassischen CNC-Maschinen hinaus hat sich diese Technik für schnelle und präzise Bewegungssteuerung in der gesamten Automatisierungstechnik bewährt.

SERCOS, auch "SERCOS-Interface" genannt, ist eine genormte digitale Antriebs-Schnittstelle auf Basis der Lichtwellenleiter-Technologie.

Die hohen Echtzeitanforderungen und die störsichere Lichtwellenleiter-Technologie sind wesentliche Merkmale dieses Bussystems.

Mit dem SERCOS-Koppler IM 253SC von VIPA ist nun auch eine SERCOS-Anbindung an die Sensor-/Aktor-Ebene möglich.

Der SERCOS-Koppler ist für den schnellen Datenaustausch auf der Sensor/Aktor Ebene konzipiert. Hier kommunizieren zentrale Steuergeräte wie z.B. SPS über eine schnelle, serielle Verbindung mit dezentralen Ein- und Ausgangsgeräten. Der Datenaustausch mit diesen dezentralen Geräten erfolgt zyklisch.

Der Master liest die Eingangsinformationen von den Slaves (Antriebstelegramm) und sendet die Ausgangsinformationen an die Slaves (Master-Daten-Telegramm).

Es können maximal 254 Slaves an einem Bus angeschlossen werden.

### Kommunikation

Bei SERCOS erfolgt die Kommunikation über drei Telegrammarten:

- *Master-Sync-Telegramm*  
Das Master-Sync-Telegramm wird von allen Antrieben gleichzeitig empfangen und dient der Synchronisation aller zeitbezogenen Aktionen in der Numerischen Steuerung (NC) und Antrieben.
- *Master-Daten-Telegramm*  
Das Master-Daten-Telegramm wird ebenso wie das Master-Sync-Telegramm von allen Antrieben gleichzeitig empfangen. Es beinhaltet die zyklischen Daten und die Servicedaten für alle Antriebe.
- *Konfigurierbares Datenfeld*  
Die Echtzeitdaten werden in jedem Kommunikationszyklus komplett im sogenannten konfigurierbaren Datenfeld übertragen. Die Antriebe senden ihre Telegramme aufeinanderfolgend in zugeteilten Zeitschlitzten.  
Mit Hilfe eines Ident-Nr.-Systems kann bei der Initialisierung festgelegt werden, welche Echtzeitdaten übertragen werden. Dies können neben numerischen Daten wie Soll- und Ist-Werten auch Bitlisten mit Ein-/Ausgabe-Anweisungen sein.

Der Austausch von Servicedaten erfolgt nur nach Aufforderung durch den Master. Servicedaten werden mit einer Handshake-Prozedur in 2, 4, 6 oder 8Byte-Portionen im Service-Datenfeld "Info" übertragen und beim Empfänger wieder zusammengesetzt.

**LWL als Übertragungsmedium**

SERCOS verwendet einen geschlossenen Lichtwellenleiter-Ring (LWL) als Übertragungsmedium. LWL hat eine hohe Immunität gegen elektromagnetische Störungen. Die Ringstruktur kommt mit der geringsten Anzahl LWL aus und erfordert keine aufwändigen T-Verzweigungen.

Die Länge jedes Übertragungsabschnitts kann mit Plastik-LWL bis 50m betragen, mit Glasfaser-LWL bis 250m. Die maximale Anzahl der Teilnehmer je Ring ist 254.

Die exakte Anzahl ist von folgenden Faktoren abhängig:

- erforderliche Kommunikations-Zykluszeit
- Betriebsdatenumfang
- Datenrate

**Buszugriffsverfahren**

Die Kommunikation erfolgt im Betrieb zyklisch als eine Master-Slave-Kommunikation. Die Zykluszeit wird bei der Initialisierung vorgegeben und kann zwischen 62µs und 65ms liegen.

Die Zykluszeiten sind so spezifiziert, dass die erforderliche Synchronisation mit fixen Arbeitszykluszeiten in Steuerung und Antrieben erzielt wird.

Kommunikations-Master in einem SERCOS-Ring ist immer die NC-Steuerung.

**Adressierung**

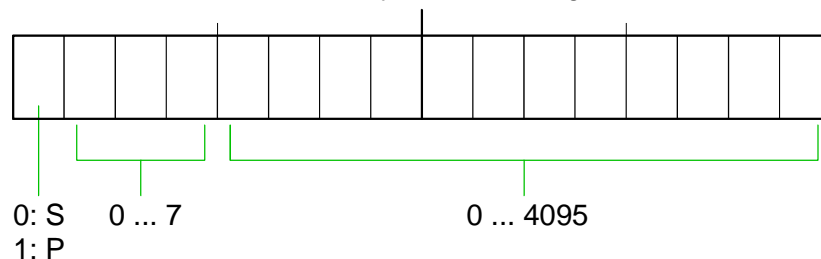
Alle Teilnehmer am Bus müssen eindeutig über ein Adresse identifizierbar sein. Jedes SERCOS-Gerät besitzt eine Möglichkeit zur Adresseinstellung.

**ID-Nummer für Datenaustausch**

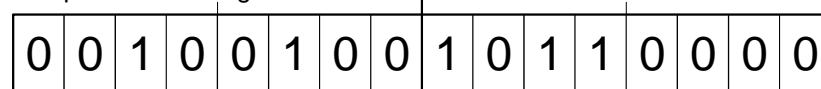
Die Adressierung der Daten beim bedarfsgesteuerten Datenaustausch und die Definition der Echtzeitdaten erfolgt bei SERCOS mittels Ident-Nummern. Für die ID-Nummern ist die ein Zahlenbereich von  $2^{16}$  festgelegt, der sich in zwei Bereiche aufteilt:

- 1 ... 32767: für Daten (S-0 ... S-7)
- 32768 ... 65535: für Parameter (P-0 ... P-7)

Ein Identifier besteht aus 2Byte und hat folgenden Aufbau:



Beispiel: Darstellung von S-2-1200





## IM 253Sercos - SERCOS-Koppler - Aufbau

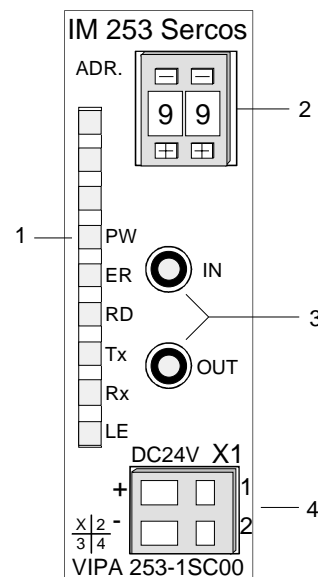
### Eigenschaften

Der SERCOS-Koppler IM 253SC ermöglicht die einfache Anbindung von dezentralen Peripheriemodulen aus dem System 200V an SERCOS.

Folgende Eigenschaften zeichnen den SERCOS-Koppler aus:

- Zum Anschluss von Lichtwellenleitern mit 1mm Fiberglas bzw. 200µm HCS®.
- Unterstützung aller SERCOS-Baudraten (2, 4, 8, 16Mbaud)
- Unterstützung aller System 200V-Module von VIPA
- max. 32 Peripheriemodule steckbar, die Anzahl der Analog-Module ist auf 16 Module begrenzt (beachten Sie hierzu auch die Angaben in den Aufbaurichtlinien)
- max. 256Byte Eingabe- und 256Byte Ausgabe-Daten
- Minimal möglicher SERCOS-Zyklus 1ms
- Adresseinsteller für Adressen (1 ... 89) und Parametrierung (90 ... 99)
- integriertes DC 24V-Netzteil zur Spannungs-Versorgung von Koppler Peripherie-Module.
- LED Statusanzeigen

### Frontansicht 253-1SC00



- [1] LED Statusanzeige
- [2] Adresseinsteller
- [3] LWL-Anschluss an SERCOS
- [4] Anschluss DC 24V Versorgungsspannung

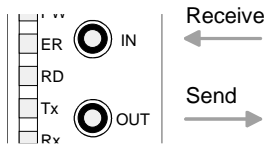
## Komponenten

### LEDs

Zur schnellen Diagnose des aktuellen Modul-Status befinden sich auf der Frontseite 6 LEDs.

Bez.	Farbe	Bedeutung
PW	gelb	Power-LED: Betriebsspannung ein
ER	rot	Fehler am Rückwandbus oder SERCOS
RD	grün	Blinkt wenn System OK ist und sich der Hochlauf unter Phase 4 befindet. Leuchtet sobald Hochlauf-Phase 4 erreicht ist,
Tx	gelb	leuchtet bei Sende-Aktivität über SERCOS
Rx	gelb	leuchtet bei Empfangs-Aktivität über SERCOS
LE	rot	Fehler in der LWL-Kommunikation (Leitungsunterbrechung bzw. Hardwaredefekt)

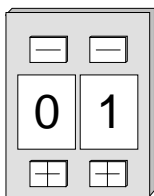
### LWL-Anschluss SERCOS



Über diese Buchse binden Sie den SERCOS-Koppler über Lichtwellenleiter in Ihren SERCOS-Ring ein.

Der Anschluss an SERCOS erfolgt über 2 LWL-Buchsen. Die Richtung der 2 Buchsen zeigt die Abbildung links. Die Buchsen sind zum Anschluss von Lichtwellenleitern mit 1mm Fiberglas bzw. 200µm HCS®.

### Adresseinsteller



Der Adresseinsteller dient:

- der Festlegung einer eindeutigen SERCOS-Adresse (1 ... 89)
- der Programmierung der Baudrate (90 ... 93)
- der Einstellung der Lichtintensität (94 ... 97)
- der Vorgabe des Modus für die Zeitschlitzberechnung (98, 99)

### Spannungsversorgung

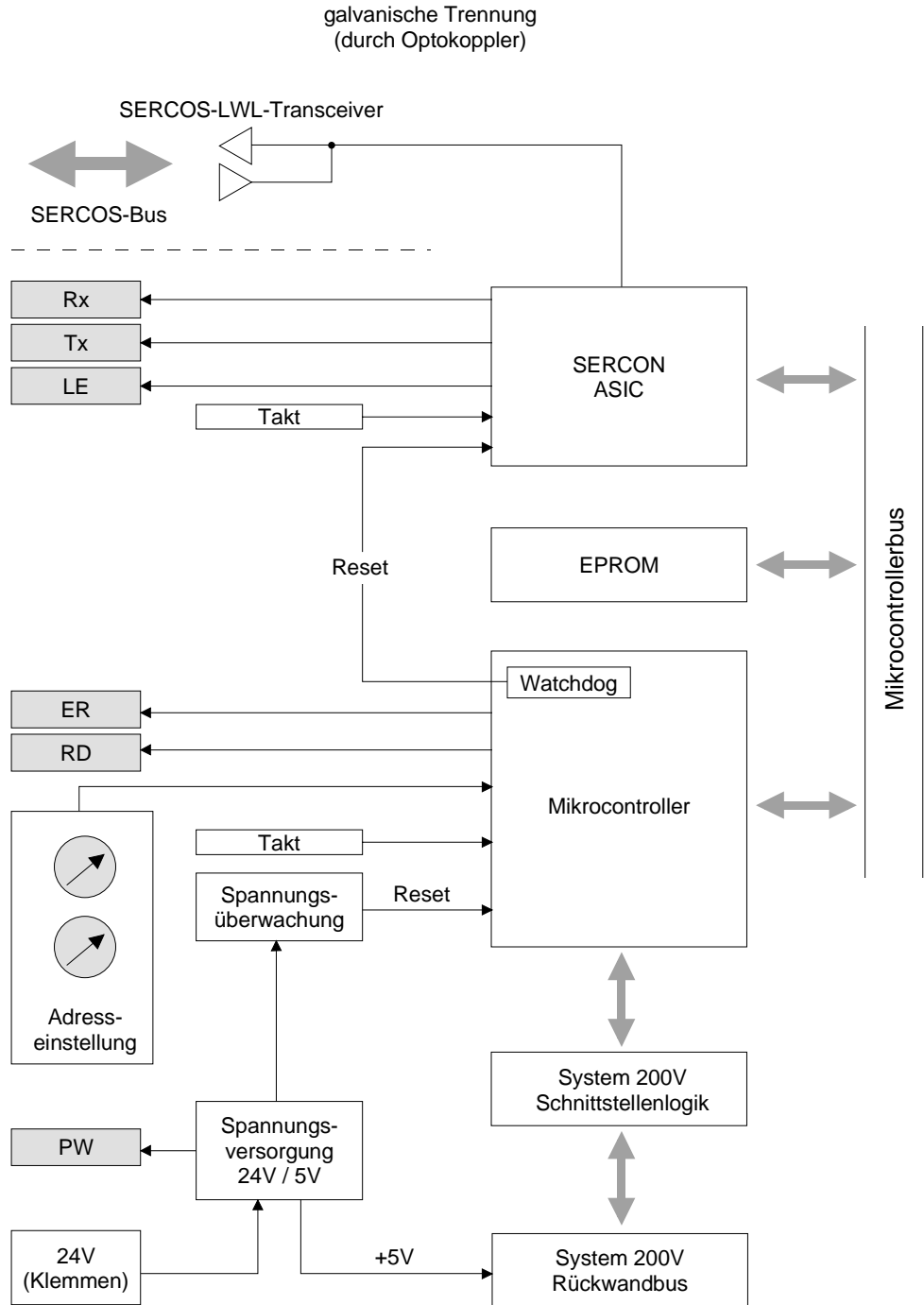
Der SERCOS-Koppler besitzt ein eingebautes Netzteil. Das Netzteil ist gegen Verpolung und Überstrom geschützt.

Hiermit werden neben der Modul-Elektronik auch die angeschlossenen Peripheriemodule über den Rückwandbus mit max. 3,5A versorgt.

Der Anschluss der Versorgungsspannung erfolgt über die Frontseite. Das Netzteil ist mit 24V DC (20,4 ... 28,8V) zu versorgen.

**Blockschaltbild**

Das nachfolgende Blockschaltbild zeigt den prinzipiellen Hardwareaufbau des SERCOS-Kopplers und die Kommunikation, die intern stattfindet:



## Grundparametrierung über Adresseinsteller

### Übersicht

Sie haben die Möglichkeit mittels des Adresseinstellers Grundeinstellungen des SERCOS-Kopplers zu ändern. Stellen Sie bei ausgeschaltetem SERCOS-Koppler den entsprechenden Adress-Code ein. Durch Einschalten der Spannungsversorgung wird dieser im SERCOS-Modul dauerhaft gespeichert.

Folgende Grundeinstellungen können auf diese Weise geändert werden:

- Übertragungsrate
- Lichtintensität
- Zeitschlitzberechnung



### Hinweis!

Bitte beachten Sie, dass Sie nur im spannungslosen Zustand den Adresseinsteller betätigen dürfen. Ansonsten kann dies zu Fehlfunktionen des SERCOS-Kopplers führen!

### Vorgehensweise

Schalten Sie die Versorgungsspannung des SERCOS-Kopplers aus.

Stellen Sie am Adresseinsteller den entsprechenden Adress-Code ein.

Schalten Sie die Spannungsversorgung ein.

→ Der eingestellte Parameter wird dauerhaft im SERCOS-Koppler gespeichert und dies über die grüne RD-LED angezeigt.

### Einstellbereiche

00: reserviert (darf nicht eingestellt werden)

01 ... 89: mögliche SERCOS-Stationsadressen

**90 ... 99: VIPA Sonderfunktionen zur Grundparametrierung**

### Übertragungsrate einstellen

Alle am Bus angeschlossenen Teilnehmer kommunizieren mit der gleichen Übertragungsrate. Sie können über den Adresseinsteller eine gewünschte Übertragungsrate vorgeben.

- Schalten Sie die Spannungsversorgung aus.
- Stellen Sie die gewünschte Baudrate am Adresseinsteller ein. Hierbei bedeuten:

90: 2Mbaud

91: 4Mbaud

92: 8Mbaud

93: 16Mbaud

- Schalten Sie die Spannungsversorgung ein.

→ Die eingestellte Übertragungsrate wird dauerhaft im SERCOS-Koppler gespeichert und dies über die grüne RD-LED angezeigt.

**Lichtintensität einstellen**

Sie haben die Möglichkeit die Lichtintensität der LWL-Diode in 4 Stufen vorzugeben.

- Schalten Sie die Spannungsversorgung aus.
- Stellen Sie die gewünschte Lichtintensität am Adresseinsteller ein. Sie haben folgende Einstellmöglichkeiten:

94: Lichtintensität 0 (Minimum)

95: Lichtintensität 1

96: Lichtintensität 2

97: Lichtintensität 3 (Maximum)

- Schalten Sie die Spannungsversorgung ein.

→ Die eingestellte Lichtintensität wird dauerhaft im SERCOS-Koppler gespeichert und dies über die grüne RD-LED angezeigt.

**Zeitschlitzberechnung**

Geben Sie hier den Betriebs-Modus für die Zeitschlitzberechnung vor. Folgende 2 Modi stehen zur Auswahl:

98: Mode\_All\_Cyclic

Die komplette Peripherie steht im zyklischen SERCOS-Betrieb zur Verfügung. Zusätzlich können Sie auch den Service Kanal verwenden. Abhängig von der Modulzahl sind SERCOS-Zyklen von 2ms oder größer erforderlich. Je mehr Peripherie gesteckt ist, desto höher ist die SERCOS-Zykluszeit zu wählen.

99: Mode\_All\_Service\_Channel

In diesem Modus steht keine Peripherie im zyklischen Betrieb zur Verfügung. Aufgrund dessen können Sie mit einer Zykluszeit von 1ms den SERCOS-Ring betreiben. Hierbei können Sie die Peripheriemodule ausschließlich über den Service-Kanal ansprechen.

## SERCOS Identifizier

### Übersicht

Der lesende und schreibende Zugriff auf das System 200V unter SERCOS erfolgt mittels Identnummern (kurz: IDN).

Hierbei gibt es für den SERCOS-Koppler IM 253SC folgende 3 Bereiche:

S-0-xxxx, S-1-xxxx: Standard IDNs, die vom SERCOS-Interface Arbeitskreis festgelegt werden

S-2-xxxx, S-3-xxxx: IDNs von VIPA zur Übertragung von Ein- und Ausgabe-Daten.

P-0-xxxx: IDNs von VIPA zur Übertragung von Parameterdaten

### Standard IDNs

#### S-0-xxxx, S-1-xxxx

Der SERCOS-Koppler IM 253SC unterstützt alle Standard IDNs. Näheres hierzu finden Sie in der SERCOS-Spezifikation des SERCOS-Arbeitskreis.

Abhängig vom Betriebsmodus werden die beiden Standard-ID-Listen gefüllt:

- Mode\_All\_Cyclic

S-0-0187: verweist auf alle Input-Identifizier S-2-xxxx

S-0-0188: verweist auf alle Output-Identifizier S-3-xxxx

- Mode\_All\_Service\_Channel

S-0-0187: Liste ist leer

S-0-0188: Liste ist leer

### VIPA-spezifische IDNs

#### S-2-xxxx, S-3-xxxx, P-0-xxxx

Da das System 200V ein modulares System ist, können bis zu 32 Module in beliebiger Reihenfolge und Mischung an den SERCOS-Koppler IM 253SC angebunden werden.

Somit entstehen dynamisch sehr unterschiedliche Konfigurationen von Ein- und Ausgabe-Kanälen. Ein Modul kann einen oder mehrere dieser Kanäle belegen. Die maximale Gesamtzahl von Ein-/Ausgabe-Kanälen (IO-Kanäle) ist auf 256 beschränkt. Das Mapping der Module und Ihrer IO-Kanäle in den S-2- bzw. S-3-Bereich und (bei parametrierbaren Modulen zusätzlich) in den P-Bereich geschieht automatisch.

**VIPA-spezifische  
Belegung der IDN  
S-2-xxxx, S-3-xxxx  
und P-0-xxxx**

Die Module werden von links nach rechts (Steckplatz 1 bis 32) abgescannt und getrennt nach Eingang und Ausgang werden Identifier angelegt:

- Eingangskanäle werden in 10er-Schritten als S-2-ccc0 Identifier angelegt. Hierbei gilt ccc = 000 ... 255.  
Bereich: S-2-0000, S-2-0010, S-2-0020, ... S-2-2550
- Ausgangskanäle werden in 10er-Schritten als S-3-ccc0 Identifier angelegt. Hierbei gilt ccc = 000 ... 255  
Bereich: S-3-0000, S-3-0010, S-3-0020, ... S-3-2550
- Stecken parametrierbare Module, so wird pro Modul ein P-0-ssxx-Identifierblock angelegt. Hierbei gilt:  
Steckplatz: ss = 01 ... 32, Parameter: xx = 00 ... 17  
Beispiel: P-0-0100 (Modul in Steckplatz 1), P-0-0200 (Modul in Steckplatz 2), ... P-0-3200 (Modul in Steckplatz 32)

**VIPA-spezifische  
S-Identifier**

Für die S-Identifier existieren folgende Informationen:

**Name (besteht aus max. 32 Zeichen)**

*Format:* S.I.T\_W.D

mit

S = Steckplatz (1..32)

I = Modulinterner Byteoffset bei mehrkanaligen Modulen (0..15)

T = Typ: (DIGITAL, ANALOG)

W = Datenbreite: (BYTE, WORD, DOUBLE =1,2,4Byte)

D = Richtung: (IN,OUT)

*Beispiel:* Name: "1.0.DIGITAL\_BYTE.IN" bedeutet:

Das Module in Steckplatz 1 stellt ab seiner internen Adresse 0 ein Byte digitale Eingangsdaten zur Verfügung.

**Attribut**

Das Attribut legt gemäß der SERCOS-Spezifikation fest, ob das Betriebsdatum les- bzw. schreibbar ist. Näheres hierzu finden Sie in der SERCOS-Spezifikation des SERCOS-Arbeitskreis.

**Betriebsdatum**

Hier wird das Ein- bzw. Ausgabedatum mit seiner Datenbreite eingeblendet.

**VIPA-spezifische  
P-Identifizier  
(immer vorhanden)**

Im SERCOS-Koppler existieren immer die beiden Identifier P-0-0000 und P-0-0001.

**P-0-0000**

*Name:* WRITE\_PARAMETER

*Attribut:* Read/Write in Phase 0..3, Read Only in Phase 4

*Betriebsdatum:* 1 = Anstoß alle Parameter in EEPROM übernehmen.

2 = Anstoß alle Parameter in EEPROM löschen.

0 = Returnwert OK

65535 (FFFFhex) = Returnwert ERROR

**P-0-0001**

*Name:* Estimated SERCOS cycle time

*Attribut:* Read Only

*Einheit:* Mikrosekunden

*Betriebsdatum:* Die von Ihnen gewählte SERCOS-Zykluszeit muss immer größer sein als dieser Wert! (Z.B. 1460 bedeutet, dass die geschätzte Zykluszeit für den vorliegenden Modulaufbau 1,46ms ist und Sie somit mindestens einen SERCOS-Zyklus von 2ms wählen müssen.)

**18 VIPA-spezifische  
P-Identifizier  
(bei parametrier-  
baren Modulen)**

Sofern parametrierbare Module zum Einsatz kommen, werden dynamisch je parametrierbarem Modul ein 18 P-0-ssxx-Identifizier umfassender Block angelegt. Hierbei steht ss für Steckplatz (1 ... 32) und xx für die Parameter-Nr. (0 ... 17).

Prinzipiell haben diese zusätzlichen P-0-Identifizier folgenden Aufbau:

**P-0-ss00**

*Name:* ss.SLOT

*Attribut:* Read Only

*Betriebsdatum:* Gibt an, dass sich auf dem Steckplatz ein parametrierbares Modul befindet

**P-0-ss01**

*Name:* ss.LENGTH

*Attribut:* Read/Write in Phase 0 ... 3, Read Only in Phase 4

*Betriebsdatum:* Anzahl der nun folgenden Parameterbytes für dieses Modul (Wert: 0 ... 15).

**P-0-ss02**

*Name:* ss.PARAMETER.0

*Attribut:* Read/Write in Phase 0..3, Read Only in Phase 4

*Betriebsdatum:* Parameterbyte 0 (Wert: 0..255)

...

**P-0-ss17**

*Name:* ss.PARAMETER.15

*Attribut:* Read/Write in Phase 0..3, Read Only in Phase 4

*Betriebsdatum:* Parameterbyte 15 (Wert: 0..255)

Die Längenangabe und eine Beschreibung der zu übergebenden Parameter finden Sie in den entsprechenden Kapiteln zu den Modulen in diesem Handbuch.

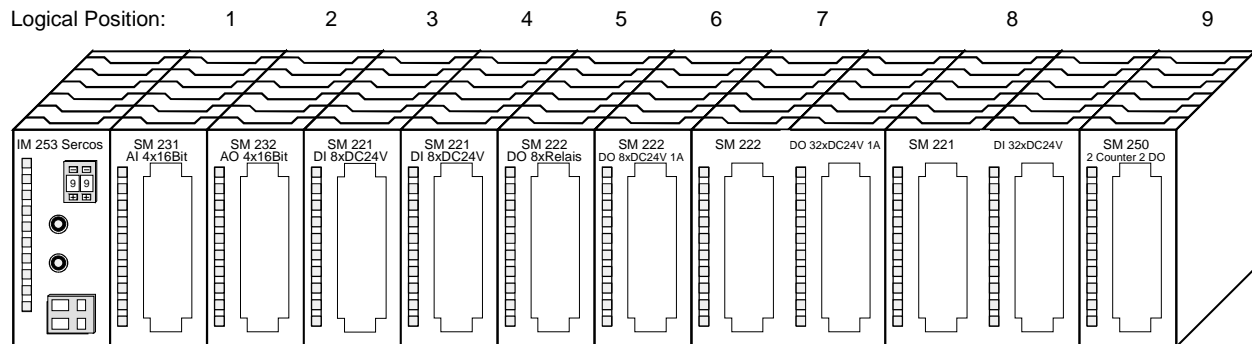


## Beispiel zur automatischen ID-Zuweisung

### Aufbau

Mit dem nachfolgenden Beispiel soll kurz gezeigt werden, wie die automatische Identifier-Zuweisung innerhalb des SERCOS-Kopplers abläuft.

Sie haben folgenden Aufbau:



Logischer Steckplatz	Modul	Input	Output	Parameter
1	VIPA 231-1BD52 (4 Kanal multi Analog Input)	ANALOG_WORD ANALOG_WORD ANALOG_WORD ANALOG_WORD		10 Byte
2	VIPA 232-1BD50 (4 Kanal multi Analog Output)		ANALOG_WORD ANALOG_WORD ANALOG_WORD ANALOG_WORD	6 Byte
3	VIPA 221-1BF00 (8bit digital Input)	DIGITAL_BYTE		-
4	VIPA 221-1BF00 (8bit digital Input)	DIGITAL_BYTE		-
5	VIPA 222-1HF00 (8bit digital Output, Relay)		DIGITAL_BYTE	-
6	VIPA 222-1BF00 (8bit digital Output, Transistor)		DIGITAL_BYTE	-
7	VIPA 222-2BL10 (32bit digital Output, Transistor)		DIGITAL_DOUBLE	-
8	VIPA 221-2BL10 (32bit digital Input)	DIGITAL_DOUBLE		-
9	VIPA 250-1BA00 (Counter Modul mit 2 mal 32Bit Counter und Steuerregister)	DIGITAL_DOUBLE DIGITAL_DOUBLE DIGITAL_BYTE DIGITAL_BYTE	DIGITAL_DOUBLE DIGITAL_DOUBLE DIGITAL_BYTE DIGITAL_BYTE	2 Byte

**Automatisch erzeugte Identifier** Für diesen Aufbau entstehen automatisch folgende Identifier:

*S-2-Identifier (Input)*

Identifier	Name	Kommentar
S-2-0000	1.0.ANALOG_WORD.IN	Modul in Steckplatz 1 Innerhalb des Moduls an Byteoffset 0 Ein analoges Wort Eingang
S-2-0010	1.2.ANALOG_WORD.IN	Modul in Steckplatz 1 Innerhalb des Moduls an Byteoffset 2 Ein analoges Wort Eingang
S-2-0020	1.4.ANALOG_WORD.IN	Modul in Steckplatz 1 Innerhalb des Moduls an Byteoffset 4 Ein analoges Wort Eingang
S-2-0030	1.6.ANALOG_WORD.IN	Modul in Steckplatz 1 Innerhalb des Moduls an Byteoffset 6 Ein analoges Wort Eingang
S-2-0040	3.0.DIGITAL_BYTE.IN	Modul in Steckplatz 3 Innerhalb des Moduls an Byteoffset 0 Ein digitales Byte Eingang
S-2-0050	4.0.DIGITAL_BYTE.IN	Modul in Steckplatz 4 Innerhalb des Moduls an Byteoffset 0 Ein digitales Byte Eingang
S-2-0060	8.0.DIGITAL_DOUBLE.IN	Modul in Steckplatz 8 Innerhalb des Moduls an Byteoffset 0 Ein digitales Doppelwort Eingang

*Fortsetzung ...*

... Fortsetzung

S-2-0070	9.0.DIGITAL_DOUBLE.IN	Modul in Steckplatz 9 Innerhalb des Moduls an Byteoffset 0 Ein digitales Doppelwort Eingang
S-2-0080	9.4.DIGITAL_DOUBLE.IN	Modul in Steckplatz 9 Innerhalb des Moduls an Byteoffset 4 Ein digitales Doppelwort Eingang
S-2-0090	9.8.DIGITAL_BYTE.IN	Modul in Steckplatz 9 Innerhalb des Moduls an Byteoffset 8 Ein digitales Byte Eingang
S-2-0100	9.9.DIGITAL_BYTE.IN	Modul in Steckplatz 9 Innerhalb des Moduls an Byteoffset 9 Ein digitales Byte Eingang

S-3-Identifizier (Output)

S-3-0000	2.0.ANALOG_WORD.OUT	Modul in Steckplatz 2 Innerhalb des Moduls an Byteoffset 0 Ein analoges Wort Ausgang
S-3-0010	2.2.ANALOG_WORD.OUT	Modul in Steckplatz 2 Innerhalb des Moduls an Byteoffset 2 Ein analoges Wort Ausgang
S-3-0020	2.4.ANALOG_WORD.OUT	Modul in Steckplatz 2 Innerhalb des Moduls an Byteoffset 4 Ein analoges Wort Ausgang
S-3-0030	2.6.ANALOG_WORD.OUT	Modul in Steckplatz 2 Innerhalb des Moduls an Byteoffset 6 Ein analoges Wort Ausgang
S-3-0040	5.0.DIGITAL_BYTE.OUT	Modul in Steckplatz 5 Innerhalb des Moduls an Byteoffset 0 Ein digitales Byte Ausgang

Fortsetzung ...

... Fortsetzung

S-3-0050	6.0.DIGITAL_BYTE.OUT	Modul in Steckplatz 6 Innerhalb des Moduls an Byteoffset 0 Ein digitales Byte Ausgang
S-3-0060	7.0.DIGITAL_DOUBLE.OUT	Modul in Steckplatz 7 Innerhalb des Moduls an Byteoffset 0 Ein digitales Doppelwort Ausgang
S-3-0070	9.0.DIGITAL_DOUBLE.OUT	Modul in Steckplatz 9 Innerhalb des Moduls an Byteoffset 0 Ein digitales Doppelwort Ausgang
S-3-0080	9.4.DIGITAL_DOUBLE.OUT	Modul in Steckplatz 9 Innerhalb des Moduls an Byteoffset 4 Ein digitales Doppelwort Ausgang
S-3-0090	9.8.DIGITAL_BYTE.OUT	Modul in Steckplatz 9 Innerhalb des Moduls an Byteoffset 8 Ein digitales Byte Ausgang
S-3-0100	9.9.DIGITAL_BYTE.OUT	Modul in Steckplatz 9 Innerhalb des Moduls an Byteoffset 9 Ein digitales Byte Ausgang

*P-0-Identifizier (Parameter) immer vorhanden*

P-0-0000	WRITE_PARAMETER	Hier Anstoß zum Schreiben/Löschen aller Parameter setzen: 1=Write, 2=Clear
P-0-0001	Estimated SERCOS cycle time	Wert hier: 1460 Mikrosekunden d.h. Sie können diesen Aufbau mit 2ms Sercos Zyklus betreiben.

*P-0-Identifizier (Parameter) bei parametrierbaren Modulen*

P-0-0100	1.SLOT	In Steckplatz 1 befindet sich ein parametrierbares Modul
P-0-0101	1.LENGTH	An das Modul in Steckplatz 1 sollen (Betriebsdatum) Bytes übertragen werden.
P-0-0102	1.PARAMETER.0	Parameterbyte0 für Modul in Steckplatz 1
P-0-0103	1.PARAMETER.1	Parameterbyte1 für Modul in Steckplatz 1
P-0-0104	1.PARAMETER.2	Parameterbyte2 für Modul in Steckplatz 1
P-0-0105	1.PARAMETER.3	Parameterbyte3 für Modul in Steckplatz 1
P-0-0106	1.PARAMETER.4	Parameterbyte4 für Modul in Steckplatz 1
P-0-0107	1.PARAMETER.5	Parameterbyte5 für Modul in Steckplatz 1
P-0-0108	1.PARAMETER.6	Parameterbyte6 für Modul in Steckplatz 1
P-0-0109	1.PARAMETER.7	Parameterbyte7 für Modul in Steckplatz 1
P-0-0110	1.PARAMETER.8	Parameterbyte8 für Modul in Steckplatz 1
P-0-0111	1.PARAMETER.9	Parameterbyte9 für Modul in Steckplatz 1
P-0-0112	1.PARAMETER.10	Parameterbyte10 für Modul in Steckplatz 1
P-0-0113	1.PARAMETER.11	Parameterbyte11 für Modul in Steckplatz 1
P-0-0114	1.PARAMETER.12	Parameterbyte12 für Modul in Steckplatz 1
P-0-0115	1.PARAMETER.13	Parameterbyte13 für Modul in Steckplatz 1
P-0-0116	1.PARAMETER.14	Parameterbyte14 für Modul in Steckplatz 1
P-0-0117	1.PARAMETER.15	Parameterbyte15 für Modul in Steckplatz 1
P-0-0200	2.SLOT	In Steckplatz 2 befindet sich ein parametrierbares Modul
P-0-0201	2.LENGTH	An das Modul in Steckplatz 2 sollen (Betriebsdatum) Bytes übertragen werden.
P-0-0202	2.PARAMETER.0	Parameterbyte0 für Modul in Steckplatz 2
P-0-0203	2.PARAMETER.1	Parameterbyte1 für Modul in Steckplatz 2
P-0-0204	2.PARAMETER.2	Parameterbyte2 für Modul in Steckplatz 2
P-0-0205	2.PARAMETER.3	Parameterbyte3 für Modul in Steckplatz 2
P-0-0206	2.PARAMETER.4	Parameterbyte4 für Modul in Steckplatz 2
P-0-0207	2.PARAMETER.5	Parameterbyte5 für Modul in Steckplatz 2
P-0-0208	2.PARAMETER.6	Parameterbyte6 für Modul in Steckplatz 2
P-0-0209	2.PARAMETER.7	Parameterbyte7 für Modul in Steckplatz 2
P-0-0210	2.PARAMETER.8	Parameterbyte8 für Modul in Steckplatz 2
P-0-0211	2.PARAMETER.9	Parameterbyte9 für Modul in Steckplatz 2
P-0-0212	2.PARAMETER.10	Parameterbyte10 für Modul in Steckplatz 2
P-0-0213	2.PARAMETER.11	Parameterbyte11 für Modul in Steckplatz 2
P-0-0214	2.PARAMETER.12	Parameterbyte12 für Modul in Steckplatz 2
P-0-0215	2.PARAMETER.13	Parameterbyte13 für Modul in Steckplatz 2
P-0-0216	2.PARAMETER.14	Parameterbyte14 für Modul in Steckplatz 2
P-0-0217	2.PARAMETER.15	Parameterbyte15 für Modul in Steckplatz 2

*Fortsetzung ...*

... Fortsetzung

P-0-0900	9.SLOT	In Steckplatz 9 befindet sich ein parametrierbares Modul
P-0-0901	9.LENGTH	An das Modul in Steckplatz 9 sollen (Betriebsdatum) Bytes übertragen werden.
P-0-0902	9.PARAMETER.0	Parameterbyte0 für Modul in Steckplatz 9
P-0-0903	9.PARAMETER.1	Parameterbyte1 für Modul in Steckplatz 9
P-0-0904	9.PARAMETER.2	Parameterbyte2 für Modul in Steckplatz 9
P-0-0905	9.PARAMETER.3	Parameterbyte3 für Modul in Steckplatz 9
P-0-0906	9.PARAMETER.4	Parameterbyte4 für Modul in Steckplatz 9
P-0-0907	9.PARAMETER.5	Parameterbyte5 für Modul in Steckplatz 9
P-0-0908	9.PARAMETER.6	Parameterbyte6 für Modul in Steckplatz 9
P-0-0909	9.PARAMETER.7	Parameterbyte7 für Modul in Steckplatz 9
P-0-0910	9.PARAMETER.8	Parameterbyte8 für Modul in Steckplatz 9
P-0-0911	9.PARAMETER.9	Parameterbyte9 für Modul in Steckplatz 9
P-0-0912	9.PARAMETER.10	Parameterbyte10 für Modul in Steckplatz 9
P-0-0913	9.PARAMETER.11	Parameterbyte11 für Modul in Steckplatz 9
P-0-0914	9.PARAMETER.12	Parameterbyte12 für Modul in Steckplatz 9
P-0-0915	9.PARAMETER.13	Parameterbyte13 für Modul in Steckplatz 9
P-0-0916	9.PARAMETER.14	Parameterbyte14 für Modul in Steckplatz 9
P-0-0917	9.PARAMETER.15	Parameterbyte15 für Modul in Steckplatz 9

**Beispiel-  
parametrierung**

Beispielsweise sollen folgende Werte gesetzt werden:

**AI 4x16Bit (231-1BD52) auf Position 1**

Länge 10Byte

Parameter:

Byte	Beschreibung	Soll-Eigenschaft	Übergabewert
0	Diagnosealarm-Byte:	deaktiviert	00h = 0dez
1	reserviert	00h	00h = 0dez
2	Funktions-Nr. Kanal 0	Spannung ±10V im S7-Format von Siemens	28h = 40dez
3	Funktions-Nr. Kanal 1	Spannung ±10V im S7-Format von Siemens	28h = 40dez
4	Funktions-Nr. Kanal 2	Strom 4...20mA im S7-Format von Siemens	2Dh = 45dez
5	Funktions-Nr. Kanal 3	Strom 4...20mA im S7-Format von Siemens	2Dh = 45dez
6	Option-Byte Kanal 0	default	00h = 0dez
7	Option-Byte Kanal 1	default	00h = 0dez
8	Option-Byte Kanal 2	default	00h = 0dez
9	Option-Byte Kanal 3	default	00h = 0dez

Hierbei ergeben sich für die Tabelle folgende Einträge:

P-0-0100	1.SLOT	In Steckplatz 1 befindet sich ein parametrierbares Modul
P-0-0101	1.LENGTH	<u>10dez</u>
P-0-0102	1.PARAMETER.0	0dez
P-0-0103	1.PARAMETER.1	0dez
P-0-0104	1.PARAMETER.2	40dez
P-0-0105	1.PARAMETER.3	40dez
P-0-0106	1.PARAMETER.4	45dez
P-0-0107	1.PARAMETER.5	45dez
P-0-0108	1.PARAMETER.6	0dez
P-0-0109	1.PARAMETER.7	0dez
P-0-0110	1.PARAMETER.8	0dez
P-0-0111	1.PARAMETER.9	0dez
P-0-0112	1.PARAMETER.10	werden angelegt aber nicht benutzt
...	...	
P-0-0117	1.PARAMETER.15	

Setzen Sie in **P-0-0000** den Wert auf 1 und die Parameter werden im SERCOS-Koppler im EEPROM gesichert.

Bei erfolgreicher Übertragung erhalten Sie den Returnwert 0 und am analogen Eingabemodul leuchten aufgrund des Strommessbereichs die LEDs F2 und F3 für Drahtbruchkennung.

**AO 4x16Bit (232-1BD50) auf Position 2**

Länge: 6Byte

Parameter:

Byte	Beschreibung	Soll-Eigenschaft	Übergabewert
0	Diagnosealarm-Byte:	deaktiviert	00h = 0dez
1	reserviert	00h	00h = 0dez
2	Funktions-Nr. Kanal 0	Spannung ±10V im S7-Format von Siemens	09h = 9dez
3	Funktions-Nr. Kanal 1	Spannung ±10V im S7-Format von Siemens	09h = 9dez
4	Funktions-Nr. Kanal 2	Strom 4...20mA im S7-Format von Siemens	0Ch = 12dez
5	Funktions-Nr. Kanal 3	Strom 4...20mA im S7-Format von Siemens	0Ch = 12dez

Hierbei ergeben sich für die Tabelle folgende Einträge:

P-0-0200	2.SLOT	In Steckplatz 2 befindet sich ein parametrierbares Modul
P-0-0201	2.LENGTH	6dez
P-0-0202	2.PARAMETER.0	0dez
P-0-0203	2.PARAMETER.1	0dez
P-0-0204	2.PARAMETER.2	9dez
P-0-0205	2.PARAMETER.3	9dez
P-0-0206	2.PARAMETER.4	12dez
P-0-0207	2.PARAMETER.5	12dez
P-0-0208	2.PARAMETER.6	werden angelegt aber nicht benutzt
...	...	
P-0-0217	2.PARAMETER.15	

Setzen Sie in **P-0-0000** den Wert auf 1 und die Parameter werden im SERCOS-Koppler im EEPROM gesichert.

Bei erfolgreicher Übertragung erhalten Sie den Returnwert 0 und am analogen Ausgabemodul leuchten aufgrund des Strommessbereichs die LED für Drahtbruchkennung.



**SM 250 2 Counter 2 DO (250-1BA00) auf Position 2**

Länge: 2Byte

Parameter:

Byte	Beschreibung	Soll-Eigenschaft	Übergabewert
0	Modus Zähler 0	Frequenz- messung	16dez
1	Modus Zähler 1		16dez

Hierbei ergeben sich für die Tabelle folgende Einträge:

P-0-0900	9.SLOT	In Steckplatz 9 befindet sich ein parametrierbares Modul
P-0-0901	9.LENGTH	2dez
P-0-0902	9.PARAMETER.0	16dez
P-0-0903	9.PARAMETER.1	16dez
P-0-0904	9.PARAMETER.2	werden angelegt aber nicht benutzt
...	...	
P-0-0917	9.PARAMETER.15	

Setzen Sie in **P-0-0000** den Wert auf 1 und die Parameter werden im SERCOS-Koppler im EEPROM gesichert.

Bei erfolgreicher Übertragung erhalten Sie den Returnwert 0.

## Technische Daten

### SERCOS-Koppler IM 253SC

Elektrische Daten	VIPA 253-1SC00
Spannungsversorgung Stromaufnahme	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil Buskoppler: 50mA inkl. Versorgung der Peripheriemodule: max. 3,5A (5V)
Ausgangsstrom Rückwandbus Potenzialtrennung zum Rückwandbus	max. 3,5A 500V eff.
Funktionsspezifische Daten	
Statusanzeige Physikalischer Anschluss SERCOS Netzwerk-Topologie Übertragungsmedium Übertragungsrate Anzahl der Teilnehmer	über LED auf der Frontseite LWL-Buchsen Ring Lichtwellenleiter mit 1mm Fiberglas bzw. 200µm HCS® 2, 4, 8, 16MBaud max. 89
Kombination mit Peripheriemodulen	
Modulanzahl Eingänge Ausgänge	max. 32 max. 256Byte max. 256Byte
Mechanische Daten	
Abmessungen (BxHxT) Gewicht	25,4x76x76mm 75g

## Teil 7 Ethernet-Koppler

### Überblick

Inhalt dieses Kapitels die Beschreibung des Ethernet-Kopplers IM 253NET von VIPA. Sie bekommen hier alle Informationen, die für Aufbau und Inbetriebnahme des Ethernet-Kopplers erforderlich sind.

Das Kapitel beginnt mit den Grundlagen. Hier sind die Grundbegriffe der Ethernet-Kommunikation aufgeführt zusammen mit den Richtlinien für den Aufbau eines Netzwerks.

Ein weiterer Teil befasst sich mit dem Hardware-Komponenten und mit den Zugriffsmöglichkeiten auf den Ethernet-Koppler.

Mit einer Beschreibung der verwendeten Protokolle, einem Beispiel zur Socketprogrammierung und den technischen Daten endet das Kapitel.

Nachfolgend sind beschrieben:

- Systemübersicht
- Grundlagen zum Thema Ethernetkommunikation
- Aufbau des Ethernet-Kopplers
- Prinzip der automatische Adressierung
- (Online-)Zugriffsmöglichkeiten auf den Ethernet-Koppler
- Programmierbeispiel
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 7 Ethernet-Koppler</b> .....	<b>7-1</b>
Systemübersicht .....	7-2
Grundlagen Ethernet .....	7-3
Planung eines Netzwerks .....	7-7
IM 253NET - Ethernet-Koppler - Aufbau .....	7-9
Zugriffsmöglichkeiten auf den Ethernet-Koppler .....	7-11
Prinzip der automatischen Adressierung.....	7-14
Projektierung unter WinNCS.....	7-15
Diagnose und Test mittels Internet Browser .....	7-16
ModbusTCP.....	7-20
Modbus-Funktionscodes.....	7-21
Siemens S5 Header Protokoll.....	7-25
Programmierbeispiel.....	7-27
Technische Daten.....	7-28

## Systemübersicht

In typischen Feldbussystemen unterscheidet man zwischen Master- und Slave-Systemen.

Master-Systeme sind an die CPU angekoppelte CPs, die eine Fernprogrammierung bzw. Visualisierung der entsprechenden CPU erlauben sowie den Datenaustausch zwischen mehreren TCP/IP-Teilnehmern ermöglichen.

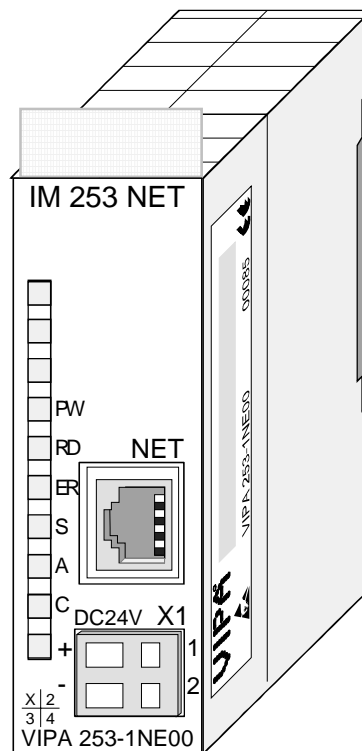
Slave-Systeme hingegen sind "Datensammler", die dem anfragenden Master die E/A-Daten der angesteckten Module zur Verfügung stellen.

Der in diesem Kapitel vorgestellte Ethernet-Koppler ist ein Slave-System.

Da aber die Kommunikation über TCP/IP erfolgt, bezeichnet man das Slave-System als Server und einen Master als Client.

Mit dem Ethernet-Koppler von VIPA können Sie bis zu 32 Module Ihrer System 200V Peripherie über Ethernet ankoppeln. Bis zu 8 Clients können je Protokoll mit dem Ethernet-Koppler gleichzeitig kommunizieren.

Folgender Ethernet-Koppler ist zur Zeit von VIPA verfügbar:



Bestelldaten  
Ethernet-Koppler

Typ	Bestellnummer	Beschreibung
IM 253NET	VIPA 253-1NE00	Ethernet-Koppler

## Grundlagen Ethernet

### Ethernet

Ethernet wurde ursprünglich von DEC, Intel und Xerox (als DIX-Standard) für die Datenübertragung zwischen Bürogeräten entwickelt. Heute versteht man darunter meist die Spezifikation *IEEE 802.3 CSMA/CD*, die 1985 veröffentlicht wurde. Diese Technologie ist durch ihren weltweiten Einsatz und die hohen Stückzahlen überall erhältlich und sehr preiswert. Eine Anbindung an vorhandene Netze kann so problemlos realisiert werden.

Ethernet transportiert Ethernet-Pakete von einem Sender zu einem oder mehreren Empfängern. Diese Übertragung verläuft ohne Quittung und ohne Wiederholung von verlorenen Paketen. Für die sichere Daten-Kommunikation stehen Protokolle wie TCP/IP zu Verfügung, die auf Ethernet aufsetzen.

### Twisted Pair

Früher gab es das Triaxial- (Yellow Cable) oder Thin Ethernet-Kabel (Cheapernet). Mittlerweile hat sich aber aufgrund der Störfestigkeit das preisgünstige Twisted Pair Netzwerkkabel durchgesetzt. Der IM 253NET Ethernet-Koppler besitzt einen Twisted-Pair-Anschluss.

Abweichend von den beiden Ethernet-Koaxialnetzen, die auf einer Bus-Topologie aufbauen, bildet Twisted Pair ein Punkt-zu-Punkt-Kabelschema.

Das hiermit aufzubauende Netz stellt eine Stern-Topologie dar. Jede Station ist einzeln direkt mit dem Sternkoppler (Hub/Switch) zu einem Ethernet verbunden.

### Hub

Ein Hub ist ein zentrales Element zur Realisierung von Ethernet auf Twisted Pair. Seine Aufgabe ist dabei, die Signale in beide Richtungen zu regenerieren und zu verstärken. Gleichzeitig muss er in der Lage sein, segmentübergreifende Kollisionen zu erkennen, zu verarbeiten und weiter zu geben. Er kann nicht im Sinne einer eigenen Netzwerkadresse angesprochen werden, da er von den angeschlossenen Stationen nicht registriert wird. Er bietet Möglichkeiten zum Anschluss an Ethernet oder zu einem anderen Hub bzw. Switch.

### Switch

Ein Switch ist ebenfalls ein zentrales Element zur Realisierung von Ethernet auf Twisted Pair. Mehrere Stationen bzw. Hubs werden über einen Switch verbunden. Diese können dann, ohne das restliche Netzwerk zu belasten, über den Switch miteinander kommunizieren. Eine intelligente Hardware analysiert für jeden Port in einem Switch die eingehenden Telegramme und leitet diese kollisionsfrei direkt an die Zielstationen weiter, die am Switch angeschlossen sind. Ein Switch sorgt für die Optimierung der Bandbreite in jedem einzeln angeschlossenen Segment eines Netzes. Switches ermöglichen exklusiv nach Bedarf wechselnde Verbindungen zwischen angeschlossenen Segmenten eines Netzes.

**Zugriffssteuerung** Bei Ethernet gibt es das Prinzip des zufälligen Buszugriffs: Jeder Teilnehmer greift bei Bedarf von sich aus auf den Bus zu. Koordiniert wird der Buszugriff dabei durch das Verfahren CSMA/CD (Carrier Sense Multiple Access/Collision Detection - Mithören bei Mehrfachzugriff/ Kollisionserkennung): Jeder Teilnehmer "hört" ständig die Busleitung ab und empfängt die an ihn adressierten Sendungen.

Ein Teilnehmer startet eine Sendung nur, wenn die Leitung frei ist. Starten zwei Teilnehmer gleichzeitig eine Sendung, so erkennen sie dies, stellen die Sendung ein und starten nach einer Zufallszeit erneut.

**Kommunikation** Der Ethernet-Koppler ist über den Rückwandbus mit den Modulen verbunden. Er sammelt deren Daten und stellt sie als "Server" (Slave) einem übergeordneten "Client" (Master-System) zur Verfügung.

Die Kommunikation erfolgt über TCP/IP mit aufgesetztem ModbusTCP- oder dem Siemens S5 Header Protokoll.

Umgekehrt empfängt der Ethernet-Koppler die an ihn über IP-Adresse und Port adressierten Daten und gibt diese an seine Ausgabe-Peripherie weiter. Zur Projektierung dient das Projektierool WinNCS von VIPA. Hier können Sie online den Ethernet-Koppler projektieren.

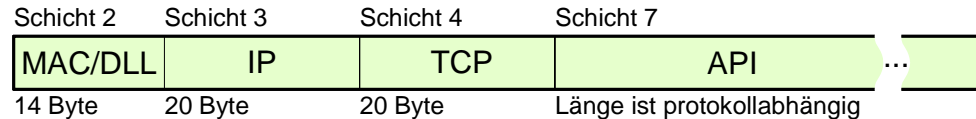
Für Test und Diagnose stellt der Ethernet-Slave einen Web-Server zur Verfügung, der lesenden und schreibenden Zugriff auf die E/A-Peripherie sowie die Parametrierung von Modulen erlaubt.

**Übersicht  
Protokolle**

In Protokollen ist ein Satz an Vorschriften oder Standards definiert, der es Computern ermöglichen, Kommunikationsverbindungen herzustellen und Informationen möglichst fehlerfrei auszutauschen.

Ein allgemein anerkanntes Modell für die Standardisierung der kompletten Computerkommunikation stellt das sog. ISO/OSI-Schichtenmodell dar, ein auf sieben Schichten basierendes Modell mit Richtlinien, die den Einsatz von Hardware und Software regeln.

Schicht	Funktion	Protokoll
Schicht 7	Application Layer (Anwendung)	Siemens S5 Header, ModbusTCP
Schicht 6	Presentation Layer (Darstellung)	
Schicht 5	Session Layer (Sitzung)	
Schicht 4	Transport Layer (Transport)	TCP
Schicht 3	Network Layer (Netzwerk)	IP
Schicht 2	Data Link Layer (Sicherung)	
Schicht 1	Physical Layer (Bitübertragung)	

**Telegrammaufbau****MAC/DLL**

Während die Ethernet-Physik mit seinem genormten Signalpegel die Schicht 1 abdeckt, erfüllt MAC/DLL die Vorgaben für die Sicherungsschicht (Schicht 2). Bei MAC (**M**edium **A**ccess **C**ontrol) / DLL (**D**ata **L**ink **L**ayer) erfolgt die Kommunikation auf unterster Ethernetebene unter Zuhilfenahme von MAC-Adressen. Jeder ethernetfähige Kommunikationsteilnehmer besitzt eine eindeutige MAC-Adresse, die nur einmal vorhanden sein darf. Durch Einsatz von MAC-Adressen werden Quelle und Ziel eindeutig spezifiziert.

**IP**

Das Internet Protokoll deckt die Netzwerkschicht (Schicht 3) des ISO/OSI-Schichtmodells ab.

Die Aufgabe des IP besteht darin, Datenpakete von einem Rechner über mehrere Rechner hinweg zum Empfänger zu senden. Diese Datenpakete sind sogenannte Datagramme. Das IP gewährleistet weder die richtige Reihenfolge der Datagramme, noch die Ablieferung beim Empfänger.

Zur eindeutigen Unterscheidung zwischen Sender und Empfänger kommen 32Bit-Adressen (IP-Adressen) zum Einsatz, die normalerweise in vier Oktetts (genau 8Bit) geschrieben werden, z.B. 172.16.192.11. Bei einem Oktett können Zahlen zwischen 0 und 255 dargestellt werden.

Ein Teil der Adresse spezifiziert das Netzwerk, der Rest dient zur Identifizierung der Rechner im Netzwerk. Die Grenze zwischen Netzwerkanteil und Host-Anteil ist fließend und hängt von der Größe des Netzwerkes ab.

**TCP**

Das TCP (Transmission Control Protokoll) setzt direkt auf dem IP auf, somit deckt das TCP die Transportschicht (Schicht 4) auf dem OSI-Schichtenmodell ab. TCP ist ein verbindungsorientiertes End-to-End-Protokoll und dient zur logischen Verbindung zwischen zwei Partnern.

TCP gewährleistet eine folgerichtige und zuverlässige Datenübertragung.

Jedes Datagramm wird mit einem mindestens 20 Byte langen Header versehen, der unter anderem auch eine Folgenummer für die richtige Reihenfolge beinhaltet. So können in einem Netzwerkverbund die einzelnen Datagramme auf unterschiedlichen Wegen zum Ziel gelangen.

**API**

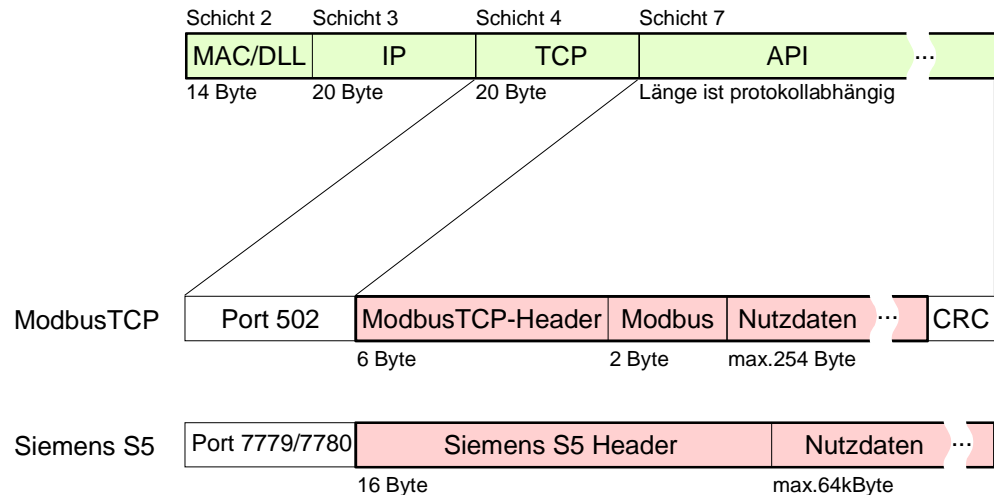
API steht für **A**pplication **P**rogramming **I**nterface. API erfüllt die Vorgaben für den Application Layer (Schicht 7).

Hier sind Header und Nutzdaten der entsprechenden Protokolle abgelegt.

Im Ethernet-Koppler IM 253NET von VIPA kommen folgende Protokolle zum Einsatz, die nachfolgend näher erläutert werden:

- ModbusTCP
- Siemens S5 Header

## API-Aufbau



## ModbusTCP

ModbusTCP ist ein auf TCP/IP aufgesetztes Modbus-RTU-Protokoll.

Das Protokoll Modbus ist ein Kommunikationsprotokoll, das eine hierarchische Struktur mit einem Master und mehreren Slaves unterstützt. ModbusTCP erweitert Modbus zu einer Client-Server-Kommunikation, wobei mehrere Clients auf einen Server zugreifen können.

Da über IP-Adressen die Adressierung erfolgt, ist die im Modbus-Telegramm eingebettete Adresse irrelevant. Auch ist die CRC-Checksumme nicht erforderlich, da die Sicherung über TCP/IP erfolgt.

Nach einer Anforderung eines Clients wartet dieser solange auf die Antwort des Servers, bis eine einstellbare Wartezeit abgelaufen ist.

Bei ModbusTCP kommt ausschließlich das RTU-Format zum Einsatz:

Hierbei wird jedes Byte als ein Zeichen übertragen. Somit haben Sie einen höheren Datendurchsatz als im Modbus-ASCII-Format. Die RTU-Zeitüberwachung entfällt, da der Header die Größe der zu empfangenden Telegrammlänge beinhaltet.

Daten, die mit ModbusTCP übertragen werden, können Bit- und Wort-Informationen enthalten. Hierbei wird bei Bitketten das höchstwertige Bit zuerst gesendet, d.h. es steht innerhalb eines Wortes ganz links. Bei Worten wird das höchstwertige Byte zuerst gesendet.

Der Zugriff auf einen Modbus-Slave erfolgt über Funktions-Codes, die in diesem Kapitel weiter unten näher erläutert sind.

## Siemens S5 Header

Das Siemens S5 Header-Protokoll dient zum Datenaustausch zwischen SPS-Systemen. Unter Einsatz des Organisationsformats (kurz ORG), das in das Siemens S5 Header-Protokoll eingebettet ist, ist die Kurzbeschreibung einer Datenquelle bzw. eines Datenziels in SPS-Umgebung möglich.

Die verwendbaren ORG-Formate entsprechen den Siemens-Vorgaben.

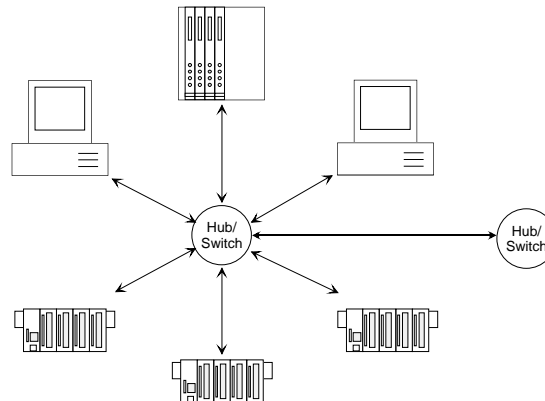


## Planung eines Netzwerks

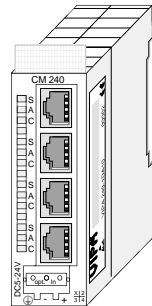
<b>Allgemeines</b>	<p>Das Hauptkennzeichen einer Busstruktur ist, dass nur ein einziger physikalischer Übertragungsweg existiert. Als physikalisches Übertragungsmedium wird dabei verwendet:</p> <ul style="list-style-type: none"> <li>• ein oder mehrere elektrische Leitungen (verdrillte Leitung)</li> <li>• Koaxialkabel (Triaxialkabel)</li> <li>• Lichtwellenleiter</li> </ul> <p>Um die Kommunikation zwischen den einzelnen Stationen zu ermöglichen, müssen Vorschriften und Regeln verabredet und eingehalten werden. Die Vereinbarungen regeln die Form des Datenprotokolls, das Zugriffsverfahren auf den Bus und weitere, für die Kommunikation wichtige Grundlagen. Basierend auf den von ISO festgelegten Standards und Normen wurde der Ethernet-Koppler IM 253NET von VIPA entwickelt.</p>														
<b>Normen und Richtlinien</b>	<p>Folgende Normen und Richtlinien im Zusammenhang mit Netzwerktechnologien sind von internationalen und nationalen Gremien festgelegt worden:</p> <table> <tr> <td style="vertical-align: top;">ANSI</td> <td>American National Standards Institute Hier werden zur Zeit in der ANSI X3T9.5 Vereinbarungen für LANs mit hohen Übertragungsgeschwindigkeiten (100 MB/s) auf Glasfaserbasis formuliert. (FDDI) Fibre Distributed Data Interface.</td> </tr> <tr> <td style="vertical-align: top;">CCITT</td> <td>Committee Consultative Internationale de Telephone et Telegraph. Von diesem beratenden Ausschuß werden unter anderem die Vereinbarungen für die Anbindung von Industriekommunikationsnetzen (MAP) und Büronetzen (TOP) an Wide Area Networks (WAN) erstellt.</td> </tr> <tr> <td style="vertical-align: top;">ECMA</td> <td>European Computer Manufacturers Association. Hier werden verschiedene Standards für MAP und TOP erarbeitet.</td> </tr> <tr> <td style="vertical-align: top;">EIA</td> <td>Electrical Industries Association (USA) Standardfestlegungen wie RS-232 (V.24) und RS-511 sind in diesem Ausschuß erarbeitet worden.</td> </tr> <tr> <td style="vertical-align: top;">IEC</td> <td>International Electrotechnical Commision. Hier werden einzelne spezielle Standards festgelegt. z.B. für Feld Bus.</td> </tr> <tr> <td style="vertical-align: top;">ISO</td> <td>International Organisation for Standardization. In diesem Verband der nationalen Normungsstellen wurde das OSI-Modell entwickelt (ISO/TC97/SC16). Es gibt den Rahmen vor, an den sich die Normungen für die Datenkommunikation halten sollen. ISO Standards gehen über in die einzelnen nationalen Standards wie z.B. UL und DIN.</td> </tr> <tr> <td style="vertical-align: top;">IEEE</td> <td>Institute of Electrical and Electronic Engineers (USA). In der Projektgruppe 802 werden die LAN-Standards für Übertragungsraten von 1 bis 20 MB/s festgelegt. IEEE Standards bilden häufig die Grundlage für ISO-Standards z.B. IEEE 802.3 = ISO 8802.3.</td> </tr> </table>	ANSI	American National Standards Institute Hier werden zur Zeit in der ANSI X3T9.5 Vereinbarungen für LANs mit hohen Übertragungsgeschwindigkeiten (100 MB/s) auf Glasfaserbasis formuliert. (FDDI) Fibre Distributed Data Interface.	CCITT	Committee Consultative Internationale de Telephone et Telegraph. Von diesem beratenden Ausschuß werden unter anderem die Vereinbarungen für die Anbindung von Industriekommunikationsnetzen (MAP) und Büronetzen (TOP) an Wide Area Networks (WAN) erstellt.	ECMA	European Computer Manufacturers Association. Hier werden verschiedene Standards für MAP und TOP erarbeitet.	EIA	Electrical Industries Association (USA) Standardfestlegungen wie RS-232 (V.24) und RS-511 sind in diesem Ausschuß erarbeitet worden.	IEC	International Electrotechnical Commision. Hier werden einzelne spezielle Standards festgelegt. z.B. für Feld Bus.	ISO	International Organisation for Standardization. In diesem Verband der nationalen Normungsstellen wurde das OSI-Modell entwickelt (ISO/TC97/SC16). Es gibt den Rahmen vor, an den sich die Normungen für die Datenkommunikation halten sollen. ISO Standards gehen über in die einzelnen nationalen Standards wie z.B. UL und DIN.	IEEE	Institute of Electrical and Electronic Engineers (USA). In der Projektgruppe 802 werden die LAN-Standards für Übertragungsraten von 1 bis 20 MB/s festgelegt. IEEE Standards bilden häufig die Grundlage für ISO-Standards z.B. IEEE 802.3 = ISO 8802.3.
ANSI	American National Standards Institute Hier werden zur Zeit in der ANSI X3T9.5 Vereinbarungen für LANs mit hohen Übertragungsgeschwindigkeiten (100 MB/s) auf Glasfaserbasis formuliert. (FDDI) Fibre Distributed Data Interface.														
CCITT	Committee Consultative Internationale de Telephone et Telegraph. Von diesem beratenden Ausschuß werden unter anderem die Vereinbarungen für die Anbindung von Industriekommunikationsnetzen (MAP) und Büronetzen (TOP) an Wide Area Networks (WAN) erstellt.														
ECMA	European Computer Manufacturers Association. Hier werden verschiedene Standards für MAP und TOP erarbeitet.														
EIA	Electrical Industries Association (USA) Standardfestlegungen wie RS-232 (V.24) und RS-511 sind in diesem Ausschuß erarbeitet worden.														
IEC	International Electrotechnical Commision. Hier werden einzelne spezielle Standards festgelegt. z.B. für Feld Bus.														
ISO	International Organisation for Standardization. In diesem Verband der nationalen Normungsstellen wurde das OSI-Modell entwickelt (ISO/TC97/SC16). Es gibt den Rahmen vor, an den sich die Normungen für die Datenkommunikation halten sollen. ISO Standards gehen über in die einzelnen nationalen Standards wie z.B. UL und DIN.														
IEEE	Institute of Electrical and Electronic Engineers (USA). In der Projektgruppe 802 werden die LAN-Standards für Übertragungsraten von 1 bis 20 MB/s festgelegt. IEEE Standards bilden häufig die Grundlage für ISO-Standards z.B. IEEE 802.3 = ISO 8802.3.														

**Übersicht der Komponenten**

Sie können ein Twisted Pair-Netzwerk nur sternförmig aufbauen.



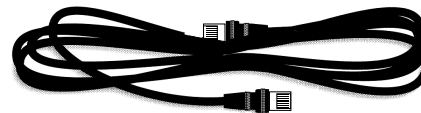
Mini-Switch CM 240



Twisted Pair Kabel

Bei einem Twisted Pair-Kabel handelt es sich um ein Kabel mit vier Adern, die paarweise miteinander verdreht sind.

Die einzelnen Adern haben einen jeweiligen Durchmesser von 0,4 bis 0,6 mm.



**Einschränkungen**

Hier ist eine Zusammenfassung der Einschränkungen und Regeln bezüglich Twisted Pair:

- Maximale Anzahl von Kopplerelementen pro Segment 2
- Maximale Länge eines Segments 100m

**Ermitteln des Netzwerkbedarfs**

- Welche Fläche muss mit dem Kabelsystem abgedeckt werden?
- Wie viele Netzwerksegmente lösen am besten die physikalischen (räumlich, störungsbedingt) Gegebenheiten der Anlage?
- Wie viele Netzwerkstationen (SPS, IPC, PC, Transceiver, evtl. Bridges) sollen an das Kabelsystem angeschlossen werden?
- In welchem Abstand stehen die Netzwerkstationen voneinander getrennt?
- Welches "Wachstum" in Größe und Anzahl der Verbindungen muss das System bewältigen können?
- Welches Datenaufkommen ist zu bewältigen (Bandbreite, Zugriffe/Sec.)?

**Zeichnen des Netzwerkplans**

Zeichnen Sie Ihren Netzwerkplan. Bezeichnen Sie jedes Stück Hardware das verwendet wird (wie Stationskabel, Hub, Switch). Halten Sie die Regeln und Grenzwerte im Auge.

Messen Sie die Distanz zwischen allen Komponenten um sicher zu gehen, dass die maximale Länge nicht überschritten wird.

## IM 253NET - Ethernet-Koppler - Aufbau

### Eigenschaften

- Ethernet-Koppler mit ModbusTCP und Siemens S5 Header Protokoll
- max. 32 Module ansteckbar mit max. 256Byte Eingabe- und 256Byte Ausgabe-Daten
- E/A-Zugriff mit beiden Protokollen über PC-Software wie beispielsweise OPC-Server von VIPA
- Online-Projektierung unter WinNCS von VIPA mit automatischer Kopplersuche und Parametrierung von Modulen in Klartext. Hier können Sie auch IP-Adresse, Subnetmask und Kopplername vorgeben und ein Firmwareupdate durchführen.
- Web-Server für Test und Diagnose integriert
- RJ45-Buchse 100BaseTX, 10BaseT
- Automatische Polaritäts- und Baudratenerkennung (auto negotiation)
- Automatische Erkennung paralleles oder gekreuztes Kabel (auto crossover)
- Netzwerk-LEDs für link/activity, speed und collision
- Status-LEDs für Ready und Error

### Auslieferungszustand

IP-Adresse: 10.0.0.1

Passwort für Änderungszugriffe über WinNCS: 00000000

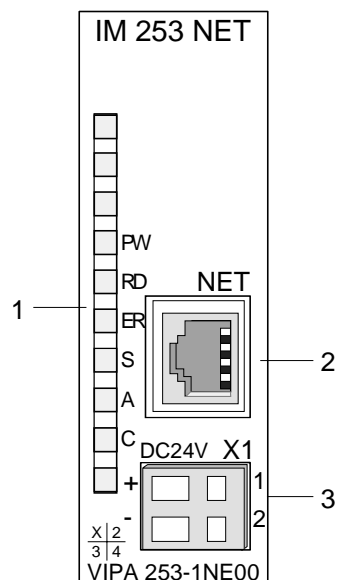


### Achtung!

Da jeder Ethernet-Koppler mit der IP-Adresse 10.0.0.1 ausgeliefert wird, dürfen sich bei der Erstinbetriebnahme nicht mehrere neue Ethernet-Koppler im Netz befinden!

Erstinbetriebnahme: Neuen Koppler mit Netzwerk verbinden, TCP/IP-Adresse vergeben, nächsten neuen Koppler verbinden usw. ...

### Frontansicht IM 253 NET



- [1] LED Statusanzeigen
- [2] RJ45-Buchse für Twisted Pair Anschluss
- [3] Anschluss für DC 24V Spannungsversorgung

**Komponenten**

**LEDs**

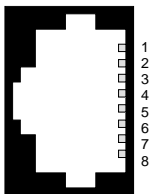
Der Ethernet-Koppler besitzt verschiedene LEDs, die der Diagnose dienen und den eigenen Betriebszustand anzeigen. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bez.	Farbe	Bedeutung
PW	Gelb	Power: Signalisiert ein anliegende DC 24V Spannungsversorgung
RD	Grün	Ready: Der Ethernet-Koppler ist hochgelaufen. Am Rückwandbus gesteckte E/A-Peripherie kann angesprochen werden.
ER	Rot	Error: Zeigt einen Fehler an wie beispielsweise Modulausfall oder Parametrierfehler (Details: siehe Koppler-Web-Site)
S	Grün	Speed: an: 100MBit aus: 10MBit
A	Grün	Activity: an: physikalisch verbunden aus: keine physikalische Verbindung blinkt: zeigt Busaktivität an
C	Grün	Collision: an: Vollduplexbetrieb aktiv aus: Halbduplexbetrieb aktiv blinkt: Collision detected

**RJ45 Ethernet-Anschluss**

Über die RJ45-Buchse haben Sie einen Twisted-Pair-Anschluss an Ethernet. Die Buchse hat folgende Belegung:

*8-polige RJ45-Buchse:*



Pin	Signal
1	Transmit +
2	Transmit -
3	Receive +
4	-
5	-
6	Receive -
7	-
8	-

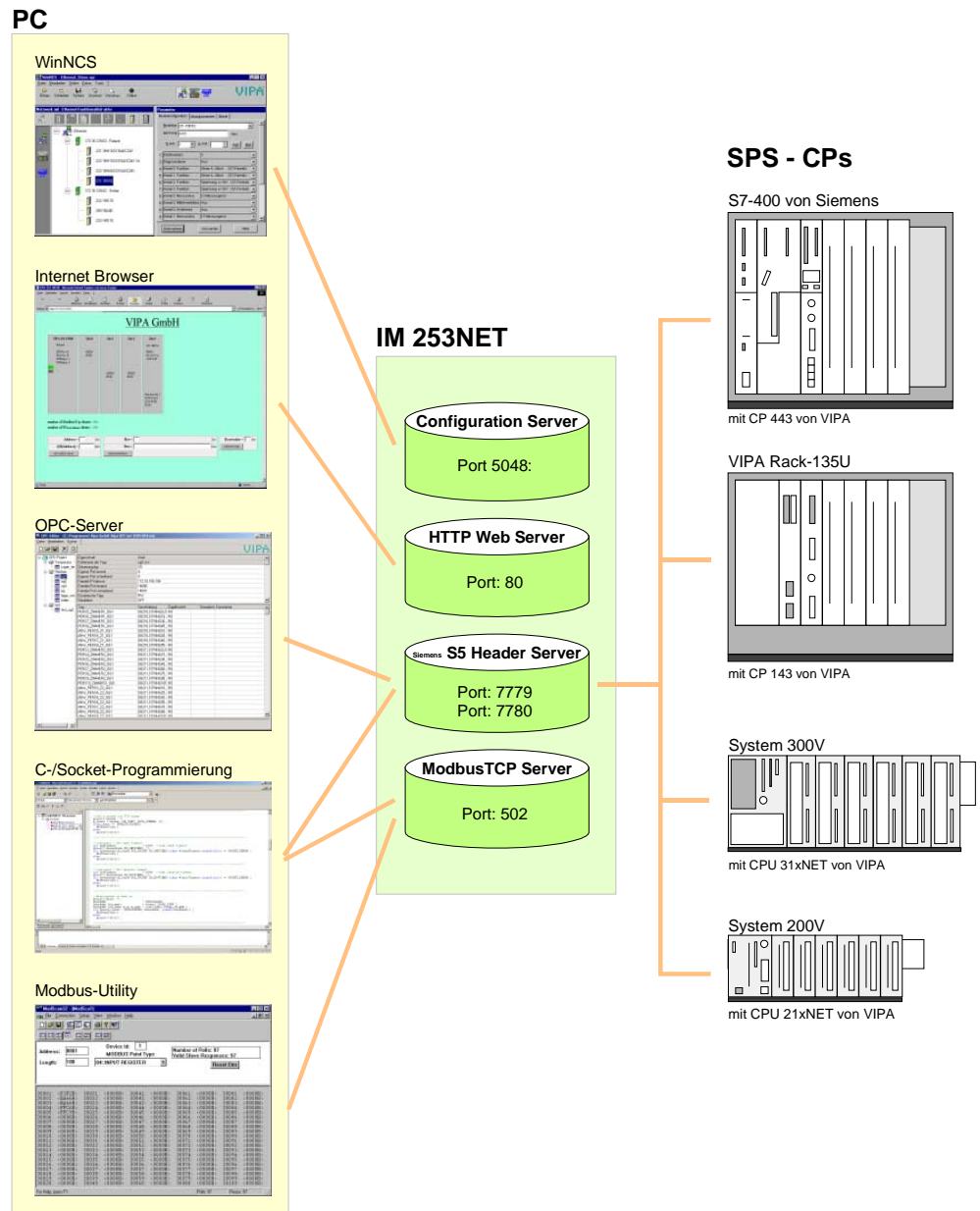
**Spannungsversorgung**

Der Ethernet-Koppler besitzt ein eingebautes Netzteil. Das Netzteil ist über die Front mit DC 24V (20,4 ... 28,8V) zu versorgen. Über die Versorgungsspannung werden neben der Buskopplerelektronik auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus mit maximal 3,5A versorgen kann. Das Netzteil ist gegen Verpolung und Überstrom geschützt. Ethernet und Rückwandbus sind galvanisch voneinander getrennt.

# Zugriffsmöglichkeiten auf den Ethernet-Koppler

## Übersicht

Die nachfolgende Abbildung zeigt die Möglichkeiten für den Zugriff auf den Ethernet-Koppler IM 253NET.



**Zugriff von  
PC-Seite***WinNCS zur Projektierung*

Der Zugriff erfolgt über Port 5048 auf Configuration Server.

Der Configuration Server ermittelt die Anzahl der gesteckten Module, deren Adress- und Parameterbereiche und stellt diese Informationen unter seiner IP-Adresse WinNCS zur Verfügung.

WinNCS sucht per Broadcast alle Koppler (Slaves) des Netzwerks. Hierbei reicht das zu durchsuchende Netzwerk bis zum Gateway.

Aus den gewonnenen Daten modelliert WinNCS ein symbolisches Netzwerk und stellt diese in seinem Netzwerk-Fenster dar.

Sie haben nun die Möglichkeit online dem symbolischen Netzwerk reelle Modultypen zuzuweisen und diese ggf. zu parametrieren.

Auch können Sie online dem Ethernet-Koppler eine IP-Adresse zuweisen und seine Firmware aktualisieren.

In WinNCS geben Sie auch die HTTP-Web-Server-Eigenschaften des Ethernet-Kopplers vor.

Alle ändernden Zugriffe erfolgen passwortgeschützt. Das Passwort wird pro Sitzung und Slave einmalig abgefragt.

**Im Auslieferungszustand ist das Passwort 00000000**

**Hinweis!**

Bevor Sie mit einem Internet-Browser auf den Ethernet-Slave zugreifen können, müssen Sie diesem eine in Ihr Firmennetz passende IP-Adresse zuweisen. Dies können Sie, wie oben erwähnt, online aus WinNCS durchführen.

*Internet Browser für Diagnose und Test*

Der Zugriff erfolgt über Port 80 auf HTTP Web Server.

Der HTTP-Server übermittelt eine dynamisch aufgebaute Web-Site, die die aktuelle Konfiguration des Ethernet-Kopplers darstellt.

Neben Firmwarestand, RDY/ERR-LED-Zustand werden hier auch die E/A-Zustände und Parameter der Module aufgelistet.

Die Web-Site bietet Ihnen auch die Möglichkeit, online Änderungen vorzunehmen, wie gezielt Ausgänge von Modulen ansteuern, deren Parameter zu ändern und einen Neustart (Reboot) des Ethernet-Kopplers auszuführen.

*OPC-Server für Datentransfer zwischen Koppler und PC*

Der Zugriff erfolgt über die Ports 7779 und 7780 auf den Siemens S5 Header Server. Über diese Ports werden Fetch- und Write-Zugriffe über den VIPA OPC-Server ermöglicht.

Mit dem OPC-Server haben Sie von VIPA ein komfortables Werkzeug für Visualisierung und Datentransfer.

*C-/Socketprogrammierung für Datentransfer zwischen Koppler und PC*

Der Zugriff erfolgt bei ModbusTCP über Port 502 auf den ModbusTCP Server und bei Siemens S5 Header über die Ports 7779 und 7780 auf den Siemens S5 Header Server.

Diese Möglichkeit des Datentransfers richtet sich an C-Programmierer, die mittels Socket-Programmierung eine offene Schnittstelle erstellen möchten.

Über einfache C-Programme ist es möglich, Daten zwischen PC und Ethernet-Koppler zu übertragen. Je nach Programmierung werden die Daten mit ModbusTCP oder mit Siemens S5 Header übertragen.

Näheres zur Programmierung mit Beispiel-Source finden Sie weiter unten in diesem Kapitel.

*Modbus-Utility*

Der Zugriff erfolgt über Port 502 auf den ModbusTCP-Server. Unter Modbus-Utility sind alle Tools und Programme zusammengefasst, die über eine ModbusTCP-Schnittstelle verfügen.

Beispielsweise finden Sie unter [www.win-tech.com](http://www.win-tech.com) das Demo-Tool "ModbusScan32" der Firma WinTech zum Download.

**Zugriff von  
SPS bzw. CP Seite***Datentransfer zwischen Koppler und CP mittels Siemens S5 Header*

Der Zugriff erfolgt über die Ports 7779 und 7780 auf den Siemens S5 Header Server. Über diese Ports werden dem VIPA-CP, OPC-Server oder Fremdgeräten Fetch- und Write-Zugriffe ermöglicht.

Für die Kommunikation ist in der CPU ein SPS-Programm erforderlich, das die Ein-/Ausgabe-Bereiche des CPs bedient. Im CP sind hierfür Fetch-/Write-Verbindungen zu projektieren.

## Prinzip der automatischen Adressierung

### Automatische Adressierung

Damit die gesteckten Peripheriemodule gezielt angesprochen werden können, müssen ihnen bestimmte Adressen im Ethernet-Koppler zugeordnet werden. Für Ein- und Ausgabe gibt es beim Ethernet-Koppler einen Adressbereich von je 256Byte.

Die Adressvergabe (auch Mapping genannt) erfolgt automatisch und kann nicht beeinflusst werden. Das Mapping können Sie sich über die Web-Site des Kopplers ausgeben lassen.

### Regeln

Beim Hochlauf vergibt der Ethernet-Koppler automatisch Adressen für seine Ein-/Ausgabe-Peripherie nach folgenden Regeln:

- Alle Module werden ab Adresse 0 von links (Ethernet-Koppler) nach rechts in aufsteigender Reihenfolge gemappt.
- Es wird zwischen Ein- und Ausgabe-Bereich unterschieden (hat beispielsweise ein Modul Ein- und Ausgabe-Daten, so können diese auf unterschiedlichen Adressen abgelegt werden).
- Eine Unterscheidung zwischen digitalen und analogen Daten findet nicht statt. Der Ethernet-Koppler generiert aus allen Modulen je einen zusammenhängenden Bereich für Ein- und Ausgabe-Daten.



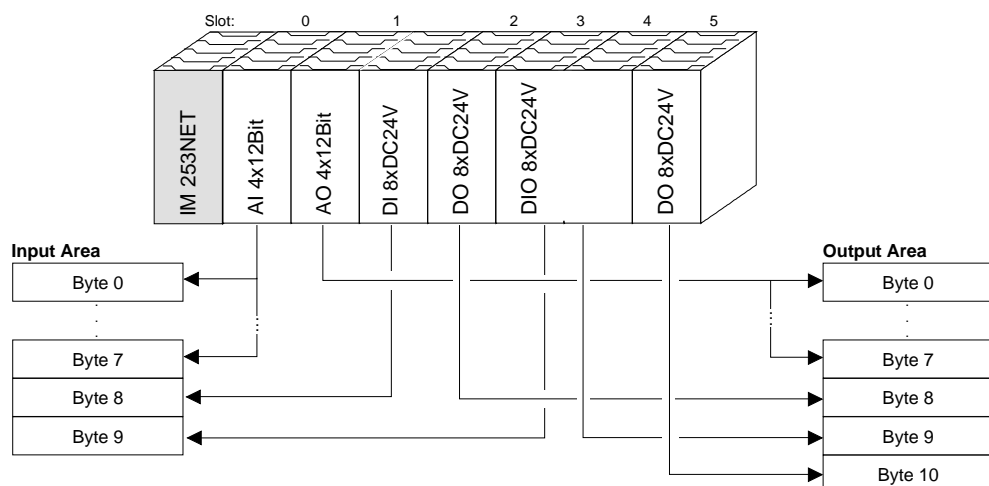
### Hinweis!

Ein Beschreibung der Ein- und Ausgabe-Bereiche, die ein Modul belegt, finden Sie in der entsprechenden Beschreibung zu dem Modul.

Bitte achten Sie darauf, dass Module, die mehr als 1 Byte belegen wie z.B. Analog-Module, ab einer geraden Adresse abgelegt werden. Ansonsten führt dies für ModbusTCP zu Problemen bei Wortzriffen.

### Beispiel zur automatischen Adresszuordnung

Die nachfolgende Abbildung soll die automatische Adresszuordnung nochmals verdeutlichen:





## Projektierung unter WinNCS

### Voraussetzung

Die Projektierung erfolgt unter WinNCS ab V3.09. Zur Projektierung sollten folgende Voraussetzungen erfüllt sein:

- Aktuelle VIPA2ETH.GSD liegt in WinNCS/GSD/Deutsch vor.  
Zur Projektierung der System 200V-Module in WinNCS bekommen Sie die Leistungsmerkmale der VIPA-Komponenten in Form einer GSD-Datei mitgeliefert.  
**Die GSD-Datei für den IM 253NET Ethernet-Koppler von VIPA lautet: VIPA2ETH.GSD**  
Kopieren Sie die GSD-Datei in WinNCS/GSD/Deutsch.  
Die aktuellste Version finden Sie unter <ftp.vipa.de/support>.
- Für die Online-Projektierung sollte der IM 253NET mit den zugehörigen Modulen aufgebaut, mit dem Ethernet verbunden und mit Spannung versorgt sein.



### Achtung!

Da jeder Ethernet-Slave mit der IP-Adresse 10.0.0.1 ausgeliefert wird, dürfen sich bei der Erstinbetriebnahme nicht mehrere neue Ethernet-Slaves im Netz befinden!

### Vorgehensweise bei der Online-Projektierung

- Starten Sie WinNCS und legen Sie mit **Datei > Projekt anlegen/öffnen** ein neues "Ethernet"-Projekt an.  
→ Es öffnet sich ein Parameterfenster zur Online-Suche von "Slaves" und "Stationen". [Slaves] listet alle Ethernet-Koppler und [Stationen] alle CPs auf.
- Klicken Sie auf [Slaves]  
→ Es werden alle Ethernet-Koppler gesucht und mit IP-Adresse und ggf. mit symbolischem Namen aufgelistet.
- Durch Doppelklick auf einen gelisteten Slave wird dieser in das Netzwerkfenster übertragen und mit seiner E/A-Peripherie aufgelistet.  
→ Sofern noch keine Parametrierung vorliegt, werden die Module symbolisch (ohne Bezeichnung) aufgelistet.
- Ordnen Sie nun im Parameterfenster dem aufgelisteten Modul-Symbol den entsprechenden Modultyp zu und stellen Sie ggf. Parameter ein. Der entsprechende Adressbereich, den ein Modul im TCP-Datenstrom belegt, wird automatisch vom Ethernet-Koppler vorgegeben.
- Sobald Sie auf [übernehmen] klicken, erfolgt eine Passwortabfrage. Die Passwortabfrage findet einmal pro Sitzung und Koppler statt. Geben Sie das entsprechende Passwort an. Im Auslieferungszustand ist das Passwort 00000000. Ist das Passwort richtig, werden die Daten online an den Ethernet-Koppler übertragen. Verfahren Sie auf diese Weise mit allen Modulen, die aufgelistet sind.
- Speichern Sie Ihr Projekt

## Diagnose und Test mittels Internet Browser

### Adressierung

Tragen Sie in Ihrem Internet Browser die projektierte IP-Adresse Ihres Ethernet-Kopplers ein. Schon haben Sie Zugriff auf eine dynamisch aufgebaute Web-Site, die der integrierte HTTP-Server liefert.

Bitte beachten Sie, dass die Web-Site immer die Informationen zum Zeitpunkt der letzten Aktualisierung beinhaltet.

Zur Aktualisierung klicken Sie auf home unten links auf der Web-Site.

### Aufbau der Web-Site

Die Web-Site ist dynamisch aufgebaut und richtet sich nach der Anzahl der am Ethernet-Koppler befindlichen Module. Die Zugriffsrechte auf diese Web-Site sind über WinNCS frei konfigurierbar

Folgende Elemente befinden sich auf der Web-Site:

- Diagnose Ethernet-Koppler
- Diagnose Ein-/Ausgabe-Peripherie
- Informationen über angebundene Clients
- Elemente für den aktiven Zugriff auf den Ethernet-Koppler

Diagnose Ethernet-Koppler	Diagnose Ein-/Ausgabe-Peripherie				
VIPA 253-1NE00	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4
Station A	221-1BH10	222-1BH10	221-1BH10	223-2BL10	231-1BD52
HWVer: 10 PLDVer: 10	IB[0]= 00 00		IB[2]= 00 00	IB[4]= 00 00	IB[6]= 00 00 00 00 00 00 00 00
FWMajor: 1 FWMinor: 3		QB[0]= 00 00		QB[2]= 00 00	
<input type="checkbox"/> RDY <input type="checkbox"/> ERR					Prm(len10)= 00 00 2d 2d 28 28 00 00 00 00

#### Informationen über angebundene Clients

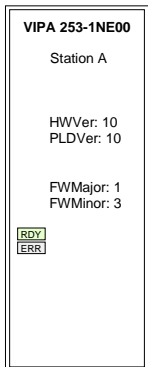
**Number of Modbus/TCP clients<2>:** [172.16.131.31] [172.16.131.55]  
**Number of S5 from Siemens clients:** <1>: [172.16.131.10]

#### Elemente für den aktiven Zugriff auf den Ethernet-Koppler

Address = <input type="text"/> dec	Slot = <input type="text"/> dec	Resetvalue = <input type="text"/> dec
QB[Address] = <input type="text"/> hex	Prm = <input type="text"/> hex	<input type="button" value="Reboot node"/>
<input type="button" value="Set output value"/>	<input type="button" value="Set parameters"/>	

home

## Diagnose Ethernet-Koppler



In diesem Bereich werden alle Informationen zum Ethernet-Koppler dargestellt wie symbolischer Name, Versionsstände und Zustandsanzeigen der LEDs.

**Symbolischer Name:** Mittels WinNCS können Sie neben einer IP-Adresse auch einen symbolischen Namen für Ihren Ethernet-Koppler vergeben, der hier angezeigt wird.

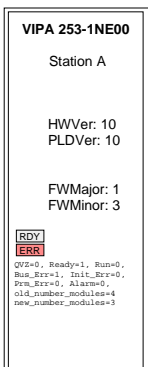
**HWVer:** Hier wird die Version der Hardware (Elektronik) festgehalten. Den HW-Ausgabestand (nur die Vorkommastelle) finden Sie auch als Kennzeichnung auf der Frontseite des Moduls.

**PLDVer:** Das PLD (**P**rogrammable **L**ogic **D**evice) ist ein programmierbarer Logik-Baustein, der die Kommunikation zwischen Rückwandbus und Prozessor steuert.

**FWMajor, FWMinor:** Die Firmwareversion ist geteilt in *FWMajor* (Hauptversion) und *FWMinor* (Unterversion). In einer Unterversion sind kleinere Änderungen durchgeführt worden. Sobald aber grundlegende Änderungen durchgeführt werden, erhöht sich auch die Hauptversionsnummer.

**RDY, ERR:** Zustandsanzeige der LEDs RD und ER

## Fehleranzeige

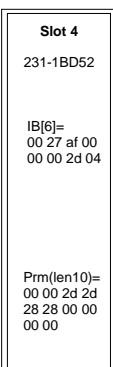


Solange der Ethernet-Koppler fehlerfrei kommuniziert, bleibt die Zustandsanzeige wie oben gezeigt. Im Fehlerfall erscheint unterhalb von ERR beispielsweise folgende Meldung:

```
QVZ=0 Ready=1, Run=0, Bus_Err=1, Init_Err=0, Prm_Err=0, Alarm=0
old_number_modules=4, new_number_modules=3
```

Diese Meldung zeigt an, dass ein Modul ausgefallen ist

## Bereich Module Slot 0 ... 31



In diesem Bereich werden alle Informationen zur Ein-/Ausgabe-Peripherie dargestellt wie Modulname, Ein-/Ausgabe-Belegung und Parameterbytes.

**Modulname:** Als Modulname dient die Bestell-Nr. des Moduls. Hierüber ist das Modul eindeutig identifizierbar.

**Ein-/Ausgabe-Belegung:** Hier werden 4 Informationen dargestellt:

- Art: Eingabe-Bereich (IB), Ausgabe-Bereich (QB)
- Die Anfangs-Adresse des Bereichs steht in Klammern
- Es wird genau die Anzahl der Bytes dargestellt, die das Modul belegt
- Die Inhalte der Bytes entsprechen denen des Ethernet-Kopplers zum Zeitpunkt der letzten Aktualisierung der Web-Site

Beispiel: Slot 4

Dies bedeutet: Das Modul auf Slot 4 belegt im Eingangsbereich ab Byte 6 8Byte mit hexadezimalen Inhalt.

```
IB[6]=
00 00 00 00
00 00 00 00
```

Das Image wird im little endian (Intel) Format ausgegeben (Low-Byte, High-Byte).

*Parameterbytes:* Die mit `PRM()` = Parameterbytes beinhalten folgende Informationen:

- Die Länge des Parameterblocks steht in Klammern mit einem vorangestellten `len`.
- Die Inhalte der Bytes zeigen die Parameterbytes des entsprechenden Moduls.

**Informationen über angebundene Clients**

In diesem Bereich erhalten Sie Informationen über Anzahl und IP-Adresse der Clients, die zurzeit mit dem Ethernet-Koppler über ModbusTCP bzw. Siemens S5 Header Protokoll kommunizieren. Es können je Protokoll maximal 8 Clients gleichzeitig mit dem Ethernet-Slave kommunizieren. Die Anzahl steht in `<2>`: gefolgt von der IP-Adresse in `[]`.

Beispiel:

**Number of ModbusTCP clients:** `<2>`: `[172.16.131.20] [172.16.140.63]`

(Es kommunizieren zurzeit 2 Clients unter ModbusTCP mit den IP-Adressen 172.16.131.20 und 172.16.140.63).

**Elemente für den aktiven Zugriff ...**

Während die oben aufgeführten Elemente der Informationsanzeige dienen, haben Sie mit den hier aufgeführten Elementen für den aktiven Zugriff die Möglichkeit, den Ethernet-Koppler und seine Module online anzusprechen.

Folgende 3 Steuerelemente stehen zur Verfügung:

- Ausgänge steuern
- Modul parametrieren
- Reset des Ethernet-Kopplers ausführen

Adress =	<input type="text" value=""/>	dec
QB[Adress] =	<input type="text" value=""/>	hex
Set output value		

*Ausgänge steuern*

Mit diesem Steuerelementen können Sie einen gewünschte Ausgabe-adressbereich mit Werten belegen und diese über `[Set output value]` an den Ethernet-Koppler übertragen.

Bitte beachten Sie, dass die Adresse als Dezimalzahl und der Wert als Hex-Wert vorzugeben ist. Sie können maximal 4Byte an die mit `Adress` vorgegebene Adresse übertragen.

Bitte beachten Sie, dass die Bytes immer mit führender Null übertragen werden. Leerzeichen dienen als Byte-Trennzeichen.

Beispiel:

Address=0		
QB[Address]= 12	→	QB[0]= 12 00
QB[Address]= 1 2	→	QB[0]= 01 02
QB[Address]= 1234	→	QB[0]= 12 34
QB[Address]= 123	→	QB[0]= 01 23

Slot =	<input type="text"/>	dec
Prm =	<input type="text"/>	hex
Set parameters		

**Modul parametrieren**

Über dieses Steuerelement können Module online mit Parametern versorgt werden, indem Sie unter Prm die Parameter-Bytes eintragen und über Slot einen Steckplatz vorgeben.

Mit [Set parameters] werden die Parameter an das entsprechende Modul übertragen.

Bitte beachten Sie, dass die Slot-Nr. als Dezimalzahl und die Parameter als Hex-Wert einzugeben sind.

Bytes werden immer mit führender Null übertragen. Als Trennzeichen muss ein Leerzeichen eingegeben werden.

**Hinweis!**

Übertragen Sie immer die vollständige Anzahl der Parameter-Bytes an ein Modul, da dies ansonsten zu Fehlern im Modul führen kann.

Die Anzahl der Parameter und deren Belegung finden Sie in der zugehörigen Beschreibung der entsprechenden Module.

Resetvalue =	<input type="text"/>	dec
Reboot node		

**Reset des Ethernet-Kopplers ausführen**

Über [Reboot node] wird ein Reset des Ethernet-Kopplers ausgelöst. Nach einem Reboot ist die Web-Site über home zu aktualisieren.

Durch Vorgabe eines *Resetvalues* können Sie zusätzlich zum Reboot des Ethernet-Kopplers die Konfiguration oder Modulparameter löschen.

Zulässige *Resetvalue*-Werte sind nur 1, 2 oder 3. Andere Werte werden ignoriert!

Resetvalue= 1 Reboot des Kopplers (Defaulteinstellung)

Resetvalue= 2 Löschen der Modul-Konfiguration (Modulnamen) und Reboot des Kopplers

Resetvalue= 3 Löschen der Modul-Parameter und Reboot des Kopplers

## ModbusTCP

**Allgemeines** ModbusTCP ist ein auf TCP/IP aufgesetztes Modbus-Protokoll, wobei die IP-Adresse der Adressierung dient. Das ModbusTCP erlaubt eine Client-Server-Kommunikation, wobei mehrere Clients von einem Server bedient werden können.

**Telegramm-Aufbau inkl. TCP/IP** Die Anforderungs-Telegramme, die ein Master sendet und die Antwort-Telegramme eines Slaves haben den gleichen Aufbau:

ModbusTCP	Slave-Adresse	Funktions-Code	Daten
6Byte-Header mit Anzahl der nachfolgenden Bytes	1Byte-Daten	1Byte-Daten	max 254Byte

**ModbusTCP-Header (6Byte)** Für Sende- und Empfangstelegramm verwendet ModbusTCP einen 6Byte großen Header, der folgenden Aufbau hat:

*ModbusTCP-Header*

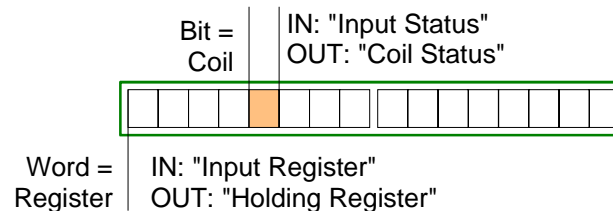
Byte	Name	Beschreibung
0	Transaction identifier (High-Byte)	wird von Server zurückgesendet (beliebig)
1	Transaction identifier (Low-Byte)	wird von Server zurückgesendet (beliebig)
2	Protocol identifier (High-Byte)	immer 0
3	Protocol identifier (Low-Byte)	immer 0
4	Length field (High-Byte)	immer 0 da Nachrichten kleiner 256Byte
5	Length field (Low-Byte)	Anzahl der nachfolgenden Bytes

In der Regel haben Byte 0 ... 4 den Wert 0. Sie können aber auch Byte 0 und 1 im Slave Hochzählen lassen und somit ein zusätzliche Kontrollinstanz einfügen.

## Modbus-Funktionscodes

### Namenskonventionen

Für Modbus gibt es Namenskonventionen, die hier kurz aufgeführt sind:



- Modbus unterscheidet zwischen Bit- und Wortzugriff; Bits = "Coils" und Worte = "Register".
- Bit-Eingänge werden als "Input-Status" bezeichnet und Bit-Ausgänge als "Coil-Status".
- Wort-Eingänge werden als "Input-Register" und Wort-Ausgänge als "Holding-Register" bezeichnet.

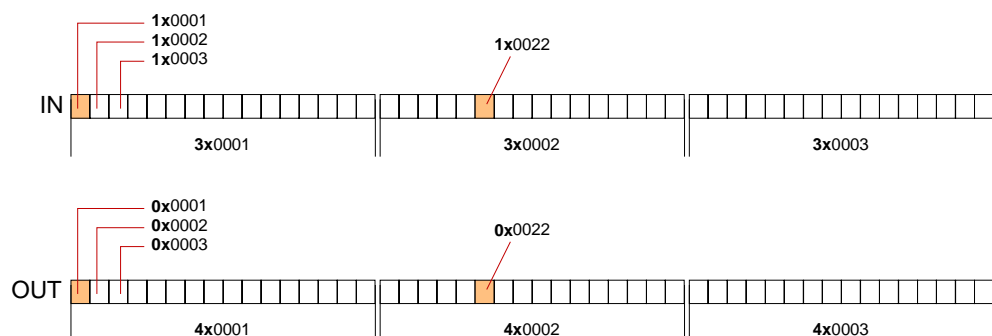
### Bereichsdefinitionen

Üblicherweise erfolgt unter Modbus der Zugriff mittels der Bereiche 0x, 1x, 3x und 4x.

Mit 0x und 1x haben Sie Zugriff auf *digitale* Bit-Bereiche und mit 3x und 4x auf *analoge* Wort-Bereiche.

Da aber beim Ethernet-Koppler von VIPA keine Unterscheidung zwischen Digital- und Analogdaten stattfindet, gilt folgende Zuordnung:

- 0x: Bit-Bereich für Ausgabe  
Zugriff über Funktions-Code 01h, 05h, 0Fh
- 1x: Bit-Bereich für Eingabe  
Zugriff über Funktions-Code 02h
- 3x: Wortbereich-Bereich für Eingabe  
Zugriff über Funktions-Code 04h
- 4x: Wortbereich-Bereich für Ausgabe  
Zugriff über Funktions-Code 03h, 06h, 10h



Eine Beschreibung der Funktions-Codes finden Sie auf den Folgeseiten.

**Übersicht**

Folgende Modbus-Funktionscodes sind implementiert:

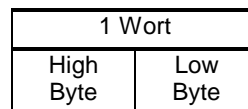
Code	Befehl	Beschreibung
01h	Read n Bits	n Bit lesen von Ausgabebereich 0x
02h	Read n Bits	n Bit lesen von Eingabebereich 1x
03h	Read n Words	n Worte lesen von Ausgabebereich 4x
04h	Read n Words	n Worte lesen von Eingabebereich 3x
05h	Write one Bit	1 Bit schreiben in Ausgabebereich 0x
06h	Write one Word	1 Wort schreiben in Ausgabebereich 4x
0Fh	Write n Bits	n Bit schreiben in Ausgabebereich 0x
10h	Write n Words	n Worte schreiben in Ausgabebereich 4x

Beim Ethernet-Koppler von VIPA wird zwischen digitalen und analogen Daten nicht unterschieden!



**Hinweis!**

Für die Byte-Reihenfolge im Wort gilt immer:



**Antwort des Kopplers**

Liefert der Slave einen Fehler zurück, so wird der Funktionscode mit 80h "verodert" zurückgesendet. Ist kein Fehler aufgetreten, wird der Funktionscode zurückgeliefert.

Koppler-Antwort: Funktionscode OR 80h → Fehler  
 Funktionscode → OK

**Read n Bits  
01h, 02h**

Die Funktion ermöglicht das bitweise Lesen aus einem Slave.

**Kommandotelegramm**

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits
x x 0 0 0 6				
6Byte	1Byte	1Byte	1Wort	1Wort

**Antworttelegramm**

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Anzahl der gelesenen Bytes	Daten 1. Byte	Daten 2. Byte	...
x x 0 0 0						
6Byte	1Byte	1Byte	1Byte	1Byte	1Byte	
					max. 252Byte	

max. 255Byte



**Read n Words  
03h, 04h**

Diese Funktion ermöglicht das wortweise Lesen aus einem Koppler.

Kommandotelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse Bit	Anzahl der Worte
x   x   0   0   0   6				
6Byte	1Byte	1Byte	1Wort	1Wort

Antworttelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Anzahl der gelesenen Bytes	Daten 1. Wort	Daten 2. Wort	...
x   x   0   0   0   6						
6Byte	1Byte	1Byte	1Byte	1Wort	1Wort max. 125Worte	

max. 253Byte

**Write a Bit  
05h**

Mit dieser Funktion können Sie ein Bit in Ihrem Koppler ändern. Eine Zustandsänderung erfolgt unter "Zustand Bit" mit folgenden Werten:

"Zustand Bit" = 0000h → Bit = 0, "Zustand Bit" = FF00h → Bit = 1

Kommandotelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse Bit	Zustand Bit
x   x   0   0   0   6				
6Byte	1Byte	1Byte	1Wort	1Wort

Antworttelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse Bit	Zustand Bit
x   x   0   0   0   6				
6Byte	1Byte	1Byte	1Wort	1Wort

**Write a word  
06h**

Diese Funktion schickt ein Wort an den Koppler. Hiermit können Sie im Koppler ein Register überschreiben.

Kommandotelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse Wort	Wert Wort
x   x   0   0   0   6				
6Byte	1Byte	1Byte	1Wort	1Wort

Antworttelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse Wort	Wert Wort
x   x   0   0   0   6				
6Byte	1Byte	1Byte	1Wort	1Wort

**Write n Bits 0Fh**

Diese Funktion schreibt n Bits an den Slave. Bitte beachten Sie, dass die Anzahl der Bits zusätzlich in Byte anzugeben sind.

Kommandotelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits	Anzahl der Bytes	Daten 1. Byte	Daten 2. Byte	...
x   x   0   0   0   0								
	1Byte max. 255Byte	1Byte	1Wort	1Wort	1Byte	1Byte	1Byte max. 248Byte	1Byte

Antworttelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits
x   x   0   0   0   6				
	1Byte	1Byte	1Wort	1Wort

**Write n Words 10h** Über diese Funktion können Sie n Worte an den Slave schicken.

Kommandotelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl der Worte	Anzahl der Bytes	Daten 1. Wort	Daten 2. Wort	...
x   x   0   0   0   0								
	1Byte max. 255Byte	1Byte	1Wort	1Wort	1Byte	1Wort	1Wort max. 124Worte	1Wort

Antworttelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl der Worte
x   x   0   0   0   6				
	1Byte	1Byte	1Wort	1Wort

## Siemens S5 Header Protokoll

### Allgemeines

Das Siemens S5 Header Protokoll dient zum Datenaustausch zwischen SPS-Systemen. Unter Einsatz des Organisationsformats (kurz ORG), das in das Siemens S5 Header Protokoll eingebettet ist, ist die Kurzbeschreibung einer Datenquelle bzw. eines Datenziels in SPS-Umgebung möglich.

### ORG-Formate

Die verwendbaren ORG-Formate entsprechen den Siemens-Vorgaben und sind in der nachfolgenden Tabelle aufgelistet.

Der ORG-Block ist bei READ und WRITE optional.

Die ERW-Kennung ist bei Einsatz mit dem Ethernet-Koppler irrelevant.

Die Anfangsadresse und Anzahl adressieren den Speicherbereich und sind im HIGH-/LOW- Format abgelegt (Motorola - Adressformat)

Beschreibung	Typ	Bereich
ORG-Kennung	BYTE	1..x
ERW-Kennung	BYTE	irrelevant
Anfangsadresse	HILOWORD	0..y
Anzahl	HILOWORD	1..z

In der nachfolgenden Tabelle sind die verwendbaren ORG-Formate aufgelistet. Die "Länge" darf nicht mit -1 (FFFFh) angegeben werden.

### ORG-Kennung 02h-05h

CPU-Bereich	MB	EB	AB	PB
ORG-Kennung	02h	03h	04h	05h
Beschreibung	Hier ist nur zulässig: Lesen MB0 mit Länge 4.  Die Gesamtlänge der Bereiche für Ein- und Ausgabe wird ermittelt und in MB0 ... MB3 nach	Quell-/Zieldaten aus/in Prozessabbild der Eingänge (PAE).	Quell-/Zieldaten aus/in Prozessabbild der Ausgänge (PAA).	Quell-/Zieldaten aus/in Peripheriemodul. Bei Quelldaten Eingabemodule, bei Zieldaten Ausgabemodule.
DBNR	folgender Form abgelegt:	irrelevant	irrelevant	irrelevant
Anfangsadresse Bedeutung	MB0: Länge In-Bereich MB1: 00 MB2: Länge Out-Bereich MB3: 00	EB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.	AB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.	PB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.
erlaubter Bereich:		0...255	0...255	0... 255
Anzahl Bedeutung		Länge des Quell-/Zieldatenblocks in Bytes.	Länge des Quell-/Zieldatenblocks in Bytes.	Länge des Quell-/Zieldatenblocks in Bytes.
erlaubter Bereich:		1...256	1...256	1...256

**Aufbau SPS-Header**

READ und WRITE generiert der Ethernet-Koppler Header für Anforderungs- und Quittungstelegramme. Diese Header sind in der Regel 16Byte lang und haben folgende Struktur:

**bei WRITE**

**Client (SPS, PC)**

Anforderungstelegramm

Systemkennung	= "S"
	= "5"
Länge.Header	=16d
Kenn.OP-Code	=01
Länge OP-Code	=03
<b>OP-Code</b>	<b>=03</b>
ORG-Block	=03
Länge ORG-Block	=08
ORG-Kennung	
DBNR	
Anfangsadresse	H
	L
Länge	H
	L
Leerblock	=FFh
Länge Leerbl.	=02
Daten bis zu 64K jedoch nur wenn Fehler-Nr.=0	

**Server (Ethernet-Slave)**

Quittungstelegramm

Systemkennung	= "S"
	= "5"
Länge.Header	=16d
Kenn.OP-Code	=01
Länge OP-Code	=03
<b>OP-Code</b>	<b>=04</b>
Quittungsblock	=0Fh
Länge Q-Block	=03
Fehler Nr.	=Nr.
Leerblock	=FFh
Länge Leerblock	=07
	frei

**bei READ**

Anforderungstelegramm

Systemkennung	= "S"
	= "5"
Länge.Header	=16d
Kenn.OP-Code	=01
Länge OP-Code	=03
<b>OP-Code</b>	<b>=05</b>
ORG-Block	=03
Länge ORG-Block	=08
ORG-Kennung	
DBNR	
Anfangsadresse	H
	L
Länge	H
	L
Leerblock	=FFh
Länge Leerbl.	=02

Quittungstelegramm

Systemkennung	= "S"
	= "5"
Länge.Header	=16d
Kenn.OP-Code	=01
Länge OP-Code	=03
<b>OP-Code</b>	<b>=06</b>
Quittungsblock	=0Fh
Länge Q-Block	=03
Fehler Nr.	=Nr.
Leerblock	=FFh
Länge Leerblock	=07
	frei
Daten bis zu 64K jedoch nur wenn Fehler-Nr.=0	

**Mögliche Fehler-Nummern**

Folgende Fehlernummern kann das Quittungstelegramm enthalten:

- 0: kein Fehler
- 3: Adresse liegt außerhalb des definierten Bereichs
- 6: Kein gültiges ORG-Format (Angabe Datenquelle/-Ziel ist fehlerhaft).  
Nur erlaubt: EB, AB, PB und MB

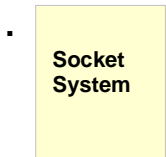
# Programmierbeispiel

## Schritte der Programmierung

Für den Einsatz des Ethernet-Kopplers an einem PC sollten Sie fundierte C-Programmiererfahrung besitzen, insbesondere im Bereich der Socket-Programmierung. In diesem Abschnitt soll Ihnen lediglich eine kurze Übersicht zur Programmierung gegeben werden.

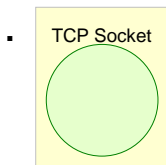
**PC**  
IP: 172.16.192.50

**Slave**  
IP: 172.16.192.11



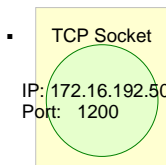
**zu 1.** Microsoft Socket System starten

```
WSAStartup (wVersionRequested, &wsaData);
```



**zu 2.** Socket-Ressourcen für TCP reservieren

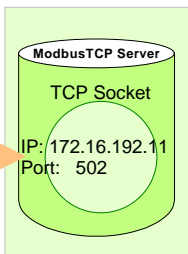
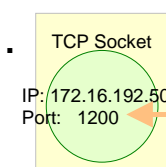
```
m_lsock = socket (AF_INET, SOCK_STREAM, 0);
```



**zu 3.** Socket an den lokalen PC anbinden

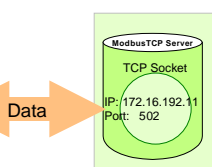
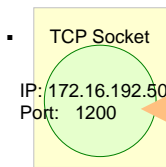
```
SocketAddr.sin_port = htons( 0 );
SocketAddr.sin_addr.S_un.S_addr = inet_addr( "0.0.0.0" );
bind(m_lsock, (LPSOCKADDR) &SocketAddr, sizeof(SocketAddr));
```

Bei Aufruf von bind mit den Werten 0 für Port und IP-Adresse, wird dem Socket die PC-IP-Adresse und der nächste freie Port zugewiesen.  
(hier: IP: 172.16.192.50, Port: 1200)



**zu 4.** Verbindung zu externem Gerät aufbauen

```
SocketAddr.sin_port = htons (m_wPort);
SocketAddr.sin_addr.S_un.S_addr = inet_addr(m_szIpAddress);
connect(m_lsock, (LPSOCKADDR) &SocketAddr, sizeof(SocketAddr));
```



**zu 5.** Für schreibenden bzw. lesenden Zugriff sind je nach Protokoll entsprechende Telegramme aufzubauen und in sndBuf abzulegen.  
sndBufLen beinhaltet die Anzahl der zu sendenden Bytes.

### Lesender Zugriff

sndBuf senden (Request)

```
send(m_lsock, (char *)sndBuf, sndBufLen, 0);
```

Telegramm in rcvBuf empfangen (Response+Daten)

```
recv(m_lsock, (char *)rcvBuf, sizeof(rcvBuf), 0);
```

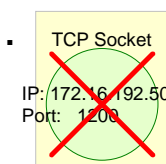
### Schreibender Zugriff

sndBuf senden (Request+Daten)

```
send(m_lsock, (char *)sndBuf, sndBufLen, 0);
```

Telegramm in rcvBuf empfangen (Response)

```
recv(m_lsock, (char *)rcvBuf, sizeof(rcvBuf), 0);
```



**zu 6.** Socket wieder schließen

```
closesocket(m_lsock);
```

Ein einfaches Programmierbeispiel finden Sie auf [ftp.vipa.de/support](http://ftp.vipa.de/support) unter Demo Client: Cx000059.

## Technische Daten

### IM 253NET

Elektrische Daten	VIPA 253-1NE00
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme	120mA
Ausgangsstrom Rückwandbus	max. 3,5A
Potenzialtrennung	≥ AC 500V
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse/Schnittstellen	RJ45 für Twisted-Pair-Ethernet
Ethernet Schnittstelle	
Ankopplung	RJ45
Netzwerk Topologie	Sterntopologie
Medium	Twisted Pair
Übertragungsrate	10/100MBit
Gesamtlänge	max. 100m pro Segment
Online-Zugriff	
Test-/Diagnose	HTTP-Server integriert, der über seine Web-Site die Konfiguration grafisch darstellt und für Tests über Parametrier- und Projektiermöglichkeiten verfügt.
Projektierung	über WinNCS mit online Koppler-Suche und Projektierung
Kombination mit Peripheriemodulen	
max. Anzahl Clients	8 je ModbusTCP bzw. Siemens S5 Protokoll
max. Anzahl Eingangs-Byte	256
max. Anzahl Ausgangs-Byte	256
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	70g

## Teil 8 PC 288 - CPU

### Überblick

In diesem Kapitel ist der Einsatz des PC 288 im System 200V beschrieben. Nach einer kurzen Einführung und Systemvorstellung wird die Projektierung eines PC-basierenden Systems gezeigt.

Ein Kommunikationsbeispiel beendet das Kapitel.

Nachfolgend sind beschrieben:

- Systemübersicht
- Grundlagen
- Aufbau
- Projektierung
- Projektierbeispiel
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 8 PC 288 - CPU</b> .....	<b>8-1</b>
Systemübersicht .....	8-2
Grundlagen.....	8-3
Eigenschaften.....	8-4
PC 288 - CPU - Aufbau .....	8-4
Komponenten .....	8-5
Einsatz der Speichermedien .....	8-9
Einsatz im System 200V .....	8-10
BIOS-Setup Bedienung.....	8-13
Registerbeschreibung.....	8-21
Technische Daten.....	8-23

## Systemübersicht

### PC 288



Der PC 288 ist ein vollwertiger 486DX-PC. Hiermit lassen sich zentrale und dezentrale Steuerungsaufgaben lösen.

Zur externen Speicherung kommen CompactFlash-Karten oder Festplatten (IBM Mikrodrive) mit einer Speicherkapazität von bis zu 1GByte zum Einsatz.

### Bestelldaten PC 288

Typ	Bestellnummer	Beschreibung
PC 288 - CPU	VIPA 288-2BL10	486 PC-LAN; 66 MHz
CompactFlash	VIPA 950-1KS00	CompactFlash Typ II
HDD	VIPA 950-1KH00	HDD 340MByte, 540MByte oder 1GByte IBM Mikrodrive



## Grundlagen

### Allgemein

Mit dem PC 288 haben Sie in kompakter Bauform einen vollwertigen PC-AT mit den Leistungen eines 486DX-Prozessors. Auf dem internen 8MByte Flash-ROM ist das Betriebssystem MS-DOS 6.22 bereits vorinstalliert.

Neben Anschlüssen für Maus, Tastatur, Monitor bzw. TFT-Displays besitzt der PC 288 eine RJ45-Buchse zur Netzwerkanbindung.

Für die externe Speicherung kommt eine CompactFlash-Karte (Typ II) zum Einsatz, die direkt an der Front gesteckt werden. Hier können Sie CompactFlash-Karten bzw. Festplatten IBM Mikrodrive mit einer Speicherkapazität von bis zu 1GByte stecken.

### Einsatzbereiche

Der PC ist System 200V-konform und kann als Master an die System 200V-Peripherie gekoppelt werden. Hiermit können Sie Maschinen- und Anlagenstrukturen im stand-alone Betrieb oder im Profibus-Netzwerk realisieren.

### Projektierung

Steuerungsaufgaben und einfache grafische Darstellungen können in den Hochsprachen C und C++ programmiert werden.

Die als Source-Code offengelegte Anwenderprogrammierschnittstelle `vbus_api` ermöglicht auf einfache Art die Erstellung von Steuerungsprogrammen.

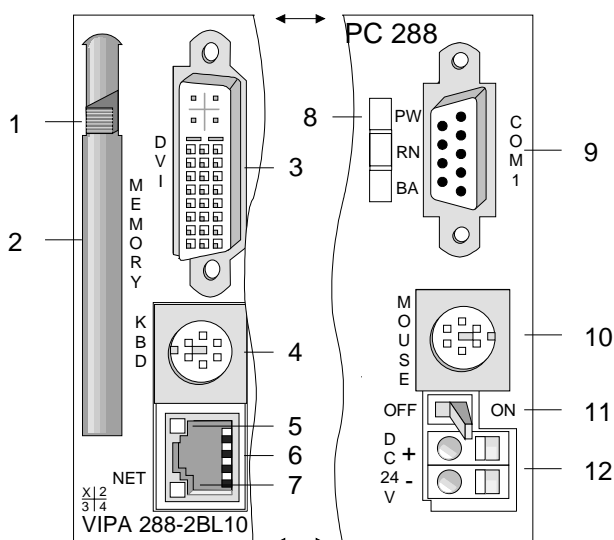
Hiermit können Sie auch Treiber für verschiedenste Betriebssysteme programmieren.

## Eigenschaften

- PC-AT-kompatibel
- STPC INDUSTRIAL 66MHz
- Hauptspeicher 32MB
- 8MB DiskOnChip<sup>®</sup>, bootfähig
- TYP II-Steckplatz für CompactFlash<sup>™</sup> Speicherkarte
- Serielle Schnittstelle COM1
- Anschluss für AT-Tastatur und dazu kompatible Tastaturen (Folientastaturen etc.) über Mini-DIN-Buchse
- Anschluss für Maus über Mini-DIN-Buchse
- Stromversorgung aus 24V-Netz
- integrierter V-Bus-Kontroller zur Steuerung der Module aus der System 200V-Serie
- integrierter Watchdog-Timer
- DVI-Schnittstelle (Digital Visual Interface)  
Anschluss TFT - LCD über PANEL LINK<sup>®</sup>

## PC 288 - CPU - Aufbau

### Frontansicht PC 288



- [1] Auswurf-Taste für Compact Flash<sup>®</sup>
- [2] Steckplatz für Compact Flash
- [3] DVI-Schnittstelle für Bildschirm/TFT
- [4] PS2-Buchse für Tastatur
- [5] LED Aktivitäts-Anzeige Ethernet
- [6] RJ45-Buchse für Ethernet
- [7] LED Geschwindigkeit 100MBit
- [8] LEDs Statusanzeige
- [9] 9polige serielle Schnittstelle
- [10] PS2-Buchse für Maus
- [11] ON/OFF-Schalter für Spannungsversorgung
- [12] Anschluss für DC 24V Spannungsversorgung

## Komponenten

### LEDs

Der PC 288 besitzt 3 LEDs, die der Statusanzeige dienen. Beim Einschalten der Stromversorgung leuchten alle 3 LEDs.

Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
PW	Gelb	Signalisiert den eingeschalteten PC. PC-Elektronik und Rückwandbus (V-Bus) werden versorgt.
RN	Grün	Leuchtet, wenn der PC sich im Software-RUN befindet und die V-Bus-Kommunikation läuft. Bei einem Fehler am V-Bus brennt die LED nicht.
BA	Rot	Leuchtet, wenn die Befehlsausgabesperre (BASP) aktiviert ist, d.h. die Ausgabebaugruppen werden nicht freigegeben.

### Schalter ON/OFF

Über den ON/OFF-Schalter schalten Sie die Spannungsversorgung für die PC-Elektronik und für den Rückwandbus.

### Spannungsversorgung

Der PC besitzt ein eingebautes Netzteil. Der Anschluss erfolgt über zwei Anschlussklemmen an der Frontseite. Mit dem ON/OFF-Schalter schalten Sie das Netzteil. In Stellung OFF ist die Versorgung des Rückwandbusses und der PC-Elektronik unterbrochen.

Das Netzteil ist mit DC 24V zu versorgen. Über die Versorgungsspannung werden neben der PC-Elektronik auch die angeschlossenen Module über den Rückwandbus mit maximal 3,5A versorgt.



#### Hinweis!

Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung!

### Steckplatz für Compact Flash

Auf diesem Steckplatz können Sie eine CompactFlash<sup>®</sup>-Speicherkarte des Typs II einsetzen. Der PC bindet die Karte in Form eines Laufwerks in das System ein.

Mit einem CompactFlash<sup>®</sup>-Adapter wird die Speicherkarte kompatibel zum "großen" PCMCIA Typ II Format. Somit können Sie Daten mit allen PCs über den PCMCIA-Slot austauschen.



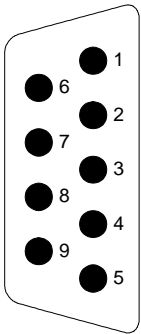
#### Hinweis!

Die Speicherkarte darf nur im ausgeschalteten Zustand des PCs gesteckt oder gezogen werden!

## Buchsen und Stecker

### Serielle Schnittstelle COM 1

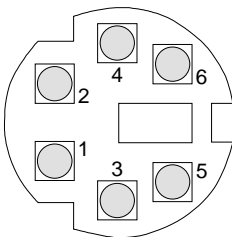
Der Stecker der seriellen Schnittstelle wird als COM 1 angesprochen und ist für Datenübertragungen für maximal 15m bei bis zu 38,4kBaude definiert. Die Kommunikation erfolgt über Daten-, Melde- und Steuerleitungen.



Pin	RS232C	RS422/485
1	DCD-	CTS-
2	RXD	RXD-
3	TXD	TXD+
4	DTR-	TXD-
5	GND	GND
6	DSR-	RXD+
7	RTS-	RTS+
8	CTS-	RTS-
9	RI-	CTS+

### PS2-Buchse KBD/MOUSE

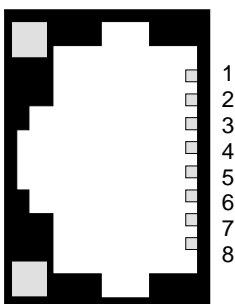
Die Pin-Belegung der beiden PS2-Buchsen ist identisch. Schließen Sie an die mit "KBD" bezeichnete Buchse Ihre Tastatur und an die mit "MOUSE" bezeichnete Buchse Ihre Maus an.



Pin	Belegung
1	+ KBD-Data (I/O)
2	reserviert
3	GND
4	+5V
5	+ KBD-Clock (I/O)
6	reserviert

### RJ45-Buchse

Über RJ45 haben Sie einen Twisted-Pair-Anschluss an Ihr Ethernet. Die Buchse hat folgende Belegung und die LEDs folgende Funktion:



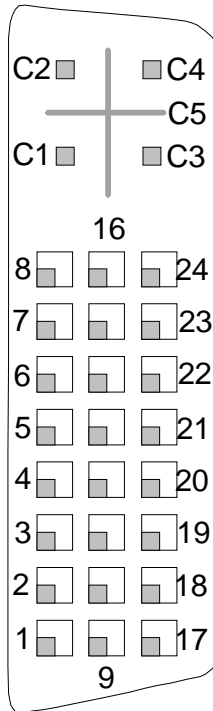
Pin	Signal
1	Transmit +
2	Transmit -
3	Receive +
4	-
5	-
6	Receive -
7	-
8	-

LED	Funktion
Ethernet-Aktivität (gelb)	Leuchtet bei aktiver Verbindung zum Ethernet blinkt bei Datentransfer
Geschwindigkeit (grün)	Leuchtet bei 100MBit Transferrate. Ansonsten brennt die LED nicht.

**DVI-Buchse**

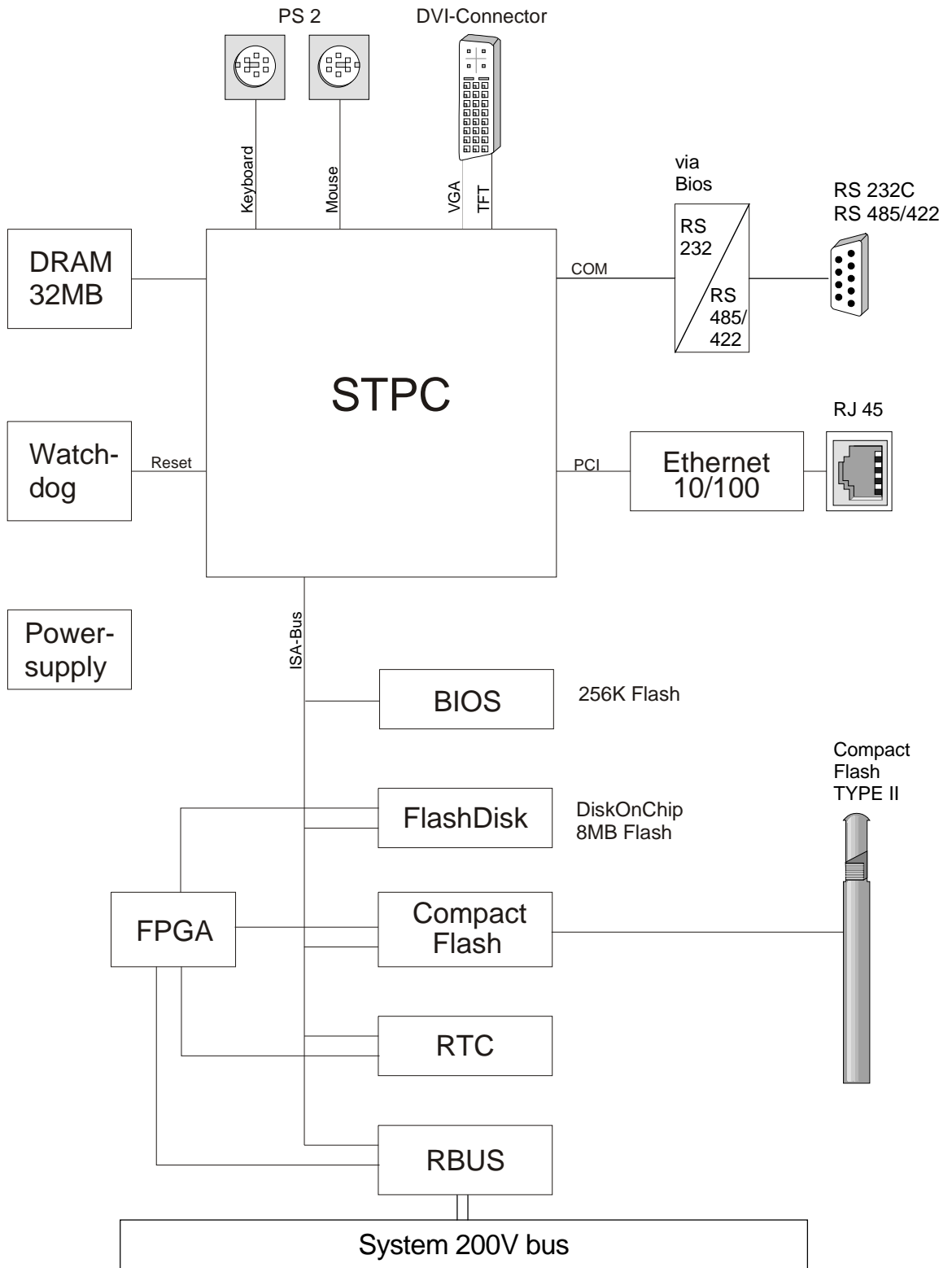
Die DVI-Buchse eignet sich zum Anschluss von analogen und digitalen Displays und Monitoren mit einer max. Auflösung von 1280x1024 Pixel.

Die Buchse hat folgende Pin-Belegung:



Pin	Signal
C1	Analog Red
C2	Analog Green
C3	Analog Blue
C4	Analog Horizontal Sync
C5	Analog RGB Return
1	T.M.D.S Data2-
2	T.M.D.S Data2+
3	T.M.D.S Data2/4 Shield
4	T.M.D.S Data4-
5	T.M.D.S Data4-
6	DDC Clock
7	DDC Data
8	Analog Vertical Sync
9	T.M.D.S Data1-
10	T.M.D.S Data1+
11	T.M.D.S Data1/3 Shield
12	T.M.D.S Data3-
13	T.M.D.S Data3+
14	+5V Power
15	Ground (return for +5V, HSync and VSync)
16	Hot Plug Detect
17	T.M.D.S Data0-
18	T.M.D.S Data0+
19	T.M.D.S Data0/5 Shield
20	T.M.D.S Data5-
21	T.M.D.S Data5+
22	T.M.D.S Clock Shield
23	T.M.D.S Clock+
24	T.M.D.S Clock-

**Blockschaltbild** Das nachfolgende Blockschaltbild zeigt den prinzipiellen Aufbau des PCs:



## Einsatz der Speichermedien

### Übersicht

Der PC 288 besitzt ein fest eingebautes Laufwerk auf Flash-ROM-Basis mit 8MB Größe und einen CompactFlash®-Slot Typ II.

Über einen CompactFlash®-Adapter wird die CompactFlash®-Karte kompatibel zum "großen" PCMCIA Typ II Format. Dies ermöglicht den Datenaustausch mit PCs mit PCMCIA-Slot.

Die Zuordnung der Laufwerke erfolgt über das BIOS-Setup-Programm.

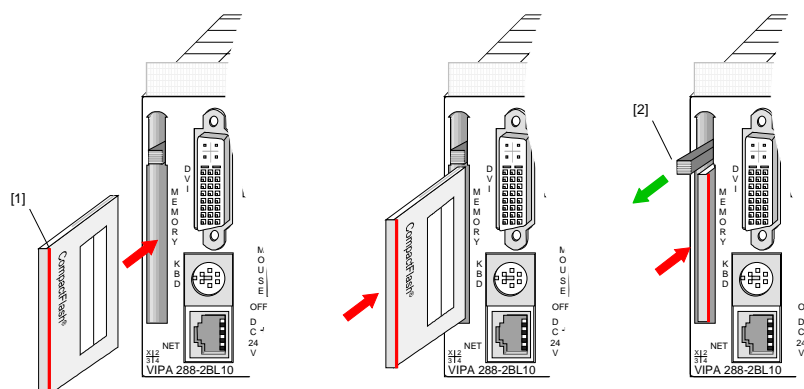
Hier gibt es verschiedene Einstellmöglichkeiten zum Bootverhalten des PC 288.

### CompactFlash® stecken/auswerfen

Jedes CompactFlash®-Speichermodul besitzt eine Ausziehkante [1]. Positionieren Sie das Modul so, dass die Ausziehkante nach rechts zeigt.

Schieben Sie ohne großen Kraftaufwand das Speicher-Modul in den PC 288 bis es einrastet und der Auswurfhebel [2] sichtbar wird.

Zum Auswurf der CompactFlash®-Karte drücken Sie den Auswurfhebel.

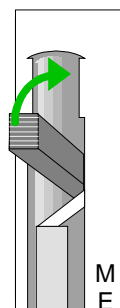


### Hinweis!

Die CompactFlash®-Speicherkarte darf nur im ausgeschalteten Zustand gezogen oder gesteckt werden!

Die Speicherkarte ist im Setup anzumelden!

### Schutz vor versehentlichem Auswerfen



Zum Schutz vor versehentlichem Auswerfen der CompactFlash®-Karte können Sie den Auswurfhebel nach oben in das Gehäuse klappen.

Zum Auswerfen klappen Sie entweder den Auswurfhebel wieder zurück oder Sie betätigen den eingeklappten Auswurfhebel mit einem spitzen Gegenstand wie z.B. einem Schraubendreher.

## Einsatz im System 200V

### Übersicht

Für den Einsatz des PC 288 sind Programmierkenntnisse in C erforderlich. VIPA liefert Ihnen mit dem PC eine offengelegte Programmierschnittstelle in Source-Codes.

Da dort alle Funktionen beschrieben sind und der Einsatz der Funktionen anhand von Beispielen näher erläutert wird, soll in diesem Handbuch nicht näher auf die Funktionen und die V-Bus-Organisation eingegangen werden.

Die Datei **vbus\_api.c** beinhaltet alle Funktionen.

In der Datei **vbus\_api.h** sind diese Funktionen beschrieben.

Ein Beispiel zum Einsatz der Funktionen finden Sie in der Datei **softsps.c**.

### automatische Adressierung

Damit die gesteckten Peripheriemodule gezielt angesprochen werden können, müssen ihnen bestimmte Adressen im PC zugeordnet werden.

Vergleichbar mit der Speicherorganisation einer CPU gibt es auch beim PC 288 einen Peripheriebereich (Adresse 0...255) und ein Prozessabbild der Ein- und Ausgänge (je Adresse 0...127).

Beim Hochlauf des PCs vergibt dieser automatisch von 0 an aufsteigend Peripherieadressen für digitale Ein-/Ausgabe-Module.

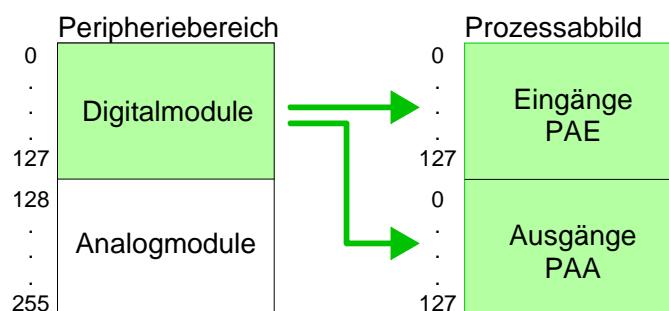
Analog-Module werden bei der automatischen Adressierung auf gerade Adressen ab Adresse 128 abgelegt.

### Signalzustände in Prozessabbild

Die Signalzustände der unteren Adresse (0...127) werden zusätzlich in einem besonderen Speicherbereich, dem *Prozessabbild* gespeichert.

Das Prozessabbild ist in zwei Teile gegliedert:

- Prozessabbild der Eingänge (PAE)
- Prozessabbild der Ausgänge (PAA)



Im Gegensatz zur CPU wird beim PC 288 das Prozessabbild nicht automatisch aktualisiert. Hierfür gibt es die Funktionen *vbus\_read\_pa* und *vbus\_write\_pa*.



**Lese- und Schreibzugriffe**

Über Lese- bzw. Schreibzugriffe auf die Peripheriebytes oder auf das Prozessabbild können Sie die Module ansprechen.



**Hinweis!**

Bitte beachten Sie, dass durch den lesenden und schreibenden Zugriff auf dieselbe Adresse unterschiedliche Module angesprochen werden können.

z.B. `vbus_read_pword (128,&w)` liest vom AI auf Steckplatz 3

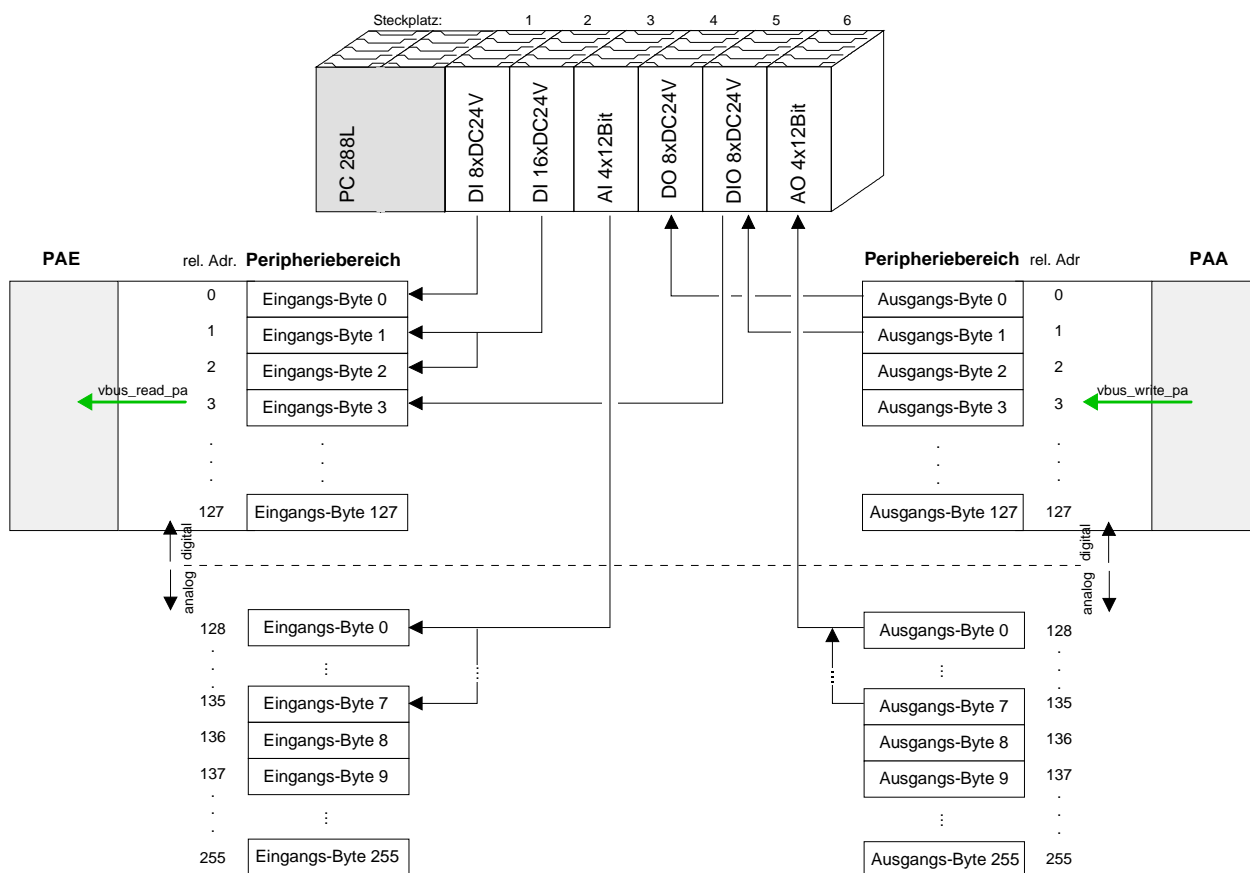
`vbus_write_pword (128,w)` schreibt auf das AO auf Steckplatz 6

Digitale und analoge Module haben bei der automatischen Adressierung getrennte Adressbereiche.

Digitalmodule: 0...127

Analogmodule: 128...255

Die nachfolgende Abbildung soll die automatische Adresszuordnung nochmals verdeutlichen:



**Zuordnung mit `set_address_table` ändern**

Mit der Funktion `set_address_table` können Sie in Ihrem Programm die automatische Adresszuordnung ersetzen, indem Sie eine eigene Zuordnungsliste erstellen und diese der Funktion `set_address_table` übergeben. Somit können auch Analogmodule in das Prozessabbild gelegt werden und Digitalmodule auf Adressen oberhalb 127!

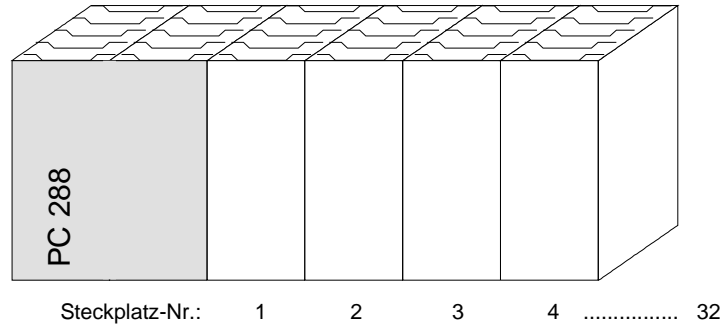
Mit der Funktion `vbus_businit` wird die neue Zuordnung übernommen.

Näheres hierzu finden Sie bei den Funktionsbeschreibungen in **vbus\_api.h**.

**Parametrierung von Modulen mit *vbus\_set\_param***

System 200V-Module wie z.B. Analog-Module können bis zu 16Byte Parametrierdaten vom PC erhalten.

Hierzu steht Ihnen die Funktion *vbus\_set\_param* zur Verfügung. *vbus\_set\_param* greift direkt über die Steckplatz-Nr. (1...32) auf das entsprechende Modul zu und legt die Parameter in einem Puffer ab.



Mit der Funktion *vbus\_businit* werden die neuen Parameter übernommen und aktiviert.

Näheres hierzu finden Sie bei den Funktionsbeschreibungen in ***vbus\_api.h***.



**Hinweis!**

Bitte beachten Sie bei beim Einsatz im System 200V, dass die Verzögerungszeit eines Peripheriemoduls bei ca. 2ms liegt, sofern nichts anderes angegeben ist!

## BIOS-Setup Bedienung

### Allgemein

Beim Einschalten des Systems erscheint am angeschlossenen Bildschirm ein Text, der die BIOS-Version anzeigt. Anschließend wird vom BIOS ein Test der Systemkomponenten und des Speichers durchgeführt. Am Ende des Tests versucht das System zu booten. Zwischen dem Systemhochlauf und dem eigentlichen Bootvorgang können Sie durch Betätigung der **[Entf]-Taste** das BIOS-Setup aufrufen.

Dies wird durch eine entsprechende Meldung am Bildschirm mitgeteilt.

Über das Setup Menü können Sie die Hardware Ihres PCs konfigurieren.

### Auswahlmenü

Nach Betätigung der [Entf]-Taste öffnet sich folgendes Auswahlmenü:

```
+-----+
|               System Bios Setup - Utility v4.2               |
|           (C) 1999 General Software, Inc. All rights reserved   |
+-----+
|
|               Basic CMOS Configuration
|               VIPA Configuration
|               Read Setup values from CMOS
|               Reset CMOS/Flash to factory defaults
|               Write to CMOS/Flash and Exit
|               Exit without changing CMOS/Flash
|
+-----+
|               <Esc> to continue (no save)
+-----+
```

Mit den Standard Cursor-Tasten können Sie die einzelnen Menüpunkte anwählen. Mit der Eingabe-Taste aktivieren Sie das angewählte Untermenü.

Über [ESC] verlassen Sie das Setup ohne zu Ihre Eingaben zu übernehmen.

**Steuertasten in den Dialogfenstern**

Jedes Dialogfenster, das Sie über das Auswahlmenü aufrufen, können Sie über folgende Tasten bedienen:

**ESC-Taste**

Mit der ESC-Taste wird das Dialogfenster geschlossen und das Auswahlmenü wieder angezeigt. Die eingestellten Parameter werden übernommen aber nicht in das CMOS geschrieben.

**Cursortasten**

Mit den Cursortasten wählen Sie einen Parameter an, den Sie ändern möchten.

**PU/PD**

Mit den Tasten PgUp und PgDn oder Bild↑ und Bild↓ oder auf dem Zahlenblock + und - können Sie den Wert eines Parameters ändern.

**Hinweis!**

Bitte beachten Sie, dass auf der Setup-Ebene noch kein Treiber für die deutsche Tastaturbelegung geladen ist. Geänderte Setup-Werte werden aber nur dann wirksam, wenn Sie ihre Speicherung beim Beenden explizit mit "Y" bestätigen. Damit Sie das Y erhalten, müssen Sie die Z-Taste betätigen.

**Basic CMOS Configuration**

In diesem Untermenü können Sie die wesentlichen Einstellungen zu Ihrem System vornehmen.

Das Menü ist in mehrere logische Untereinheiten aufgeteilt. Die Navigation erfolgt mit den Cursortasten.

```

+-----+
|                                     |
|               System Bios Setup - Basic CMOS Configuration               |
|               (C) 1999 General Software, Inc. All rights reserved         |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| DRIVE ASSIGNMENT ORDER: | Date:>Jan 01, 2000 | Typematic Delay : 250 ms | |
| Drive A: (None)           | Time: 10 : 03 : 25 | Typematic Rate   : 30 cps | |
| Drive B: (None)           | NumLock: Disabled | Seek at Boot     : None  | |
| Drive C: CompactFlash    |-----+-----+-----+-----+-----+-----+-----+-----+
| Drive D: (None)           | BOOT ORDER:      | Boot 1st: Drive C: | Config Box       : Enabled
| Drive E: (None)           | Boot 2nd: (None)   | Boot 3rd: (None)   | F1 Error Wait   : Enabled
| Drive F: (None)           | Boot 4th: (None)   | Boot 5th: (None)   | Parity Checking  : (Unused)
| Drive G: (None)           | Boot 6th: (None)   | Memory Test Tick  : Enabled
| Drive H: (None)           |                     | Test Above 1 MB  : Disabled
| Drive I: (None)           |                     | Long Memory Test  : (Unused)
| Drive J: (None)           |                     | Hexadecimal Case  : Upper
| Drive K: (None)           |-----+-----+-----+-----+-----+-----+-----+-----+
| Boot Method: Boot Sector | IDE DRIVE GEOMETRY: | Sect Hds Cyls | Memory
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
| FLOPPY DRIVE TYPES:   | Ide 0: 2 = AUTOCONFIG, PHYSICAL | Base:
| Floppy 0: Not installed  | Ide 1: Not installed             | 640KB
| Floppy 1: Not installed  | Ide 2: Not installed             | Ext:
|                     | Ide 3: Not installed             | 30MB
|-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |
|               U/D/L/R/<CR>/<Tab> to select
|               or <PgUp>/<PgDn>/+ /+ to modify
|                                     |
+-----+
    
```

**Drive Assignment Order und IDE Drive Geometry**

In diesem Teil werden den logischen Laufwerken physikalische Laufwerke zugewiesen.

Im VIPA BIOS wird nur das Laufwerk "C" unterstützt.

Für das Laufwerk "C" sind folgende Einstellungen gültig:

- "None": wenn keine CompactFlash Karte gesteckt ist
- "CompactFlash": wenn eine CompactFlash Karte gesteckt ist.



**Hinweis!**

Allen übrigen Laufwerken muss der Typ "None" zugeordnet sein. Andere Einstellungen können zu Fehlfunktionen des PCs führen.

Zu beachten ist, dass gleichzeitig mit dem Parameter für Laufwerk "C" auch die Einstellung für "Ide0" im Abschnitt *IDE Drive Geometry* verändert werden muss. Folgende Einstellungen sind gültig:

CompactFlash	DRIVE ASSIGNMENT ORDER \ Drive C:	IDE DRIVE GEOMETRY \ Ide0:
Gesteckt	CompactFlash	2 = AUTOCONFIG, PHYSICAL
nicht gesteckt	None	Not installed

Das integrierte DiskOnChip Laufwerk (DOC) wird im "VIPA Konfiguration"-Menü konfiguriert.

**Floppy drive types** Normalerweise melden Sie hier Ihr Diskettenlaufwerk an. Da im PC 288 kein Diskettenlaufwerk vorhanden ist, muss hier der Typ "Not Installed" eingestellt sein! Andernfalls kommt es zu größeren Verzögerungen während des Systemtests und beim Systemhochlauf.

**Date, Time, NumLock** Stellen Sie hier das aktuelle Datum und die Uhrzeit ein.  
Der Parameter "NumLock" legt den Status der NumLock Taste beim Booten fest.

**Hinweis!**

Steht die Echtzeituhr still, so ist anzunehmen, dass der Pufferakku des CMOS leer oder defekt ist.

Sollte nach einem Tag Betrieb der Akku noch leer sein, setzen Sie sich bitte mit der VIPA Hotline in Verbindung.

Auch bei leerem Akku sind Ihre CMOS-Einstellungen gesichert. Lediglich die Uhr und das Datum müssen neu eingestellt werden.

**Boot order** Dieser Bereich ist fest auf "Drive C" eingestellt und dient der Angabe der Bootreihenfolge.

**IDE DRIVE GEOMETRY** In diesem Abschnitt werden die Geometriedaten der IDE Laufwerke festgelegt.

Bei Einsatz einer CompactFlash stellen Sie hier "autoconfig, physical" ein. Ist keine CompactFlash gesteckt, müssen Sie hier "None" einstellen.

**Typematic Rate/Delay** Diese Parameter konfigurieren das Keyboard Interface und legen die Wiederholrate der Zeichen fest. Hier sollten Sie nichts ändern.

**Seek at Boot** Diese Option gibt an auf welche Laufwerke vor dem Booten ein "SEEK" Kommando ausgegeben wird. Grundeinstellung ist "NONE" um einen möglichst kurzen Bootvorgang zu erreichen.

**Show "Hit Del"** Ist diese Option aktiv, wird während des Hochlaufs die Möglichkeit geboten, über die [Entf]-Taste in das Setup Menü zu gelangen.

**Config Box** Hier stellen Sie ein, ob beim Hochlauf die Konfigurationseinstellungen auf dem Bildschirm ausgegeben werden sollen.

<b>F1 Error Wait</b>	Ist dieser Punkt aktiviert, so wird bei Erkennen eines Fehlers während des Hochlaufs das System angehalten und Sie können die weitere Vorgehensweise beeinflussen. Durch Drücken von [F1] setzt das System den Hochlauf fort. Mit [Entf] gelangen Sie in das Setup Menü.
<b>Parity Checking</b>	Dieser Menüpunkt wird nicht benutzt und ist bedeutungslos.
<b>Memory Test Tick</b>	Ist diese Option eingeschaltet wird während des Speichertests ein Klick Signal auf den PC-Speaker ausgegeben.
<b>Test Above 1 MB</b>	Hier bestimmen Sie, ob während des Speichertests der Speicher oberhalb der 1MB Grenze getestet werden soll oder nicht.
<b>Long Memory Test</b>	Dieser Menüpunkt wird nicht benutzt und ist bedeutungslos.
<b>Hexadecimal Case</b>	Dieser Menüpunkt legt fest in welchem Format das BIOS hexadezimale Ziffern darstellt. Mögliche Einstellung sind "UPPER" (Großschreibung) und "LOWER" (Kleinschreibung).
<b>Memory Base / Ext.</b>	Dieser Punkt zeigt die Konfiguration des Speichers unterhalb 1MB (Base Memory) und oberhalb 1MB (Extended memory) an. Die Parameter dienen rein zur Information und können nicht geändert werden.

**VIPA Configuration**

In diesem Untermenü werden die boardspezifischen Einstellungen vorgenommen. Die Navigation erfolgt mit den Cursortasten.

```

+-----+
|              System BIOS Setup - VIPA Configuration              |
|              (C) 1999 General Software, Inc. All rights reserved  |
+-----+-----+
| VGA Frame Buffer Size      : 1.0 Mb      | VGA Palette Snoop      : Enabled | |
| Graphic Clock Speed      : 85 MHz      | Watchdog               : Enabled |
| Drive C                  : CF          | Coml Mode              : RS232  |
| L1 Cache                 : Enabled     |                          |
|                          |                          |                          |
|                          |                          |                          |
|                          |                          |                          |
| Bios Version: 1.0.0     |                          |                          |
| Seriennummer: 040002   |                          |                          |
| Ausgabestand: 1        |                          |                          |
| FPGA Version: 10       |                          |                          |
|                          |                          |                          |
+-----+-----+
|              U/D/L/R/<CR>/<Tab> to select              |
|              or <PgUp>/<PgDn>/+/? to modify          |
+-----+-----+

```

**VGA Frame Buffer Size**                    Stellen Sie hier ein wie viel Hauptspeicher als Grafikspeicher für den internen Grafikkontroller verwendet werden soll.  
Dieser Speicherbereich steht dann nicht mehr als Arbeitsspeicher für den Prozessor zur Verfügung.

**Graphics Clock Speed**                    Mit dieser Option bestimmen Sie den Takt des internen Grafikkontrollers. Die Option sollte stets auf "85MHz" eingestellt sein.

**Drive C**                                    Hier legen Sie das Laufwerk fest von dem gebootet werden soll:  
"CF"                    CompactFlash (C:) dann internes Laufwerk (D:)  
"DOC"                    DiskOnChip - internes Laufwerk (C:) dann CompactFlash (D:)  
Das jeweils andere Laufwerk ist dann als Laufwerk D: ansprechbar.

**L1 Cache**                                    Mit dieser Option wird der L1 Cache im Prozessor ein- bzw. ausgeschaltet.

**VGA Palette Snoop**                    Mit diesem Punkt wird festgelegt, ob Zugriffe auf die VGA Palette nur innerhalb des STPC (disabled) erfolgen, oder auch nach außen auf den PCI Bus gelegt werden (enabled).



- Watchdog** Über diese Option können Sie einen Watchdog aktivieren, der nach einer gewissen Zeitspanne (30s) automatisch einen Reset durchführt.  
Wenn Sie in Ihrem Anwenderprogramm den Watchdog deaktivieren, können Sie sicherstellen, dass Ihr System ohne Fehler hochgefahren ist. Ansonsten wird das System nach Ablauf der Watchdog-Zeit neu gebootet. Die Watchdog-Zeit und das Watchdog-Verhalten können Sie im Systemregister einstellen.
- Com1 Mode** Hier stellen Sie die physikalischen Eigenschaften der seriellen Schnittstelle COM 1 ein. Zur Auswahl stehen:  
RS232      Betrieb als RS232C-Schnittstelle  
RS422      Betrieb als RS422/485-Schnittstelle
- Versionsdaten** Im linken Bereich des "VIPA Configuration"-Menüs werden Hardware-spezifische Daten angezeigt:  
BIOS-Version:    Versionsstand des BIOS  
Serien-Nr.:      Jeder PC 288 hat ein eigene Serien-Nr. Die Serien-Nr. ist identisch mit der Serien-Nr., die sich auf der Gehäuse-seite befindet  
Ausgabestand:    Entspricht dem Ausgabestand auf der Gehäusefront  
FPGA-Version:    Version des FPGAs, das den Zugriff auf den V-Bus verwaltet.

**Hinweis!**

Verwenden Sie diese Daten, wenn Sie Anfragen an den Service der VIPA GmbH weiterleiten, um eine effiziente Bearbeitung zu gewährleisten!

**Hinweis!**

Bei den nachfolgenden Menüpunkten erfolgt bei Anwahl eine Abfrage die mit "Yes" oder "No" zu beantworten ist.

Beachten Sie bei Verwendung einer Tastatur, dass im BIOS Setup das amerikanische Tastaturlayout benutzt wird und der Buchstabe "Y" zur positiven Bestätigung über die Taste [Z] eingegeben wird.

---

**Read Setup values from CMOS**

Diese Option lädt im Setup Menü die zuletzt im CMOS gespeicherten Werte.

---

**Reset CMOS/Flash to factory defaults**

Diese Option lädt im Setup Menü die im Auslieferungszustand eingestellten Werte.

Zur Übernahme der Werte in das CMOS ist beim Beenden der Menü-Punkt "Write to CMOS and Exit" anzuwählen.

---

**Write to CMOS/Flash and Exit**

Mit diesem Menü-Punkt speichern Sie Ihre Änderungen im CMOS-RAM.

Nach dem Speichern der Werte erfolgt automatisch ein Neustart des Systems, bei dem die geänderten Konfigurationsdaten geladen werden.

**Hinweis!**

Da die Daten parallel auch im EEPROM gespeichert werden, kann der Vorgang einige Sekunden dauern.

Während dieser Zeit darf das System **nicht** spannungslos geschaltet werden und es darf kein manueller Reset ausgelöst werden!

---

**Exit without changing CMOS/Flash**

Bei Anwahl dieses Menüpunkts wird das Setup Menü verlassen, ohne dass die Änderungen im CMOS bzw. EEPROM gespeichert und aktiviert werden.

## Registerbeschreibung

**Adressbereich** Von VIPA werden folgende Adressen belegt:

<b>270h - 277h</b>	Watchdog
<b>280h - 28Fh</b>	
280h - 284h	reserviert
285h	EEPROM Port
286h - 28Dh	reserviert
28Eh	Version
28Fh	Device ID = 84h
<b>290h - 297h</b>	
290h	WD-Timer
291h	RS232/RS422
292h	C165 Steuerregister
293h	Enable Register
294h - 297h	reserviert

**Watchdog** (I/O-Adressbereich **270h-277h**)

Der Watchdog ist nach Systemanlauf und nach Reset abgeschaltet und kann über Software ein- und ausgeschaltet werden.

Das Watchdog-Register wird über I/O-Adresse 270h mit folgenden Werten angesprochen:

Watchdog einschalten	40h nach Adresse 270h laden
Watchdog ausschalten	50h nach Adresse 270h laden
Watchdog triggern	60h nach Adresse 270h laden und anschließend 70h nach Adresse 270h laden

Der Watchdog muss nach dem Einschalten getriggert werden. Die Triggerzeit ist programmierbar:

**Triggerzeit** Parameter für die Triggerzeit I/O-Register **290h** R/W (Freigabe über **293h**)

Zeit	= Inhalt des Registers <b>290h</b> x 117ms
Inhalt	= "0" Watchdog abgeschaltet

**Freigabe**

Für WD-Timer I/O-Port **293h** nur schreiben.

Über folgende Sequenz wird das I/O-Register **290h** und **292h** für das Schreiben freigegeben:

```
WR 293h          03h
WR 293h          06h
WR 293h          03h
WR 293h          01h
```

Über folgende Sequenz wird das I/O-Register **290h** und **292h** für das Schreiben gesperrt:

```
WR 293h          03h
WR 293h          06h
WR 293h          03h
WR 293h          00h
```



**Hinweis!**

Nach Reset ist das Schreiben auf **290h** und **292h** gesperrt.

**Seriennummer**

Die Seriennummer ist im Register **271h** und **272h** für den Anwender lesbar.

**EEPROM**

Nähere Informationen über I/O-Port **285h** erhalten Sie bei VIPA.

**RS232/RS422**

Umschaltung COM1 RS232/RS422 I/O-Port **291h** R/W

	"0"	"1"
Bit 0	RS232	RS422/485
Bit 1	RX enable	TX disable
Bit 2	TX disable	RX enable
Bit 3...7	reserviert	

**C165 Steuerregister**

Die C165 Steuerregister I/O-Port **292h** R/W (Freigabe über **293h**) sind für den Download reserviert.

## Technische Daten

### PC-CPU PC 288

Elektrische Daten	VIPA 288-2BL10
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme	max. 1,5A
Ausgangsstrom Rückwandbus	max. 3,5A
Statusanzeigen (LED)	über LEDs auf der Frontseite
Anschlüsse / Schnittstellen	2polig                    Spannungsversorgung Mini-DIN                AT-Tastatur Mini-DIN                Maus 9polig                    COM1: serielle Schnittstelle PCMCIA                  CompactFlash-Karte Slot Typ II DVI                        Schnittstelle für Bildschirm/TFT RJ45                      Twisted Pair Anschluss für Ethernet ON/OFF-Schalter für Spannungsversorgung
Pufferung Uhr und CMOS	Lithium-Akku, 30 Tage Pufferung
Kombination mit Peripheriemodulen	
max. Modulanzahl	32
max. digital E/A	32
max. analoge E/A	16
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x76
Gewicht	170g



## Teil 9 Kommunikationsprozessor CP 240

### Überblick

In diesem Kapitel finden Sie Informationen über den Aufbau und die Anschlussbelegung des Kommunikationsprozessors CP 240. Den Kommunikationsprozessor CP 240 erhalten Sie von VIPA mit verschiedenen Übertragungsprotokollen, auf deren Einsatz hier näher eingegangen wird. Auch finden Sie hier die Beschreibung der Standardhantierungsbausteine.

Nachfolgend sind beschrieben:

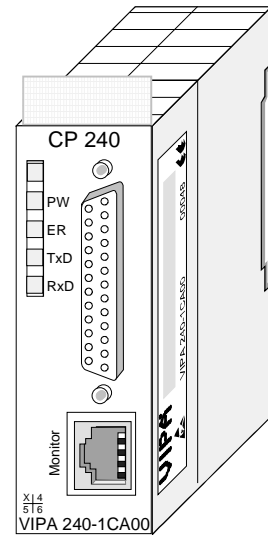
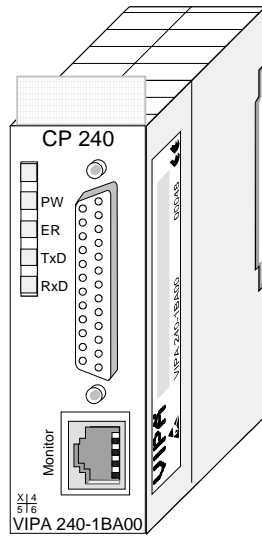
- Systemübersicht
- Protokolle ASCII, STX/ETX, 3964(R), RK512 und Modbus
- Projektierung und Anwendung des Kommunikationsprozessors
- Standardhantierungsbausteine
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 9 Kommunikationsprozessor CP 240</b> .....	<b>9-1</b>
Systemübersicht .....	9-2
Grundlagen ASCII, STX/ETX, 3964(R), RK512 .....	9-3
Grundlagen Modbus .....	9-10
CP 240 mit 20mA/RS232C-Schnittstelle - Aufbau.....	9-11
CP 240 mit RS422/RS485-Schnittstelle - Aufbau.....	9-16
Parametrierung .....	9-22
Zugriff auf das CP 240 Interface unter ASCII, STX/ETX, 3964(R) .....	9-30
Einsatz unter Modbus .....	9-32
Modbus-Funktionscodes .....	9-36
Beispiel zum Einsatz unter Modbus .....	9-39
Kommunikation über Standardhantierungsbausteine.....	9-45
Standardhantierungsbausteine für CPU 24x.....	9-46
Standardhantierungsbausteine für CPU 21x.....	9-61
Technische Daten.....	9-78

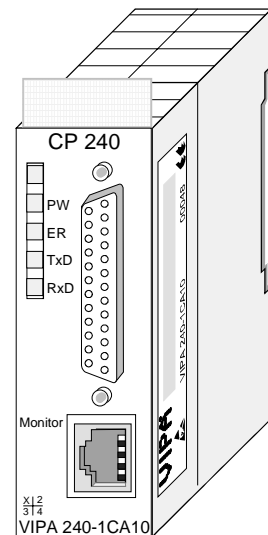
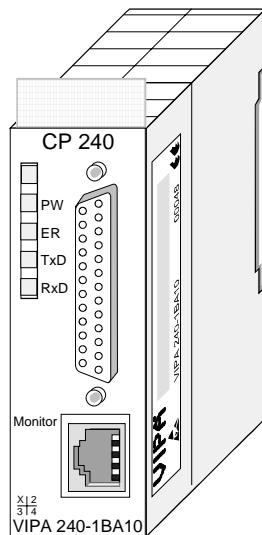
# Systemübersicht

CP 240 mit ASCII-, STX/ETX-, 3964(R)-, und RK512-Protokoll



Typ	Bestellnummer	Beschreibung
CP 240	VIPA 240-1BA00	CP 240 mit 20mA/RS232C-Schnittstelle Protokolle: ASCII, STX/ETX, 3964(R), RK512
CP 240	VIPA 240-1CA00	CP 240 mit RS422/485-Schnittstelle Protokolle: ASCII, STX/ETX, 3964(R), RK512
Diagnose-Tool	UPI-FOX2	Diagnosekabel inklusive Software

CP 240 mit Modbus-Protokoll



Typ	Bestellnummer	Beschreibung
CP 240 Modbus	VIPA 240-1BA10	CP mit 20mA/RS232C-Schnittstelle Protokoll: Modbus
CP 240 Modbus	VIPA 240-1CA10	CP mit RS422/485-Schnittstelle Protokoll: Modbus
Diagnose-Tool	UPI-FOX2	Diagnosekabel inklusive Software



## Grundlagen ASCII, STX/ETX, 3964(R), RK512

- Allgemein** Die Kommunikationsprozessoren CP 240 ermöglichen die serielle Prozess-  
ankopplung zu verschiedenen Ziel- oder Quellsystemen.  
Die CP 240-Module besitzen eine integrierte serielle Schnittstelle, die je  
nach äußerer Beschaltung und Modultyp als RS232C- oder 20mA- bzw.  
RS422 oder RS485-Schnittstelle arbeitet.  
Die CP 240-Module werden über den Rückwandbus mit der  
Betriebsspannung versorgt.
- Protokolle** Unterstützt werden die Protokolle und Prozeduren ASCII, STX/ETX, 3964(R)  
und RK512 oder Modbus (VIPA 240-1BA10 bzw. VIPA 240-1CA10).
- Diagnose-  
möglichkeit** Zu Diagnose- und Servicezwecken befindet sich auf der Frontseite eine  
Diagnoseschnittstelle, an der die Signale RxD und TxD mit RS232-Pegel  
(TTL-Pegel) anliegen.  
Bei VIPA ist ein Diagnoseadapter erhältlich (Best.-Nr.: UPI-FOX2), der die  
serielle Anbindung der Diagnoseschnittstelle an Ihren PC ermöglicht. Über  
die mitgelieferte Software können Sie den Signalfluss analysieren und die  
Funktion Ihrer Schnittstelle testen.
- Parametrierung** Zur Parametrierung können dem CP 16Byte Parameterdaten übergeben  
werden, die je nach gewähltem Protokoll entsprechend belegt sind.
- Kommunikation** Die serielle Kommunikation erfolgt unter Einsatz von Hantierungs-  
bausteinen, die für die CPU-Familien 21x und 24x von VIPA mitgeliefert  
werden.  
Die zu versendenden Daten zwischen dem CP 240 und einem Kommuni-  
kationspartner werden über die serielle Schnittstelle in einem 9 bzw. 12Bit-  
Zeichenrahmen übertragen. Für jeden Zeichenrahmen stehen je 3 Daten-  
formate zur Verfügung. Die Datenformate unterscheiden sich durch Anzahl  
der Datenbits, mit oder ohne Paritätsbit und Anzahl der Stopbits.

Zeichenrahmen	Startbit	Datenbits	Paritätsbit	Stopbit
9Bit	1	7	-	1
10Bit	1	7	-	2
10Bit	1	7	1	1
10Bit	1	8	-	1
11Bit	1	7	1	2
11Bit	1	8	1	1
11Bit	1	8	-	2
12Bit	1	8	1	2

**Protokolle und Prozeduren**

Der Datenaustausch zwischen zwei Kommunikationspartnern erfolgt über Protokolle bzw. Prozeduren wie

- ASCII-Übertragung
- STX/ETX
- 3964(R)
- RK512

---

**ASCII**

Die Datenkommunikation via ASCII ist eine einfache Form des Datenaustauschs und kann mit einer Multicast/Broadcast-Funktion verglichen werden.

Die logische Trennung der Telegramme wird über 2 Zeitfenster gesteuert. Der Sender muss innerhalb der "Zeichenverzugszeit" (ZVZ), die im Empfänger vorgegeben wird, sein Telegramm schicken.

Der Empfängers muss innerhalb der "Zeit nach Auftrag" (ZNA), die im Sender vorgegeben wird, den Telegrammempfang bestätigen.

Mit diesen beiden Zeitangaben kann eine einfach serielle SPS-SPS-Kommunikation aufgebaut werden.

Ein Sendeauftrag wird erst dann mit "Auftrag fertig ohne Fehler" (AFOF) gekennzeichnet, wenn die Daten gesendet wurden und die ZNA abgelaufen ist.

---

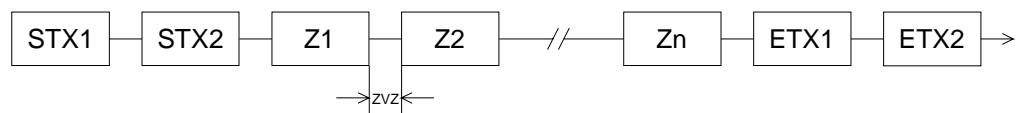
**STX/ETX**

STX/ETX ist ein einfaches Protokoll mit Header und Trailer. Die Prozedur STX/ETX wird zur Übertragung von ASCII-Zeichen (20h...7Fh) eingesetzt. Sie arbeitet ohne Blockprüfung (BCC). Sollen Daten von der Peripherie eingelesen werden, muss als Startzeichen STX (Start of Text) vorhanden sein, anschließend folgen die zu übertragenden Zeichen. Als Schlusszeichen muss ETX (End of Text) vorliegen.

Die Nutzdaten, d.h. alle Zeichen zwischen STX und ETX, werden nach Empfang des Schlusszeichens ETX an die CPU übergeben.

Beim Senden der Daten von der CPU an ein Peripheriegerät werden die Nutzdaten an den CP 240 übergeben und von dort, mit STX als Startzeichen und ETX als Schlusszeichen, an den Kommunikationspartner übertragen.

Telegrammaufbau:



Sie können bis zu 2 Anfangs- und Endezeichen frei definieren. Auch hier kann eine ZNA für den Sender vorgegeben werden.

**3964(R)**

Die Prozedur 3964(R) steuert die Datenübertragung bei einer Punkt-zu-Punkt-Kopplung zwischen dem CP 240 und einem Kommunikationspartner. Die Prozedur fügt bei der Datenübertragung den Nutzdaten Steuerzeichen hinzu. Durch diese Steuerzeichen kann der Kommunikationspartner kontrollieren, ob die Daten vollständig und fehlerfrei bei ihm angekommen sind.

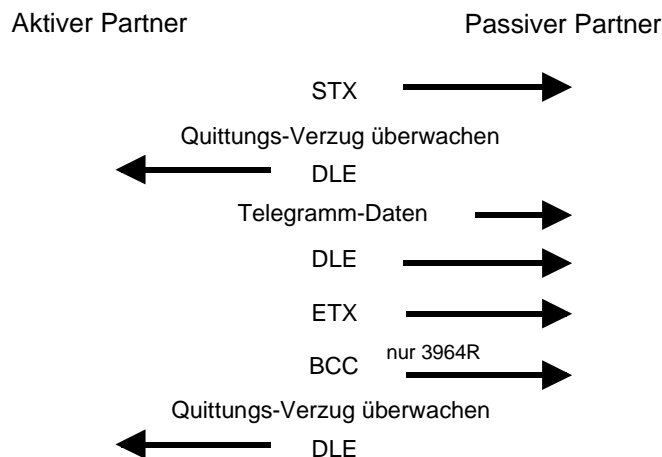
Die Prozedur wertet die folgenden Steuerzeichen aus:

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

**Hinweis!**

Wird ein DLE als Informationszeichen übertragen, so wird dieses zur Unterscheidung vom Steuerzeichen DLE beim Verbindungsauf- und -abbau auf der Sendeleitung doppelt gesendet (DLE-Verdoppelung). Der Empfänger macht die DLE-Verdoppelung wieder rückgängig.

Unter 3964(R) muss dem Kommunikationspartner eine niedrigere Priorität zugeordnet sein. Wenn beide Kommunikationspartner gleichzeitig einen Sendeauftrag erteilen, dann stellt der Partner mit niedriger Priorität seinen Sendeauftrag zurück.

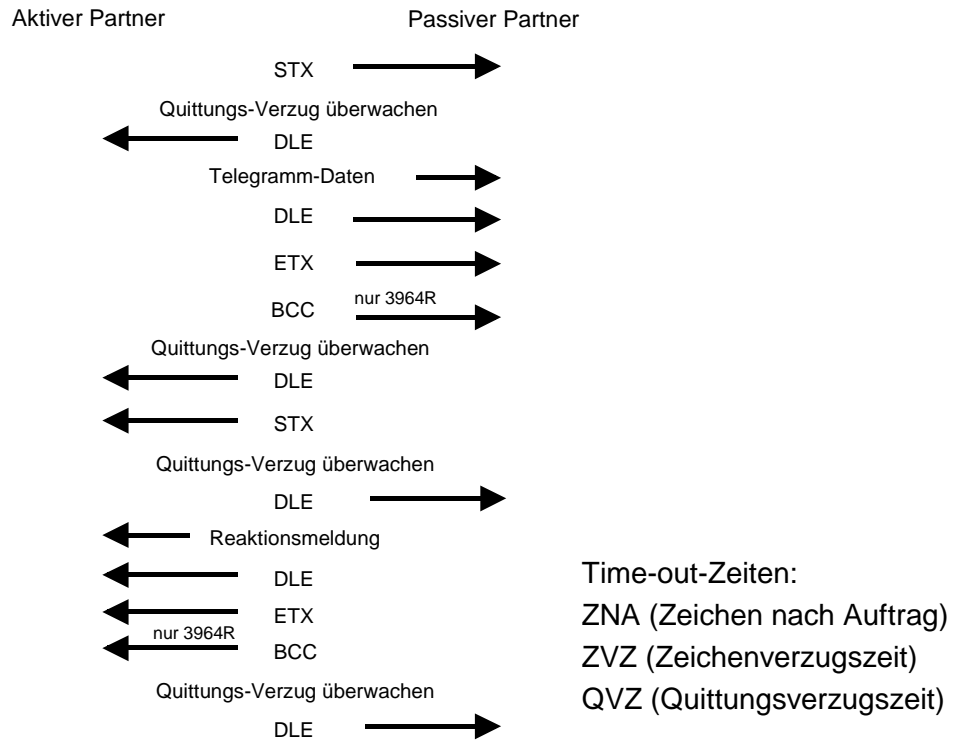
**Prozedurablauf**

Sie können pro Telegramm maximal 250Byte übertragen.

**3964(R)  
mit RK512**

Das RK512 ist ein erweitertes 3964(R). Es wird lediglich vor der Übertragung der Nutzdaten ein Telegrammkopf gesendet. Der Telegrammkopf enthält für den Kommunikationspartner Informationen über Größe, Art und Länge der Nutzdaten.

**Prozedurablauf**



**Time-out-Zeiten**

QVZ wird überwacht zwischen STX und DLE sowie zwischen BCC und DLE. ZVZ wird während des gesamten Telegramm-Empfangs überwacht. Bei Verstreichen der QVZ nach STX wird erneut STX gesendet, nach 5 Versuchen wird ein NAK gesendet und der Verbindungsaufbau abgebrochen. Dasselbe geschieht, wenn nach einem STX ein NAK oder ein beliebiges Zeichen empfangen wird. Bei Verstreichen der QVZ nach dem Telegramm (nach BCC-Byte) oder bei Empfang eines Zeichens ungleich DLE werden der Verbindungsaufbau und das Telegramm wiederholt. Auch hier werden 5 Versuche unternommen, danach ein NAK gesendet und die Übertragung abgebrochen.

**Passivbetrieb**

Wenn der Prozedurtreiber auf den Verbindungsaufbau wartet und ein Zeichen ungleich STX empfängt, sendet er NAK. Bei Empfang eines Zeichens NAK sendet der Prozedurtreiber keine Antwort. Wird beim Empfang die ZVZ überschritten, wird ein NAK gesendet und auf erneuten Verbindungsaufbau gewartet. Wenn der Prozedurtreiber beim Empfang des STX noch nicht bereit ist, sendet er ein NAK.

**Block-Check-Character (BCC-Byte)**

Zur weiteren Datensicherung wird bei der Prozedur 3964R am Ende des Telegramms ein Block-Check-Charakter angehängt. Das BCC-Byte wird durch eine XOR-Verknüpfung über die Daten des gesamten Telegramms einschließlich DLE/ETX gebildet.

Beim Empfang eines BCC-Bytes, das vom selbst Ermittelten abweicht, wird anstatt des DLEs ein NAK gesendet.

**Initialisierungskonflikt**

Versuchen beide Partner gleichzeitig innerhalb der QVZ einen Verbindungsaufbau, so sendet der Partner mit der niedrigeren Priorität das DLE und geht auf Empfang.

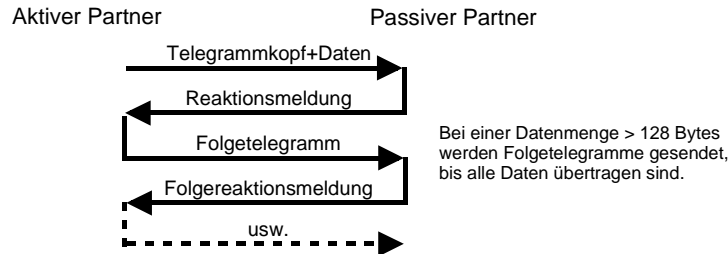
**Data Link Escape (DLE-Zeichen)**

Das DLE-Zeichen in einem Telegramm wird vom Prozedurtreiber verdoppelt, d.h. es wird DLE/DLE gesendet. Beim Empfang werden doppelte DLEs als ein DLE im Puffer abgelegt. Als Ende des Telegramms gilt immer die Kombination DLE/ETX/BCC (nur bei 3964R).

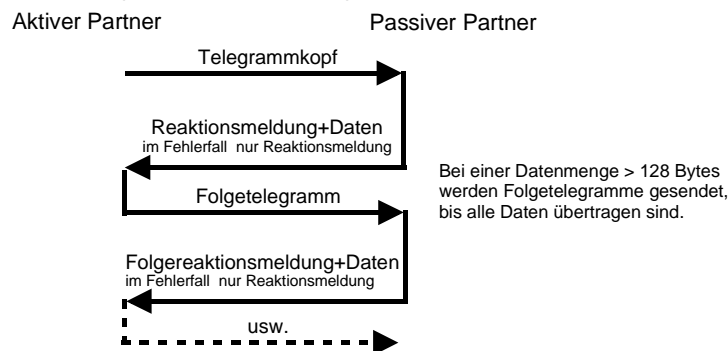
- Die Steuercodes :
- 02h = STX
  - 03h = ETX
  - 10h = DLE
  - 15h = NAK

**Logischer Telegrammablauf**

*SEND (Senden von Daten)*



*FETCH (Holen von Daten)*



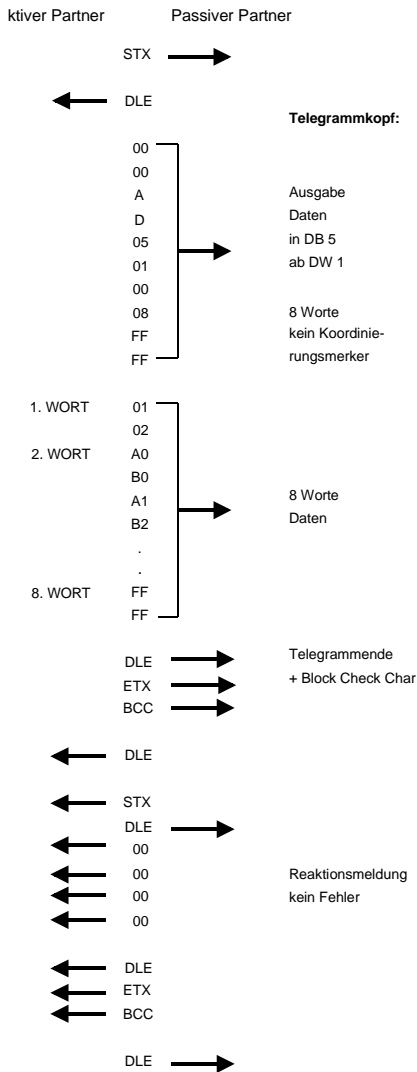
Die Prozedur wartet in beiden Fällen maximal 5s auf die Reaktionsmeldung, danach bricht sie den Empfang ab.

**Inhalt der Telegramme**

Jedes Telegramm besitzt einen Kopf. Je nach der Vorgeschichte des Telegrammverkehrs enthält dieser alle erforderlichen Informationen.

**Aufbau Ausgabe-Telegramm**

*Beispiel Ausgabetelegramm*



**Normales Telegramm**

Byte	Hex	Description
0	00	Kennung für Telegramm
1	00	
2	A	Ausgabebefehl Art der Daten
3	X	
4	xx	Parameter 1 Ziel
5	xx	
6	yy	Parameter 2 Anzahl
7	yy	
8	zz	Parameter 3 Koordinierungsmerker
9	zz	
10	aa	Daten
-	bb	
N	xy	

mit N = 10 ... 127

**Reaktionstelegramm**

Byte	Hex	Description
0	00	Kennung für Reaktionsmeldung
1	00	
2	00	Fehlercode
3	xx	

Bei Datenmengen >128Byte werden Folgetelegramme gesendet.

*Aufbau Folgetelegramme*

**Folgetelegramm**

Byte	Hex	Description
0	FF	Kennung für Folgetelegramm
1	00	
2	A	Ausgabebefehl Art der Daten
3	X	
4	aa	Daten
-	bb	
N	xy	

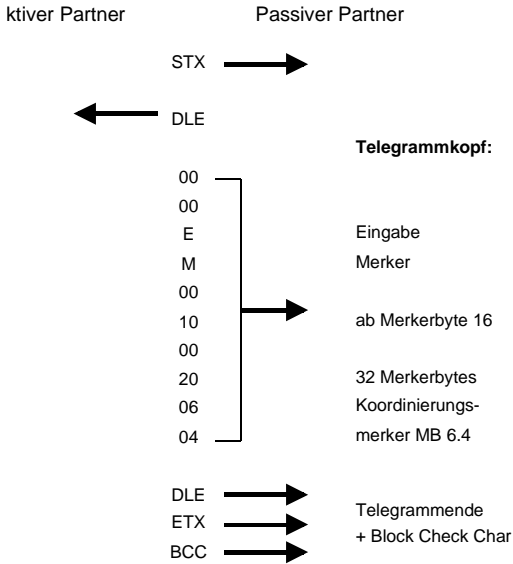
mit N = 4 ... 127

**Folge-Reaktionstelegramm**

Byte	Hex	Description
0	FF	Kennung für Folge-Reaktionstelegramm
1	00	
2	00	Fehlercode
3	xx	

### Aufbau Eingabe-Telegramm

#### Beispiel Eingabetelegramm



#### Normales Telegramm

Byte	Hex	Meaning
0	00	Kennung für Telegramm
1	00	
2	E	Eingabebefehl
3	X	Art der Daten
4	xx	Parameter 1
5	xx	Ziel
6	yy	Parameter 2
7	yy	Anzahl
8	zz	Parameter 3
9	zz	Koordinierungsmerker

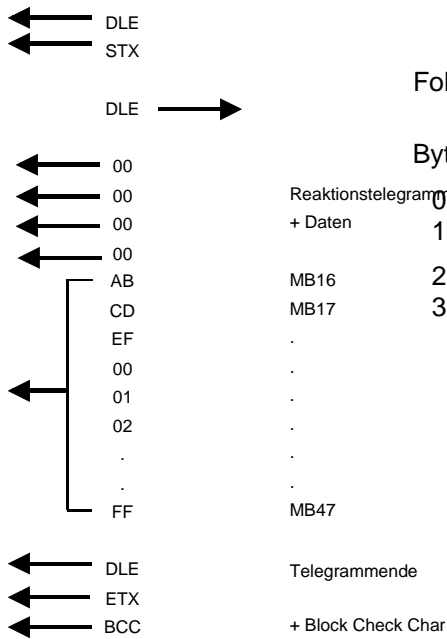
#### Reaktionstelegramm

Byte	Hex	Meaning
0	00	Kennung für Reaktionsmeldung
1	00	
2	00	
3	xx	Fehlercode
4	aa	Daten
-	bb	
N	xy	

mit N = 4 ... 127

Bei Datenmengen >128Byte werden Folgetelegramme gesendet.

#### Aufbau Folgetelegramme



#### Folgetelegramm

Byte	Hex	Meaning
0	FF	Kennung für Folgetelegramm
1	00	
2	E	Eingabebefehl
3	X	Art der Daten

#### Folge-Reaktionstelegramm

Byte	Hex	Meaning
0	FF	Kennung für Folge-Reaktionstelegramm
1	00	
2	00	
3	xx	Fehlercode
4	aa	Daten
-	bb	
N	xy	

mit N = 4 ... 127

### Koordinierungsmerker

Der Koordinierungsmerker wird im Aktiv-Betrieb im Partner-AG bei Empfang eines Telegramms gesetzt. Dies geschieht sowohl bei Eingabe- als auch bei Ausgabe-Befehlen. Ist der Koordinierungsmerker gesetzt und wird ein Telegramm mit diesem Merker empfangen, so werden die Daten nicht übernommen (bzw. übergeben), sondern es wird eine Fehler-Reaktionsmeldung gesendet (Fehlercode 32h). In diesem Fall muss der Koordinierungsmerker vom Anwender im Partner-AG zurückgesetzt werden.

## Grundlagen Modbus

### Übersicht

Das Protokoll Modbus ist ein Kommunikationsprotokoll, das eine hierarchische Struktur mit einem Master und mehreren Slaves festlegt.

Physikalisch arbeitet Modbus über eine serielle Halbduplex-Verbindung als Punkt-zu-Punkt- unter RS232 oder als Mehrpunktverbindung unter RS485.

### Master-Slave-Kommunikation

Es treten keine Buskonflikte auf, da der Master immer nur mit einem Slave kommunizieren kann. Nach einer Anforderung vom Master wartet dieser solange auf die Antwort des Slaves bis eine einstellbare Wartezeit abgelaufen ist. Während des Wartens ist eine Kommunikation mit einem anderen Slave nicht möglich.

### Telegramm-Aufbau

Die Anforderungs-Telegramme, die ein Master sendet und die Antwort-Telegramme eines Slaves haben den gleichen Aufbau:

Start- zeichen	Slave- Adresse	Funktions- Code	Daten	Fluss- kontrolle	Ende- zeichen
-------------------	-------------------	--------------------	-------	---------------------	------------------

### Broadcast mit Slave-Adresse = 0

Eine Anforderung kann an einen bestimmten Slave gerichtet sein oder als Broadcast-Nachricht an alle Slaves gehen. Zur Kennzeichnung einer Broadcast-Nachricht wird die Slave-Adresse 0 eingetragen.

Nur Schreibaufträge dürfen als Broadcast gesendet werden.

### ASCII-, RTU-Modus

Bei Modbus gibt es zwei unterschiedliche Übertragungsmodi

- ASCII-Modus: Jedes Byte wird im 2 Zeichen ASCII-Code übertragen. Die Daten werden durch Anfang- und Ende-Zeichen gekennzeichnet. Dies macht die Übertragung transparent aber auch langsam.
- RTU-Modus: Jedes Byte wird als ein Zeichen übertragen. Hierdurch haben Sie einen höheren Datendurchsatz als im ASCII-Modus. Anstelle von Anfang- und Ende-Zeichen wird eine Zeitüberwachung eingesetzt.

Die Modus-Wahl erfolgt bei der Parametrierung.

### Modbus auf dem CP 240 von VIPA

Das CP 240-Modul ist ein Kommunikationsprozessor, der je 16Byte für Ein- und Ausgabe-Daten an beliebiger Stelle in der CPU belegt.

Der CP ist über die Hardware-Konfiguration einzubinden. Hierzu ist die mitgelieferte GSD erforderlich.

Den Modbus-CP von VIPA können Sie als Master und als intelligenten Slave betreiben. Im Master-Betrieb steuern Sie die Kommunikation über Ihr SPS-Anwenderprogramm. Hierzu sind die SEND- und RECEIVE-Hantierungsbausteine erforderlich.



## CP 240 mit 20mA/RS232C-Schnittstelle - Aufbau

### Eigenschaften

- Der Kommunikationsprozessor hat die Best.-Nr.: VIPA 240-1BA00 und VIPA 240-1BA10 mit Modbus-Protokoll.
- Schnittstelle als RS232C- oder 20mA-Schnittstelle je nach äußerer Beschaltung.
- Schnittstelle kompatibel zu SSM BG41-43 mit MD26 (20mA/RS232) von VIPA und CP525 von Siemens.
- Unterstützt werden die Protokolle ASCII, STX/ETX, 3964(R), RK512 oder Modbus
- 20mA-Schnittstelle aktiv oder passiv
- 16Byte Parameterdaten
- 8 Empfangspuffer mit je 256Byte und 1 Sendepuffer mit 256Byte.
- Diagnosefunktion über Rückwandbus.
- Diagnoseschnittstelle mit TTL-Pegel.
- Die serielle Schnittstelle besitzt keine Potenzialtrennung zum Rückwandbus.
- Spannungsversorgung über Rückwandbus

### Merkmale

#### 20mA-Schnittstelle

- Logische Zustände als Strompegel
- Datenübertragung je nach Baudrate bis zu 1000m Entfernung
- Datenübertragungsrate bis 19,2kBaud
- Punkt-zu-Punkt-Verbindung (aktiv/passiv)
- Busverbindung

Optionen der Betriebsart 20mA:

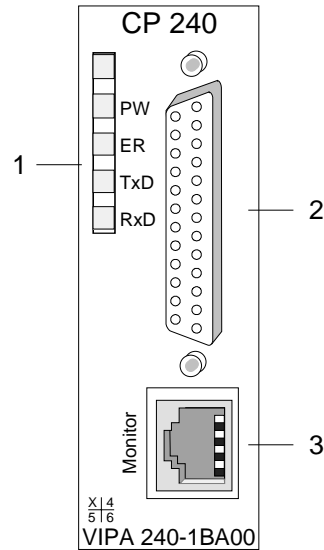
- Modul ist aktiver Teilnehmer und liefert den Linienstrom
- Modul ist passiver Teilnehmer, Gegenstelle übernimmt Linienstromversorgung

### Merkmale

#### RS232C-Schnittstelle

- Logische Zustände als Spannungspegel
- Punkt-zu-Punkt-Kopplung mit serieller Vollduplex-Übertragung in 2-Draht-Technik
- Datenübertragung bis 15m Entfernung
- Datenübertragungsrate bis 115kBaud

**Aufbau**



- [1] LED Statusanzeigen
- [2] 25polige serielle SubD-Buchse (RS232C oder 20mA)
- [3] Diagnosebuchse zur Fehlersuche

**Komponenten**

**Spannungsversorgung**

Der Kommunikationsprozessor bezieht seine Versorgungsspannung über den Rückwandbus. Bitte beachten Sie, dass das Modul extern mit 24V über die SubD-Buchse zu versorgen ist, wenn diese als aktive 20mA-Schnittstelle betrieben werden soll.

Näheres hierzu finden Sie unter "Einsatz als aktive 20mA-Schnittstelle".

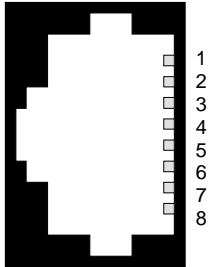
**LEDs**

Der Kommunikationsprozessor besitzt 4 LEDs, die der Betriebszustand-Anzeige dienen. Die Bedeutung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bez.	Farbe	Bedeutung
PW	Gelb	Signalisiert eine anliegende Betriebsspannung
ER	Rot	Bei Modbus: Signalisiert internen Fehler. ansonsten: Signalisiert einen Fehler durch: Leitungsunterbrechung, Überlauf, Paritätsfehler oder Zeichenrahmenfehler. Automatisches Rücksetzen der Fehler-LED nach 4s. Bei aktivierter Diagnose erfolgt dann die Fehleranzeige über die Diagnosebytes.
TxD	Gelb	Daten senden (transmit data)
RxD	Gelb	Daten empfangen (receive data)

**Diagnose-schnittstelle**

An der RJ45-Buchse stehen die Signale RxD und TxD der seriellen Schnittstelle zur Verfügung. Die Signale sind bereits auf RS232-Pegel gewandelt. Die Diagnoseschnittstelle hat folgende Pinbelegung:

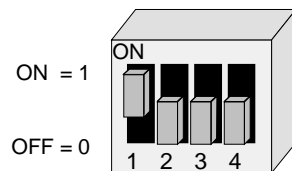


Pin	Belegung
1	reserviert
2	RxD
3	TxD
4	reserviert
5	GND
6	VCC (5V)
7	Kanalselektion
8	Kanalselektion

Über ein bei VIPA erhältliches Diagnosekabel inklusive Software (Best.-Nr.: VIPA-UPI-FOX2) können Sie mit Hilfe der Software die Signale anzeigen und auswerten.

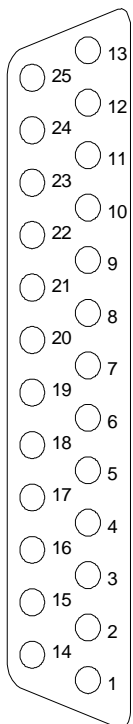


**Hinweis!**



Bitte beachten Sie, dass eine Diagnose nur dann erfolgen kann, wenn am DIP-Schalter des Diagnosekabels nebenstehende Kombination eingestellt ist.

**25polige SubD-Buchse**



Pin	RS232C	20mA	Pin	RS232C	20mA
25	nicht belegt	nicht belegt	13	reserviert	RxD+
24	reserviert	Stromsenke 20mA -Tx	12	reserviert	Stromquelle 20mA +Tx
23	RI	reserviert	11	reserviert	+24V
22	reserviert	GND 24V	10	reserviert	TxD+
21	reserviert	Stromsenke 20mA -Rx	9	nicht belegt	nicht belegt
20	Umschaltung RS232C / 20mA		8	DTR	reserviert
19	reserviert	TxD-	7	GND	GND
18	CD	reserviert	6	DSR	reserviert
17	GND	GND	5	CTS	reserviert
16	reserviert	Stromquelle 20mA +Rx	4	RTS	reserviert
15	+5V	+5V	3	RxD	reserviert
14	reserviert	RxD-	2	TxD	reserviert
			1	Schirm	Schirm

## Verkabelung

Wie oben gezeigt, besitzt der Kommunikationsprozessor eine Schnittstelle. Diese kann wahlweise als RS232C- oder als 20mA-Schnittstelle betrieben werden. Die Umschaltung zwischen den beiden Betriebsarten erfolgt über den Pin 20 an der 25-poligen Buchse:

- Pin 20 auf Masse → Betrieb als RS232C-Schnittstelle
- Pin 20 offen → Betrieb als 20mA-Schnittstelle

In der Betriebsart 20mA können Sie die Schnittstelle "aktiv" oder "passiv" betreiben.

## 20mA-Stromquelle

Für den aktiven Betrieb befindet sich auf dem CP 240 eine 20mA-Stromquelle, die über die Pins 11 (+24V) und 22 (Masse 24V) mit DC 24V zu versorgen ist.

## RS232C-Schnittstelle

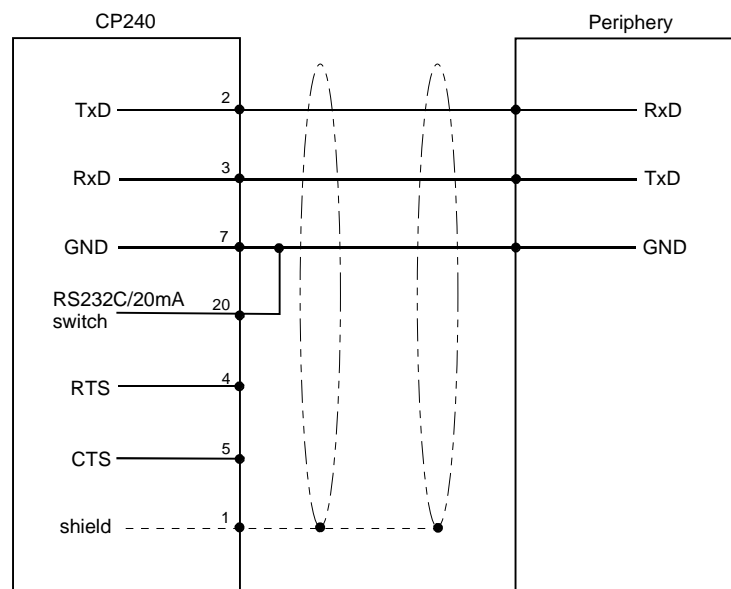
Folgende RS232C-Signale werden zur Zeit vom CP 240 unterstützt:

### TxD Transmit Data

Die Sendedaten werden über die TxD-Leitung übertragen. Die Sendeleitung wird vom CP 240 im Ruhezustand auf logisch "1" gehalten.

### RxD Receive Data

Die Empfangsdaten kommen über die RxD-Leitung an. Die Empfangsleitung muss vom Sendepartner im Ruhezustand auf logisch "1" gehalten werden.

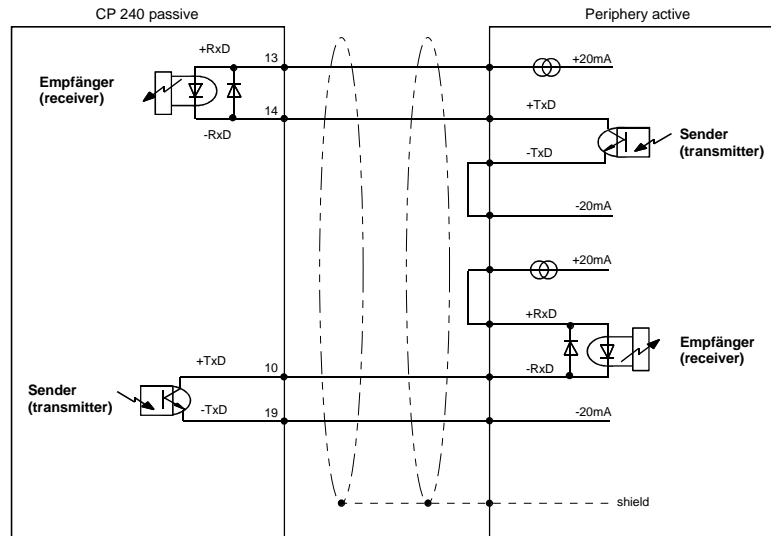


**20mA-Schnittstelle** Optionen der Betriebsart 20mA

- Modul ist passiver Teilnehmer, Gegenstelle übernimmt Linienstromversorgung
- Modul ist aktiver Teilnehmer und liefert den Linienstrom

*Passive 20mA-Schnittstelle*

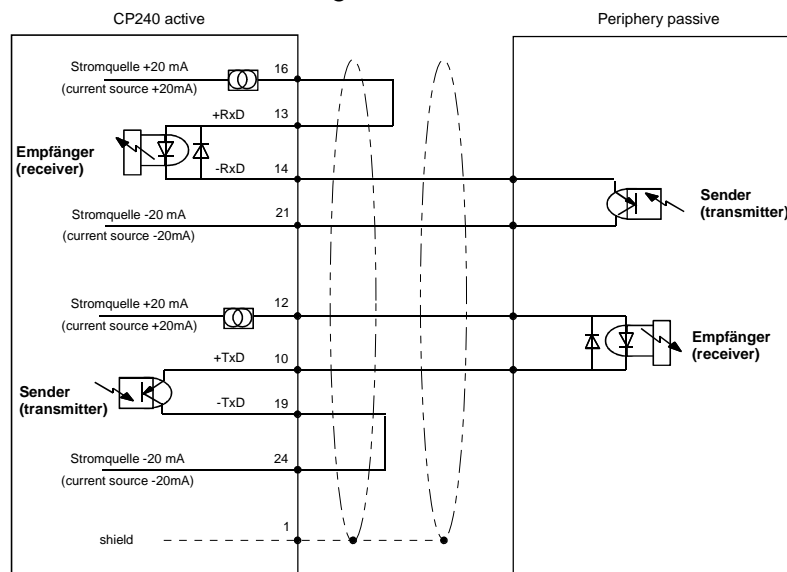
Wird der CP 240 mit "passiver" 20mA-Schnittstelle betrieben, kommt die Stromversorgung von der angeschlossenen Peripherie. Ein externe Spannungsversorgung des CP 240 ist dann nicht erforderlich.



*Aktive 20mA Schnittstelle*

Wird der CP 240 mit "aktiver" 20mA-Schnittstelle betrieben, liefert eine interne Stromquelle die 20mA. In diesem Fall ist die angeschlossene Peripherie der passive Teilnehmer.

Bitte beachten Sie, dass die Stromquelle im CP 240 extern über Pin 11 und 22 mit DC 24V zu versorgen ist.



## CP 240 mit RS422/RS485-Schnittstelle - Aufbau

### Eigenschaften

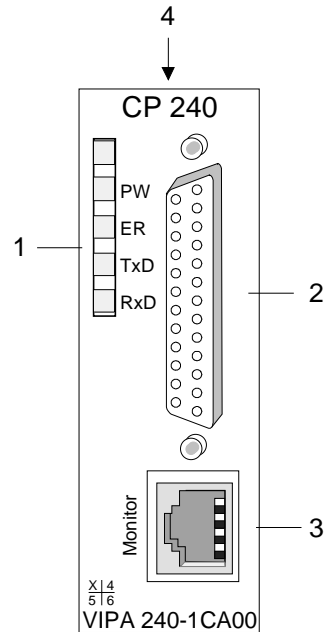
- Der Kommunikationsprozessor hat die Best.-Nr.: VIPA 240-1CA00 und VIPA 240-1CA10 mit Modbus-Protokoll
- Schnittstelle als RS422 oder RS485
- Schnittstelle kompatibel zu SSM BG41-43 mit MD21 (RS422/RS485) von VIPA und CP525 von Siemens
- Unterstützt werden die Protokolle ASCII, STX/ETX, 3964(R), RK512 oder Modbus
- 16Byte Parameterdaten
- 8 Empfangspuffer mit je 256Byte und 1 Sendepuffer mit 256Byte
- Diagnosefunktion über Rückwandbus
- Diagnoseschnittstelle mit TTL-Pegel
- Potenzialtrennung zum Rückwandbus
- Spannungsversorgung über Rückwandbus

### RS422-Schnittstelle

- Logische Zustände als Spannungsdifferenz zwischen zwei verdrehten Adern
- Punkt-zu-Punkt-Kopplung mit serieller Vollduplex-Übertragung in 4-Draht-Technik
- Multidrop-Verbindung
- Hohe Störfestigkeit
- Anschaltung von bis zu 16 Teilnehmern
- Datenübertragung bis 1000m Entfernung
- Datenübertragungsrate bis 115kBaude

### RS485-Schnittstelle

- Logische Zustände als Spannungsdifferenz zwischen zwei verdrehten Adern
- Serielle Busverbindung in Zweidrahttechnik im Halbduplex-Verfahren
- Multidrop-Verbindung
- Hohe Störfestigkeit
- Anschaltung von bis zu 32 Teilnehmern
- Datenübertragung bis 500m Entfernung
- Datenübertragungsrate bis 115kBaude

**Aufbau**

- [1] LED Statusanzeigen
- [2] 25polige serielle SubD-Buchse (RS422/485)
- [3] Diagnosebuchse zur Fehlersuche
- [4] Zuschaltbarer Abschlusswiderstand

**Komponenten****Spannungsversorgung**

Der Kommunikationsprozessor bezieht seine Versorgungsspannung über den Rückwandbus.

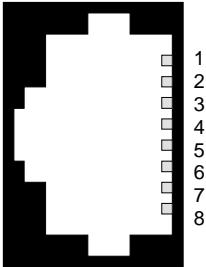
**LEDs**

Der Kommunikationsprozessor besitzt 4 LEDs, die der Betriebszustand-Anzeige dienen. Die Bedeutung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bez.	Farbe	Bedeutung
PW	Gelb	Signalisiert eine anliegende Betriebsspannung
ER	Rot	Bei Modbus: Signalisiert internen Fehler. ansonsten: Signalisiert einen Fehler durch: Leitungsunterbrechung, Überlauf, Paritätsfehler oder Zeichenrahmenfehler. Automatisches Rücksetzen der Fehler-LED nach 4s. Bei aktivierter Diagnose erfolgt dann die Fehleranzeige über die Diagnosebytes.
TxD	Gelb	Daten senden (transmit data)
RxD	Gelb	Daten empfangen (receive data)

**Diagnose-schnittstelle**

An der RJ45-Buchse stehen die Signale RxD und TxD der seriellen Schnittstelle zur Verfügung. Die Signale sind bereits auf RS232-Pegel gewandelt. Die Diagnoseschnittstelle hat folgende Pinbelegung:

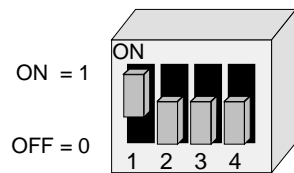


Pin	Belegung
1	reserviert
2	RxD
3	TxD
4	reserviert
5	GND
6	VCC (5V)
7	Kanalselektion
8	Kanalselektion

Über ein bei VIPA erhältliches Diagnosekabel inklusive Software (Best.-Nr.: VIPA-UPI-FOX2) können Sie mit Hilfe der Software die Signale anzeigen und auswerten.

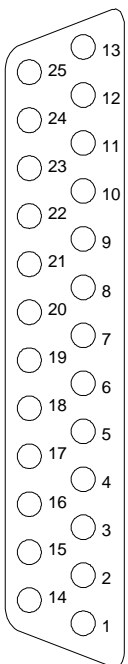


**Hinweis!**



Bitte beachten Sie, dass eine Diagnose nur dann erfolgen kann, wenn am DIP-Schalter des Diagnosekabels folgende Kombination eingestellt ist.

**25polige SubD-Buchse**



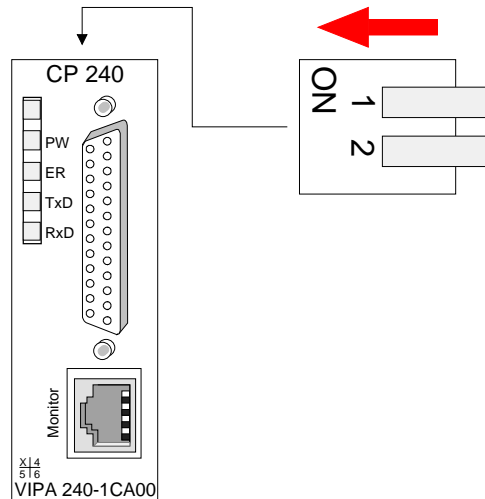
Pin	RS422/485	Pin	RS422/485
25	nicht belegt	13	CTS-
24	nicht belegt	12	CTS+
23	nicht belegt	11	nicht belegt
22	nicht belegt	10	RTS+
21	nicht belegt	9	RTS-
20	nicht belegt	8	RxD+
19	nicht belegt	7	GND
18	nicht belegt	6	RxD-
17	GND iso	5	TxD-
16	nicht belegt	4	TxD+
15	+5V	3	5V iso
14	nicht belegt	2	nicht belegt
		1	Schirm



### Abschlusswiderstand

Von der Moduloberseite kann auf der Platine ein 2-Kanal DIP-Schalter bedient werden. Hiermit können Sie zwischen die RxD- und TxD-Leitungen je einen Abschlusswiderstand von  $100\Omega$  schalten.

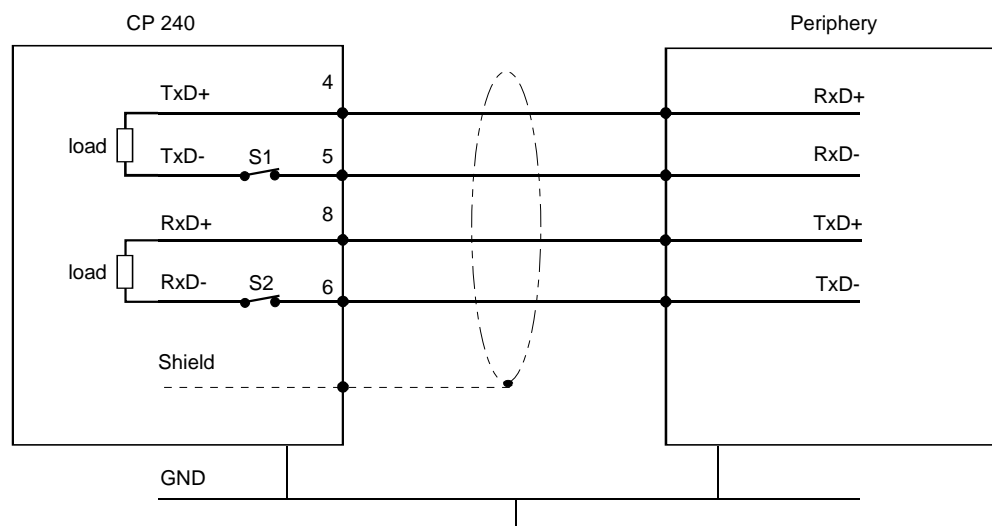
Dies ist bei großen Leitungslängen bzw. hohen Übertragungsraten erforderlich, wenn sich Ihr Modul am physikalischen Busende befindet.



### Verkabelung

Die Schnittstelle können Sie wahlweise für Punkt-zu-Punkt-Kopplung (RS422) oder für ein Bussystem mit Senden und Empfangen auf der gleichen Leitung (RS485) einsetzen. In diesem Fall wird über einen Busmaster durch entsprechende Bedienung des SEL-Signals der Betriebsmodus umgeschaltet.

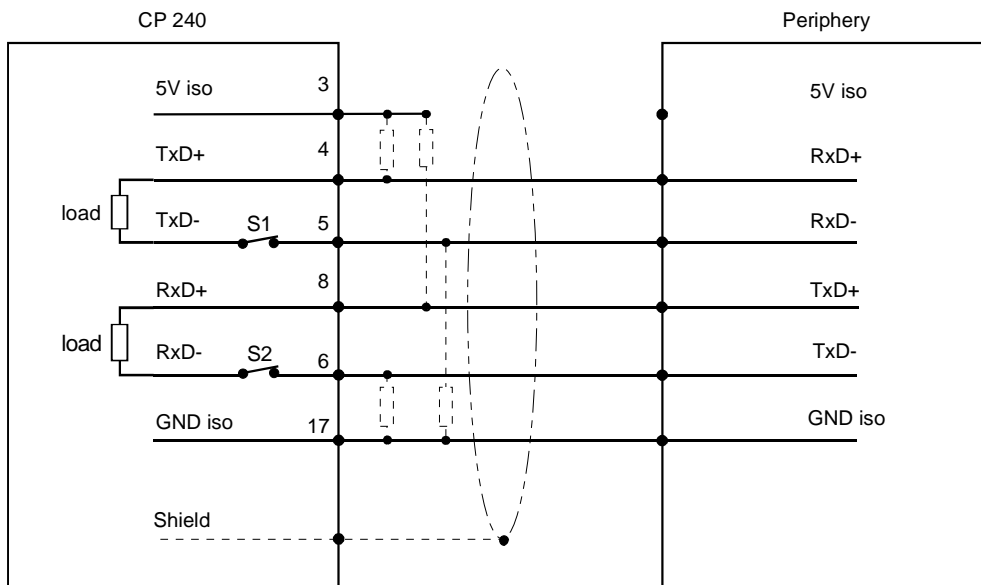
### RS422 potenzialgebunden



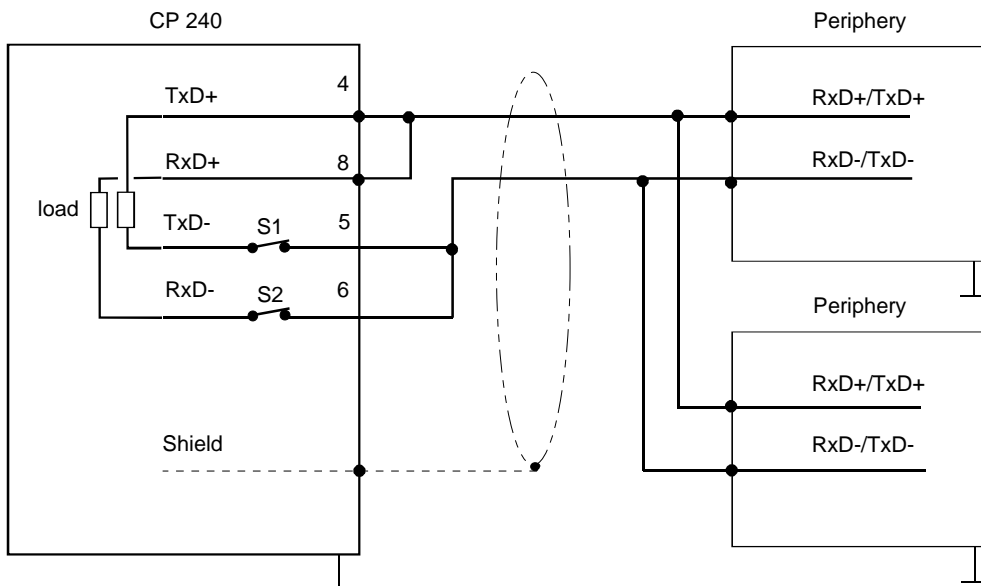
### Achtung!

Die Massen der Schnittstellen der beiden Geräte müssen verbunden sein.

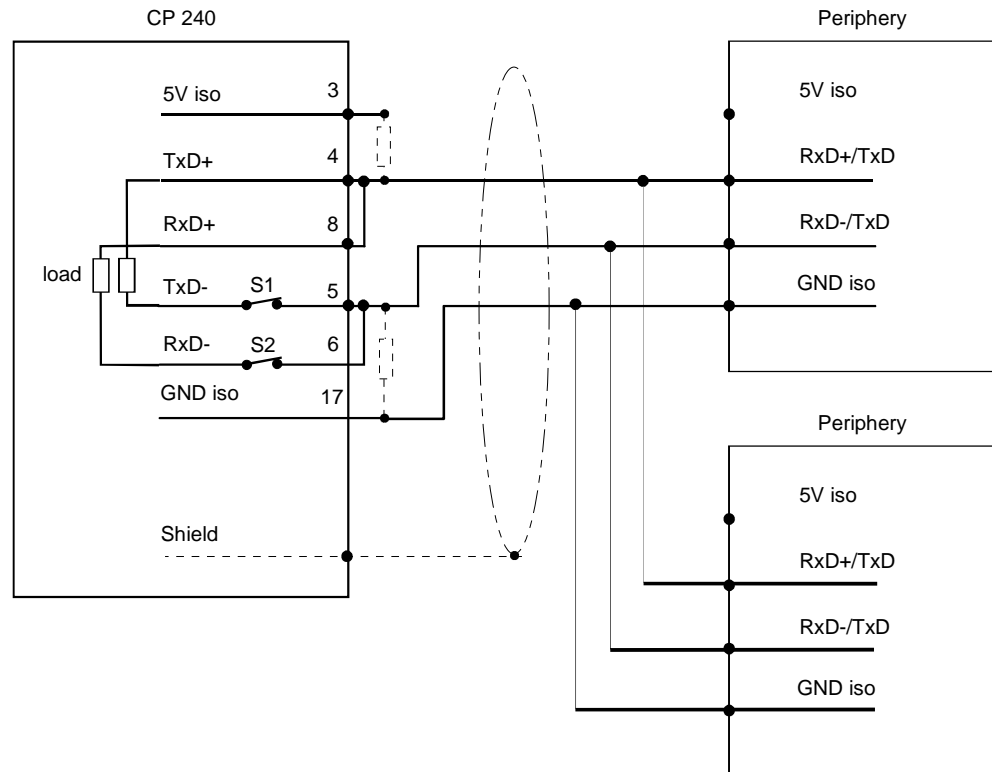
**RS422**  
potenzialgetrennt



**RS485**  
potenzialgebunden



## RS485 potenzialgetrennt



### Definierte Ruhepegel über Widerstände

Bei potenzialgetrennten Schnittstellen haben Sie auf Pin 3 isolierte 5V und an Pin 17 die zugehörige Masse GND. Mit dieser isolierten Spannung können Sie über Widerstände zu den Signalleitungen definierte Ruhepegel vergeben und für einen reflexionsarmen Abschluss sorgen.

## Parametrierung

### Allgemein

Sie können dem CP 240 zur Parametrierung 16Byte Parameterdaten übergeben. Der Aufbau der Parameterdaten richtet sich nach dem gewählten Protokoll bzw. gewählter Prozedur.

Nachfolgend finden Sie eine Auflistung der Parameterbytes mit ihren Default-Werten.

### Aufbau Parameterbytes bei ASCII

Byte	Funktion	Wertebereich	Defaultparameter
0	Baudrate	0h: Default (9600Baud) 1h: 150Baud 2h: 300Baud 3h: 600Baud 4h: 1200Baud 5h: 1800Baud 6h: 2400Baud 7h: 4800Baud 8h: 7200Baud 9h: 9600Baud Ah: 14400Baud Bh: 19200Baud Ch: 38400Baud Dh: 57600Baud Eh: 115200Baud	0: 9600Baud
1	Protokoll	1: ASCII	1 (ASCII)
2	Bit 1/0 Datenbits	00b: 5 Datenbits 01b: 6 Datenbits 10b: 7 Datenbits 11b: 8 Datenbits	11b: 8 Datenbits
	Bit 3/2 Parity	00b: none 01b: odd 11b: even	11b: none
	Bit 5/4 Stopbits	01b: 1 10b: 1,5 11b: 2	01b: 1 Stopbit
	Bit 7/6 Flusskontrolle	00b: keine 01b: Hardware 10b: XON/XOFF	00b: keine
3	Reserviert	0	0
4	ZNA (*20ms)	0..255	0
5	ZVZ (*20ms)	0..255	10
6	Anz.Receivebuffer	1..8	1
7...15	reserviert		

**Aufbau Parameter-  
bytes bei STX/ETX**

Byte	Funktion	Wertebereich	Defaultparameter
0	Baudrate	0h: Default (9600Baud) 1h: 150Baud 2h: 300Baud 3h: 600Baud 4h: 1200Baud 5h: 1800Baud 6h: 2400Baud 7h: 4800Baud 8h: 7200Baud 9h: 9600Baud Ah: 14400Baud Bh: 19200Baud Ch: 38400Baud Dh: 57600Baud Eh: 115200Baud	0: 9600Baud
1	Protokoll	2: STX/ETX	2 (STX/ETX)
2	Bit 1/0 Datenbits	00b: 5 Datenbits 01b: 6 Datenbits 10b: 7 Datenbits 11b: 8 Datenbits	11b: 8 Datenbits
	Bit 3/2 Parity	00b: none 01b: odd 11b: even	00b: none
	Bit 5/4 Stopbits	01b: 1 10b: 1,5 11b: 2	01b: 1 Stopbit
	Bit 7/6 Flusskontrolle	00b: keine 01b: Hardware 10b: XON/XOFF	00b: keine
3	Reserviert	0	0
4	ZNA (*20ms)	0..255	0
5	TMO (*20ms)	0..255	10
6	Anzahl Startkennungen	0..2	01
7	Startkennung 1	0..255	02
8	Startkennung 2	0..255	0
9	Anzahl Endekennungen	0..2	01
10	Endekennung 1	0..255	03
11	Endekennung 2	0..255	0
12	not used		
13	not used		
14	not used		
15	not used		

**Aufbau Parameter-  
bytes bei 3964(R) /  
3964(R) mit RK512**

Byte	Funktion	Wertebereich	Defaultparameter
0	Baudrate	0h: Default (9600Baud) 1h: 150Baud 2h: 300Baud 3h: 600Baud 4h: 1200Baud 5h: 1800Baud 6h: 2400Baud 7h: 4800Baud 8h: 7200Baud 9h: 9600Baud Ah: 14400Baud Bh: 19200Baud Ch: 38400Baud Dh: 57600Baud Eh: 115200Baud	0: 9600Baud
1	Protokoll	3: 3964 4: 3964R 5: 3964 + RK512 6: 3964R + RK512	3: 3964
2	Bit 1/0 Datenbits	00b: 5 Datenbits 01b: 6 Datenbits 10b: 7 Datenbits 11b: 8 Datenbits	11b: 8 Datenbits
	Bit 3/2 Parity	00b: none 01b: odd 11b: even	00b: none
	Bit 5/4 Stopbits	01b: 1 10b: 1,5 11b: 2	01b: 1 Stopbit
	Bit 7/6 Flusskontrolle	00b: keine 01b: Hardware 10b: XON/XOFF	00b: keine
3	Reserviert	0	0
4	ZNA (*20ms)	0..255	0
5	ZVZ (*20ms)	0..255	10
6	QVZ (*20ms)	0..255	25
7	BWZ (*100ms)	0..255	100
8	STX Wiederholungen	0..255	3
9	DBL	0..255	6
10	Priorität	0: low 1: high	0: low
11	reserviert	0	0
12	QVZ (*100ms) Anwenderquittung RK512	0..255	50
13	not used		
14	not used		
15	not used		

**Aufbau Parameter bei Modbus**

Byte	Funktion	Wertebereich	Defaultparameter
0	Baudrate	0h: 9600Baud 6h: 2400Baud 7h: 4800Baud 9h: 9600Baud Ah: 14400Baud Bh: 19200Baud Ch: 38400Baud	0h: 9600Baud
1	Protokoll	Ah: Modbus Master ASCII Bh: Modbus Master RTU Ch: Modbus Slave ASCII Dh: Modbus Slave RTU	Bh: Modbus-Master RTU
2	Bit 1/0 Datenbits	00b: 5 Datenbits 01b: 6 Datenbits 10b: 7 Datenbits 11b: 8 Datenbits	11b: 8 Datenbits
	Bit 3/2 Parity	00b: none 01b: odd 11b: even	00b: none
	Bit 5/4 Stopbits	01b: 1 10b: 1,5 11b: 2	01b: 1 Stopbit
	Bit 7/6 Flusskontrolle	00b: keine 01b: Hardware 10b: XON/XOFF	00b: keine
3	Reserviert	0	0
4	Adresse	1...255	1
5	Debug	0: Debug aus 1: Debug ein	0
6...7	Wartezeit	0: automat. Berechnung 1 ... 60000: Zeit in ms	0
8	nicht benutzt		
9	nicht benutzt		
10	nicht benutzt		
11	nicht benutzt		
12	nicht benutzt		
13	nicht benutzt		
14	nicht benutzt		
15	nicht benutzt		

**Hinweis zu den Defaultparametern!**

Sofern keine Parametrierung vorhanden ist und der CP 240 über Auto-Adressierung eingebunden werden soll, besitzt der CP folgende Default-Parameter:

Baudrate: 9600Baud, Protokoll: ASCII, Datenbits: 8, **Parity: even**, Stopbits: 1, Flusskontrolle: keine, ZNA: 0, ZVZ: 200ms, Receivebuffer: 1

---

**Parameter-  
beschreibung****Baudrate  
(bei allen  
Protokollen)**

Geschwindigkeit der Datenübertragung in bit/s (Baud).  
Sie haben folgende Einstellmöglichkeiten:

0h:	Default (9600Baud)
1h:	150Baud
2h:	300Baud
3h:	600Baud
4h:	1200Baud
5h:	1800Baud
6h:	2400Baud
7h:	4800Baud
8h:	7200Baud
9h:	9600Baud
Ah:	14400Baud
Bh:	19200Baud
Ch:	38400Baud
Dh:	57600Baud
Eh:	115200Baud

*Default: 0 (9600Baud)*

**Protokoll**

Das Protokoll, das verwendet werden soll. Diese Einstellung beeinflusst den weiteren Aufbau der Parameterdaten.

Abhängig vom eingesetzten CP 240-Modul haben Sie folgende Einstellmöglichkeiten:

nur VIPA 240-1BA00 und VIPA 240-1CA00:

1:	ASCII
2:	STX/ETX
3:	3964
4:	3964R
5:	3964 und RK512
6:	3964R und RK512

nur VIPA 240-1BA10 und VIPA 240-1CA10:

Ah:	Modbus Master mit ASCII
Bh:	Modbus Master mit RTU
Ch:	Modbus Slave mit ASCII
Dh:	Modbus Slave mit RTU



**Übertragungsparameter-Byte (bei allen Protokollen)**

Über Bitmuster stellen Sie hier die physikalischen Übertragungsparameter ein. Das Byte hat folgenden Aufbau:

Byte	Funktion	Wertebereich	Defaultparameter
2	Bit 1/0 Datenbits	00b: 5 Datenbits 01b: 6 Datenbits 10b: 7 Datenbits 11b: 8 Datenbits	11b: 8 Datenbits
	Bit 3/2 Parity	00b: none 01b: odd 11b: even	00b: none
	Bit 5/4 Stopbits	01b: 1 10b: 1,5 11b: 2	01b: 1 Stopbit
	Bit 7/6 Flusskontrolle	00b: keine 01b: Hardware 10b: XON/XOFF	00b: keine

*Datenbits*

Anzahl der Datenbits, auf die ein Zeichen abgebildet wird.

*Parity*

Die Parität ist je nach Wert gerade oder ungerade. Zur Paritätskontrolle werden die Informationsbits um das Paritätsbit erweitert, das durch seinen Wert ("0" oder "1") den Wert aller Bits auf einen vereinbarten Zustand ergänzt. Ist keine Parität vereinbart, wird das Paritätsbit auf "1" gesetzt, aber nicht ausgewertet.

*Stopbits*

Die Stopbits werden jedem zu übertragenden Zeichen nachgesetzt und kennzeichnen das Ende eines Zeichens.

*Flusskontrolle*

Mechanismus, der den Datentransfer synchronisiert, wenn der Sender schneller Daten schickt als der Empfänger verarbeiten kann. Die Flusskontrolle kann hardware- oder softwaremäßig (XON/XOFF) erfolgen. Bei der Hardware-Flusskontrolle werden die Leitungen RTS und CTS verwendet, die dann entsprechend zu verdrahten sind.

Die Software-Flusskontrolle verwendet zur Steuerung die Steuerzeichen XON=11h und XOFF=13h. Bitte beachten Sie, dass dann Ihre Daten diese zwei Steuerzeichen nicht beinhalten dürfen.

*Default: 13h (Datenbits: 8, Parität: keine, Stopbit: 1, Flusskontrolle: keine)*

<b>Zeit nach Auftrag (ZNA)</b> (bei allen Protokollen außer Modbus)	Wartezeit, die eingehalten wird, bis ein Auftrag ausgeführt wird. Die ZNA wird in 20ms-Einheiten angegeben. <i>Bereich: 0 ... 255</i>	<i>Default: 0</i>
<b>Zeichenverzugszeit (ZVZ)</b> (bei ASCII, 3964(R) und RK512)	Die Zeichenverzugszeit definiert den maximal zulässigen zeitlichen Abstand zwischen zwei empfangenen Zeichen innerhalb eines Telegramms. Die ZVZ wird in 20ms-Einheiten angegeben. Bei ZVZ=0 berechnet sich der CP anhand der Baudrate die ZVZ selbst (ca. doppelte Zeichenzeit). <i>Bereich: 0 ... 255</i>	<i>Default: 10</i>
<b>Anzahl Receive-buffer</b> (nur bei ASCII)	Legt die Anzahl der Empfangspuffer fest. Solange nur 1 Empfangspuffer verwendet wird und dieser belegt ist, können keine weiteren Daten empfangen werden. Durch Aneinanderreihung von bis zu 8 Empfangspuffern können die empfangenen Daten in einen noch freien Empfangspuffer umgeleitet werden. <i>Bereich: 1 ... 8</i>	<i>Default: 1</i>
<b>Time-out (TMO)</b> (nur bei STX/ETX)	Mit TMO definieren Sie den maximal zulässigen zeitlichen Abstand zwischen zwei Telegrammen. TMO wird in 20ms-Einheiten angegeben. <i>Bereich: 0 ... 255</i>	<i>Default: 10</i>
<b>Anzahl Startkennungen</b> (nur bei STX/ETX)	Hier können Sie 1 oder 2 Startkennungen einstellen. Ist "1" als Startkennung eingestellt, wird der Inhalt des 2. Startkennzeichens (Byte 8) ignoriert. <i>Bereich: 0 ... 2</i>	<i>Default: 1</i>
<b>Startkennung 1 und 2 (STX)</b> (nur bei STX/ETX)	ASCII-Wert des Startzeichens, das einem Telegramm vorausgeschickt wird und den Start einer Übertragung kennzeichnet. Sie können 1 oder 2 Startzeichen verwenden. Bei Einsatz von 2 Startzeichen müssen Sie unter "Anzahl Startkennungen" eine 2 eintragen. <i>Anzahl Startkennungen: Bereich: 0 ... 2</i> <i>Startkennung 1, 2: Bereich: 0 ... 255</i>	<i>Default: 1</i> <i>Default: 2 (Kennung 1)</i> <i>0 (Kennung 2)</i>
<b>Endekennung 1 und 2 (ETX)</b> (nur bei STX/ETX)	ASCII-Wert des Endezeichens, das nach einem Telegramm folgt und das Ende einer Übertragung kennzeichnet. Sie können 1 oder 2 Endezeichen verwenden. Bei Einsatz von 2 Endezeichen müssen Sie unter "Anzahl Endekennungen" eine 2 eintragen. <i>Anzahl Endekennungen: Bereich: 0 ... 2</i> <i>Endekennung 1, 2: Bereich: 0 ... 255</i>	<i>Default: 1</i> <i>Default: 3 (Kennung 1)</i> <i>0 (Kennung 2)</i>

<b>Quittungs- verzugszeit (QVZ)</b> (bei 3964(R), RK512)	Die Quittungsverzugszeit definiert den maximal zulässigen zeitlichen Abstand bis zur Quittung des Partners bei Verbindungsaufbau. Die QVZ wird in 20ms-Einheiten angegeben. <i>Bereich: 0 ... 255</i> <span style="float: right;"><i>Default: 25</i></span>
<b>Blockwartezeit (BWZ)</b> (bei 3964(R), RK512)	Die BWZ wird in 100ms-Einheiten angegeben. <i>Bereich: 0 ... 255</i> <span style="float: right;"><i>Default: 100</i></span>
<b>STX- Wiederholungen</b> (bei 3964(R), RK512)	Maximale Anzahl der Versuche des CP 240 eine Verbindung aufzubauen. <i>Bereich: 0 ... 255</i> <span style="float: right;"><i>Default: 3</i></span>
<b>Wiederholung Datenblöcke (DBL)</b> (bei 3964(R), RK512)	Maximale Anzahl der Telegrammwiederholungen (inkl. des 1. Telegramms) bei Fehlern <i>Bereich: 0 ... 255</i> <span style="float: right;"><i>Default: 6</i></span>
<b>Priorität</b> (bei 3964(R), RK512)	Ein Kommunikationspartner hat hohe Priorität, wenn sein Sendeversuch Vorrang gegenüber dem Sendewunsch des Partners hat. Bei niedriger Priorität muss dieser hinter dem Sendewunsch des Partners zurückstehen. Bei den Protokollen 3964(R) müssen die Prioritäten beider Partner unterschiedlich sein. Sie haben folgende Einstellmöglichkeiten: 0: low 1: high <span style="float: right;"><i>Default: 0 (low)</i></span>
<b>QVZ Anwender- quittung RK512</b> (nur bei RK512)	Dies ist die Wartezeit innerhalb der eine Quittung vom Partner kommen muss, die besagt, dass die Daten empfangen und weiterverarbeitet wurden (Reaktionstelegramm). Die QVZ wird in 100ms-Einheiten angegeben. <i>Bereich: 0 ... 255</i> <span style="float: right;"><i>Default: 50</i></span>
<b>Adresse</b> (nur Modbus)	Stellen Sie hier im Slave-Modus die Modbus-Slave-Adresse ein. <i>Bereich: 1 ... 255</i> <span style="float: right;"><i>Default: 1</i></span>
<b>Debug</b> (nur Modbus)	Dieser Modus ist für interne Tests. Diese Funktion sollte immer deaktiviert sein. <i>Bereich: 0, 1</i> <span style="float: right;"><i>Default: 0</i></span>

## Zugriff auf das CP 240 Interface unter ASCII, STX/ETX, 3964(R)

Die hier gezeigten Zugriffsmethoden gelten **nicht** für das **Modbus-Protokoll!**

### Daten senden und empfangen

Daten, die von der CPU über den Rückwandbus in den entsprechenden Datenkanal geschrieben werden, werden vom Kommunikationsprozessor in den entsprechenden Sendepuffer (256Byte) geschrieben und von dort über die Schnittstelle ausgegeben.

Empfängt der Kommunikationsprozessor Daten über die Schnittstelle, werden die Daten in einem Ringpuffer abgelegt (8x256Byte). Die empfangenen Daten können über den Datenkanal von der CPU gelesen werden.

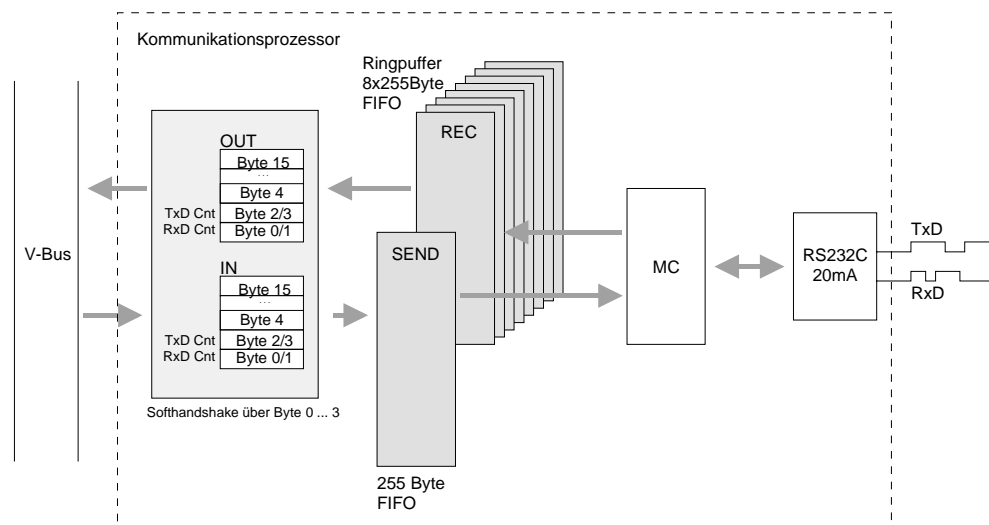
### Kommunikation über Rückwandbus

Der Austausch von empfangenen Telegrammen über den Rückwandbus erfolgt asynchron. Ist ein komplettes Telegramm über die serielle Schnittstelle eingetroffen (Ablauf der ZVZ), so wird dies in einem 8x256Byte großen Ringpuffer abgelegt. Aus der Länge des Ringpuffers ergibt sich die max. Länge eines Telegramms. Es können 8 unabhängige je 256Byte große Telegramme gespeichert werden. Ist der Puffer voll, werden neu ankommende Telegramme verworfen. Ein komplettes Telegramm wird in je 12Byte große Blöcke unterteilt und an den Rückwandbus übergeben. Das Zusammensetzen der Datenblöcke hat in der CPU zu erfolgen.

### Aufgaben der CPU

Ein zu sendendes Telegramm ist in der CPU in 12Byte große Blöcke zu unterteilen und über den Rückwandbus an den CP 240 zu übergeben. Im CP 240 werden diese Blöcke im Sendepuffer zusammengesetzt und bei Vollständigkeit des Telegramms über die serielle Schnittstelle 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**

Es soll z.B. ein Telegramm mit der Länge von 30Byte gesendet werden. So werden von der CPU die ersten 12Byte Nutzdaten des Telegramms in die Bytes 4 bis 15 und in Byte 2/3 die Länge des Telegramms (also "30") geschrieben. Der CP 240 empfängt die Daten über den Rückwandbus und kopiert die 12Byte Nutzdaten in den Sendepuffer. Zur Quittierung des Empfangs schreibt der CP 240 in Byte 2/3 den Wert "30" (Länge des Telegramms) zurück.

Beim Empfang der "30", kann die CPU weitere 12Byte Nutzdaten in Byte 4 bis 15 und die Restlänge des Telegramms ("18" Byte) in Byte 2/3 an den CP 240 senden. Dieser speichert wieder die Nutzdaten im Sendepuffer und gibt die Restlänge des Telegramms ("18") auf Byte 2/3 an die CPU zurück.

Beim Empfang der "18", kann die CPU die restlichen 6Byte Nutzdaten in den Byte 4 bis 9 und die Restlänge des Telegramms (also "6") in Byte 2/3 an den CP 240 senden. Dieser speichert die Nutzdaten im Sendepuffer ab und schreibt den Wert "6" auf Byte 2/3 an die CPU zurück.

Beim Empfang der "6" auf Byte 2/3 sendet die CPU eine "0" auf Byte 2/3. Der CP 240 stößt daraufhin das Senden des Telegramms über die serielle Schnittstelle an und schreibt, wenn alle Daten übertragen sind, eine "0" auf Byte 2/3 zurück.

Beim Empfang der "0" kann die CPU ein neues Telegramm an den CP 240 senden.

**Beispiel Daten  
empfangen**

Die Schnittstelle des CP 240 hat z.B. ein Telegramm mit 18Byte Länge über die serielle Schnittstelle empfangen. Der CP 240 schreibt die ersten 12Byte Nutzdaten in die Bytes 4 bis 15 des Empfangspuffers und in Byte 0/1 die Länge des Telegramms (also "18"). Die Daten werden über den Rückwandbus an die CPU übertragen. Die CPU speichert die 12Byte Nutzdaten und sendet den Wert "18" auf Byte 0/1 an den CP 240 zurück.

Beim Empfang der "18", schreibt der CP 240 die restlichen 6Byte Nutzdaten in die Byte 4 bis 9 des Empfangspuffers und in Byte 0/1 die Länge ("6") der übergebenen Nutzdaten. Die CPU speichert die Nutzdaten und gibt an den CP 240 in Byte 0/1 den Wert "6" zurück.

Beim Empfang der "6" sendet der CP 240 den Wert "0" auf Byte 0/1, für Telegramm komplett, an die CPU zurück. Die CPU sendet eine "0" auf Byte 0/1 an den CP 240 zurück.

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

## Einsatz unter Modbus

### Übersicht

Sie können den CP 240 Modbus im Master- oder im Slave-Modus betreiben. In beiden Modi belegt das Modul für Ein- und Ausgangs-Daten je 16Byte an beliebiger Stelle in der CPU.

Die Kommunikation im Master-Modus erfolgt über Datenbausteine unter Einsatz der SEND-RECEIVE-Hantierungsbausteine. Hier können unter Einsatz einer Blockung bis zu 250Byte Nutzdaten übertragen werden.

Im Slave-Modus ist die Anzahl der Nutzdaten für Ein- und Ausgabe auf 16Byte begrenzt. Für den Einsatz des Slave sind keine Hantierungsbausteine erforderlich. Dieser ist lediglich zu parametrieren.

Die Parametrierung erfolgt in der Hardware-Konfiguration der CPU unter Verwendung der mitgelieferten GSD.

Nach jedem Hochlauf der CPU übergibt diese die Parameter an das Modbus-Modul. Sind die Parameter gültig, werden diese übernommen. Das Modbus-Modul befindet sich nun in Kommunikationsbereitschaft.

### Voraussetzung für den Betrieb

Folgende Komponenten sind zum Einsatz der System 200V Modbus-Module erforderlich:

- VIPA 240-1BA10 bzw. VIPA 240-1CA10 (CP 240 Modbus)
- VIPA CPU 21x (programmierbar in Step<sup>®</sup>7 von Siemens) oder VIPA CPU 24x (programmierbar in Step<sup>®</sup>5 von Siemens).
- Projektierool Step<sup>®</sup>7 Manager (V5.1) von Siemens oder VIPA MC5
- Programmierkabel für MPI-Kopplung (Green Cable von VIPA)
- GSD-Datei **VIPA04D5.gsd** (V1.31 oder höher)
- VIPA Hantierungsbausteine Fx000002\_V109.zip
- Modbus-Kabel

### Installation

Die Installation der Modbus-Moduls erfolgt nach folgender Vorgehensweise:

1. Installieren Sie die GSD-Datei **VIPA04D5.gsd** in Ihrem Projektierool.
2. Installieren Sie die VIPA Library **Fx000002\_V109.zip** mit den Hantierungsbausteinen in Ihrem Projektierool.
3. Legen Sie die Parameter für den CP 240 fest und übertragen Sie die Projektierung in die SPS.
4. Schließen Sie das Modbus-Kabel an.

### Projektierung über GSD

Die Adresszuordnung und die Parametrierung des Modbus-Moduls erfolgt im STEP<sup>®</sup>7 Manager von Siemens 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 STEP<sup>®</sup>7 Manager von Siemens jederzeit gegeben ist. Ihr Projekt übertragen Sie über MPI in die CPU.

**GSD einbinden**

Folgende Schritte sind zur Installation der GSD erforderlich:

- Kopieren Sie die mitgelieferte VIPA GSD-Datei **VIPA04D5.GSD** (V. 1.31 oder höher) in Ihr GSD-Verzeichnis ... \siemens\step7\s7data\gsd
- Starten Sie den Hardware-Konfigurator von Siemens
- Schließen Sie alle Projekte
- Gehen Sie auf **Extras** > *Neue GSD-Datei installieren*
- Geben Sie hier **VIPA04D5.gsd** an

Die Module des System 200V von VIPA sind jetzt im Hardwarekatalog integriert und können projiziert werden.

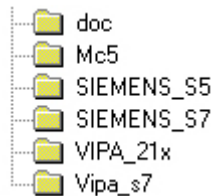
**VIPA-Bibliothek einbinden**

Die VIPA spezifischen SFCs befinden sich in Form von Bibliotheken im Lieferumfang der CPUs und der CPs. Die Bibliotheken liegen als selbstentpackende exe-Datei vor.

Sobald Sie VIPA spezifische SFCs verwenden möchten, sind diese in Ihr Projekt zu importieren.

Folgende Schritte sind hierzu erforderlich:

- Zum Entpacken der Datei FX00000z\_Vxxx.exe ausführen:  
Durch einen Doppelklick auf die Datei FX00000z\_Vxxx.exe starten Sie das integrierte Entpackprogramm. Geben Sie ein Zielverzeichnis an, in das die Dateien zu entpacken sind und wählen Sie <Extrahieren> an.
- Bibliothek "dearchivieren" :  
Zur Dearchivierung Ihrer SFC-Bibliothek starten Sie den STEP®7-Manager von Siemens. Über **Datei** > *Dearchivieren* öffnen Sie ein Dialogfenster zur Auswahl des Archivs. Sie finden die SFC-Bibliothek in der Verzeichnis-Struktur unter VIPA\_S7. Die Datei lautet VIPA.ZIP.



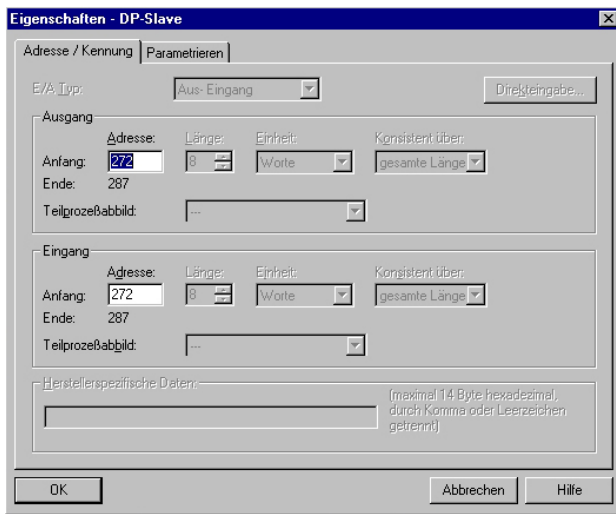
Wählen Sie VIPA.ZIP an und klicken Sie auf [Öffnen].

- Geben Sie ein Zielverzeichnis an, in dem die Bausteine abzulegen sind. Mit [OK] startet der Entpackvorgang.
- Bibliothek öffnen und SFCs in Projekt übertragen:  
Öffnen Sie die Bibliothek nach dem Entpackvorgang. Öffnen Sie Ihr Projekt und kopieren Sie die erforderlichen SFCs aus der Bibliothek in das Verzeichnis "Bibliothek" Ihres Projekts. Nun haben Sie in Ihrem Anwenderprogramm Zugriff auf die VIPA spezifischen Bausteine.

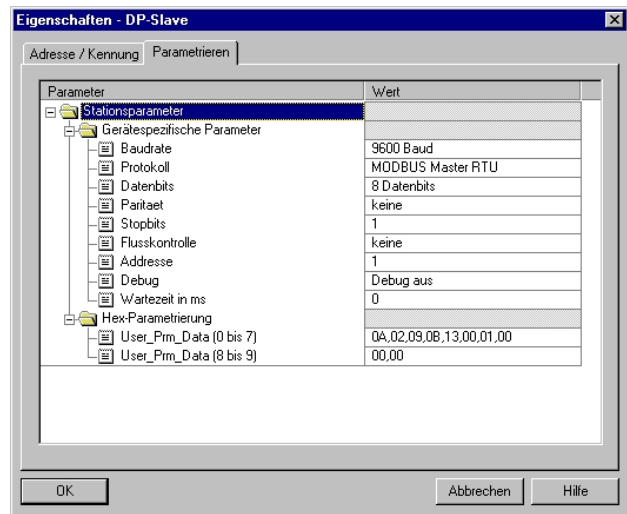
**Parametrierung**

Die Parametrierung führen Sie in der Hardwarekonfiguration durch. Folgende Parameter stehen Ihnen zur Verfügung:

Dialog für Adress-Eingabe



Dialog für Modbus-Parameter



Eine nähere Beschreibung der Parameter finden Sie weiter oben in diesem Kapitel.

**Master-Modus**

Den Master-Modus geben sie in der Parametrierung vor.

Mittels der SEND und RECEIVE-Bausteine von VIPA können Sie beliebige Modbus-Telegramme senden bzw. empfangen. Der Telegrammaufbau ist im ASCII- und RTU-Modus für den Anwender derselbe, da Telegrammstart/-ende, Checksumme und Protokollüberwachung vom Modul übernommen werden.

**Slave-Modus**

Den Slave-Modus geben sie in der Parametrierung vor. Im Slave-Modus arbeitet das Modul als intelligenter Slave, der max. 16Byte Ein- und Ausgangsdaten verarbeiten kann.

Bei Einsatz als Slave ist diesem über die Parametrierung eine Slave-Adresse zuzuteilen.

**Modbus-Slave-Funktions-Codes**

Folgende Funktions-Codes kann der CP 240 Modbus von VIPA zur Zeit im Slave-Modus verarbeiten:

Code	Befehl	Beschreibung
01h, 02h	Read n Bits	n Bit lesen
03h, 04h	Read n Words Status	n Worte lesen
05h	Write one Bit	ein Ausgangsbit schreiben
06h	Write one Word	ein Ausgangswort schreiben
0Fh	Write n Bits	n Bit schreiben
10h	Write n Words	n Worte schreiben



**Inbetriebnahme**

Nach Einschalten der Spannungsversorgung leuchten am Modbus-Modul die LEDs ER, TxD und RxD. Das Modul signalisiert hiermit, dass es noch keine gültigen Parameter von der CPU erhalten hat. Sobald Sie die CPU in RUN schalten, werden die Modbus-Parameter an das Modul übertragen. Bei gültigen Parametern erlöschen die LEDs ER, TxD und RxD. Das Modbus-Modul ist nun bereit für die Kommunikation.

Bei Einsatz im Master-Modus können Sie nun entsprechende Schreib-/Lesebefehle in Ihrem Anwenderprogramm ausführen lassen.

Sollte die ER-LED nicht erlöschen, liegt ein interner Fehler vor. Bei einem vorübergehenden Fehler können Sie diesen durch einen STOP-RUN-Übergang der CPU rücksetzen.

**Zugriff auf mehrere Slaves**

Bei Einsatz mehrerer Slaves können keine Buskonflikte auftreten, da der Master immer nur mit einem Slave kommunizieren kann. Der Master schickt an den über die Adresse spezifizierten Slave ein Kommando-telegramm und wartet eine gewisse Zeit, in der der Slave sein Antworttelegramm senden kann. Während des Wartens ist eine Kommunikation mit einem anderen Slave nicht möglich.

Zur Kommunikation mit mehreren Slaves ist für jeden Slave ein SEND-Datenbaustein für das Kommando-telegramm und ein RECEIVE-Datenbaustein für das Antworttelegramm erforderlich.

Eine Applikation mit mehreren Slaves würde aus entsprechend vielen Datenbausteinen mit Kommandos bestehen.

Diese werden der Reihe nach abgearbeitet:

1. Slave:        Sende Kommando-telegramm an Slave-Adresse 1.Slave  
                  Empfange Antworttelegramm von Slave-Adresse 1.Slave  
                  Werte Antworttelegramm aus
  
2. Slave:        Sende Kommando-telegramm an Slave-Adresse 2.Slave  
                  Empfange Antworttelegramm von Slave-Adresse 2.Slave  
                  Werte Antworttelegramm aus

... usw.

Eine Anforderung kann an einen bestimmten Slave gerichtet sein oder als Broadcast-Nachricht an alle Slaves gehen. Zur Kennzeichnung einer Broadcast-Nachricht wird die Slave-Adresse 0 eingetragen.

Nur Schreibaufträge dürfen als Broadcast gesendet werden.

**Hinweis!**

Nach einem Broadcast wartet der Master nicht auf ein Antworttelegramm.

## Modbus-Funktionscodes

### Übersicht

Folgende Modbus-Funktionscodes sind im Slave-Modus implementiert:

Code	Befehl	Beschreibung
01h, 02h	Read n Bits	n Bit lesen
03h, 04h	Read n Words Status	n Worte lesen
05h	Write one Bit	ein Ausgangsbit schreiben
06h	Write one Word	ein Ausgangswort schreiben
0Fh	Write n Bits	n Bit schreiben
10h	Write n Words	n Worte schreiben



### Hinweis!

Die aufgezeigten Prüfsummen CRC bei RTU- und LRC bei ASCII-Modus werden automatisch an jedes Telegramm angehängt. Sie werden nicht im Datenbaustein angezeigt.

Für die Byte-Reihenfolge im Wort gilt immer:

1 Wort	
High Byte	Low Byte

### Read n Bits 01h, 02h

Die Funktion ermöglicht das bitweise Lesen aus einem Slave.

#### Kommandotelegramm

Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Wort	1 Wort	1 Wort

#### Antworttelegramm

Slave-Adresse	Funktions-Code	Anzahl der gelesenen Bytes	Daten 1. Byte	Daten 2. Byte	...	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Byte	1 Byte	1 Byte		1 Wort
				max. 250 Byte		

**Read n Words  
03h, 04h**

Diese Funktion ermöglicht das wortweise Lesen aus einem Slave.

## Kommandotelegramm

Slave-Adresse	Funktions-Code	Adresse Bit	Anzahl der Worte	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Wort	1 Wort	1 Wort

## Antworttelegramm

Slave-Adresse	Funktions-Code	Anzahl der gelesenen Bytes	Daten 1. Wort	Daten 2. Wort	...	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Byte	1 Wort	1 Wort		1 Wort
				max. 125 Worte		

**Write a Bit  
05h**

Mit dieser Funktion können Sie ein Bit in Ihrem Slave ändern. Eine Zustandsänderung erfolgt unter "Zustand Bit" mit folgenden Werten:

"Zustand Bit" = 0000h → Bit = 0

"Zustand Bit" = FF00h → Bit = 1

## Kommandotelegramm

Slave-Adresse	Funktions-Code	Adresse Bit	Zustand Bit	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Wort	1 Wort	1 Wort

## Antworttelegramm

Slave-Adresse	Funktions-Code	Adresse Bit	Zustand Bit	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Wort	1 Wort	1 Wort

**Write a word  
06h**

Diese Funktion schickt ein Wort an den Slave. Hiermit können Sie im Slave ein Register überschreiben.

## Kommandotelegramm

Slave-Adresse	Funktions-Code	Adresse Wort	Wert Wort	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Wort	1 Wort	1 Wort

## Antworttelegramm

Slave-Adresse	Funktions-Code	Adresse Wort	Wert Wort	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Wort	1 Wort	1 Wort

**Write n Bits 0Fh** Diese Funktion schreibt n Bits an den Slave. Bitte beachten Sie, dass die Anzahl der Bits zusätzlich in Byte anzugeben sind.

Kommandotelegramm

Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits	Anzahl der Bytes	Daten 1. Byte	Daten 2. Byte	...	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Wort	1 Wort	1 Byte	1 Byte	1 Byte	1 Byte	1 Wort
						max. 250 Byte		

Antworttelegramm

Slave-Adresse	Funktions-Code	Adresse 1. Bit	Anzahl der Bits	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Wort	1 Wort	1 Wort

**Write n Words 10h** Über diese Funktion können Sie n Worte an den Slave schicken.

Kommandotelegramm

Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl der Worte	Anzahl der Bytes	Daten 1. Wort	Daten 2. Wort	...	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Wort	1 Wort	1 Byte	1 Wort	1 Wort	1 Wort	1 Wort
						max. 125 Worte		

Antworttelegramm

Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl der Worte	Prüfsumme CRC/LRC
1 Byte	1 Byte	1 Wort	1 Wort	1 Wort

## Beispiel zum Einsatz unter Modbus

### Übersicht

Nachdem Sie die Installation durchgeführt haben, können Sie das hier vorbereitete Beispielprojekt für erste Schritte nutzen.

Bei Bedarf können Sie das Beispielprojekt von VIPA beziehen.

In dem Beispiel wird eine Kommunikation zwischen einem Master und einem Slave über Modbus aufgebaut. Weiter soll das Beispiel zeigen, wie Sie unter Einsatz der Hantierungsbausteine auf einfache Weise die Kontrolle über die Kommunikationsvorgänge haben.

### Voraussetzung

Folgende Komponenten sind für das Beispiel erforderlich:

- 1 CPU 21x mit CP 240 Modbus als Mastersystem
- Projektierool STEP<sup>®</sup>7 Manager von Siemens mit Übertragungskabel
- 1 Modbus-Slave (wie CPU 21x und CP 240 Modbus)
- Modbus-Kabel-Verbindung

### Vorgehensweise

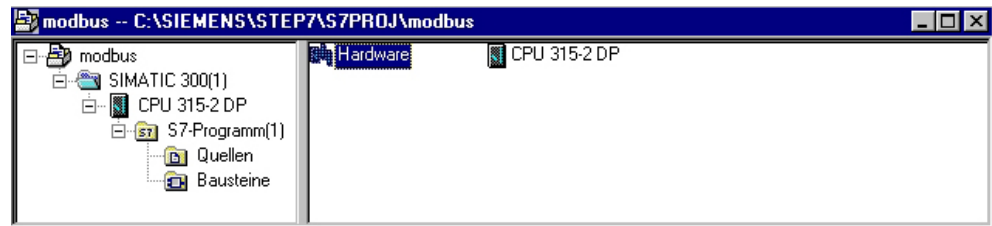
- Bauen Sie ein Modbus-System bestehend aus Master-, Slave-System und Modbus-Kabel auf.
- Projektieren Sie die Master-Seite! Öffnen Sie hierzu das Beispielprojekt in Ihrem Projektierool. Passen Sie die Übertragungsparameter entsprechend an.  
Stellen Sie unter *Protokoll* "Modbus Master RTU" ein.  
Editieren Sie den OB1 und gleichen Sie ggf. die Modul-Adressen mit den Adressen der Parametrierung ab.  
Übertragen Sie Ihr Projekt in die CPU 21x.
- Projektieren Sie die Slave Seite. Öffnen Sie hierzu das Beispielprojekt in Ihrem Projektierool. Passen Sie die CP 240 Parameter entsprechend an. Stellen Sie unter *Protokoll* "Modbus Slave RTU" ein. Geben Sie unter *Adresse* eine Slave-Adresse an.  
Für die Kommunikation unter Modbus ist auf Slave-Seite kein SPS-Programm erforderlich, da Quelle und Ziel vom Master übermittelt werden.

### Projekt einspielen

Zum Einspielen in Ihr Konfigurationstool gehen Sie nach folgenden Schritten vor:

- Starten Sie den STEP<sup>®</sup>7 Manager von Siemens
- Zum Entpacken der Datei Modbus.zip gehen Sie auf **Datei** > *dearchivieren*.
- Wählen sie die Beispieldatei Modbus.zip aus und geben Sie als Zielverzeichnis "s7proj" an.
- Öffnen Sie nach dem Entpacken das Projekt.

**Projekt-Struktur** Das Projekt hat folgende Struktur:



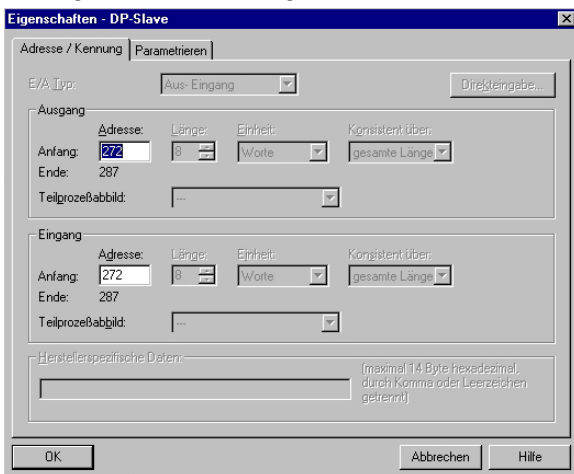
**Master-Projektierung**

Das Beispiel beinhaltet schon SPS-Programm und die Parameter für den Modbus-Master. Sie müssen lediglich die Modbus-Parameter anpassen.

**Parametrierung**

Starten Sie hierzu den Hardware-Konfigurator und wählen Sie das Modul 240-1CA10 an. Durch Doppelklick gelangen Sie in die Parametrierung:

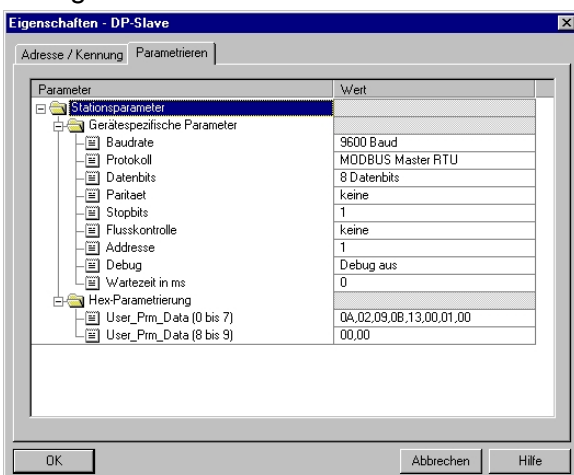
**Dialog für Adress-Eingabe**



Hier können Sie angeben, ab welcher Adresse die 16Byte für Ein- und Ausgabe in der CPU abliegen.

Bitte beachten Sie, dass Sie die Adressen, die Sie hier ändern, auch in Ihren SEND und RECEIVE-Bausteinen ändern müssen.

**Dialog für Modbus-Parameter**



In diesem Teil der Parametrierung stellen Sie die Modbus-Parameter ein.

Folgende Parameter müssen bei allen Busteilnehmern gleich sein: Baudrate, Datenbits, Parität, Stopbits und Flusskontrolle.

Stellen Sie unter *Protokoll* "Modbus Master RTU" ein

Die Angabe einer Adresse ist nur auf der Slave-Seite erforderlich.

Bei der Master-Parametrierung wird die Adresse ignoriert.

**SPS-Programm**

Die gewünschten Modbus-Befehle geben Sie über Ihr SPS-Programm vor. Im vorliegenden Beispiel wird im OB1 der Einsatz von SEND und RECEIVE gezeigt.

OB 1:

```

CALL "SEND_ACII_STX_3964"
  ADR      :=256           //Ausgangsadresse des Moduls
  _DB      :=DB10         //In diesem Datenbaustein erstellen
                          //Sie das zu sendende Telegramm
  ABD      :=W#16#0       //Ab diesem Byte-Offset beginnt
                          //das Telegramm im _DB
  ANZ      :=MW12         //Telegrammlaenge (zu sendende Laenge) in Byte
  PAFE     :=MB14         //Fehlerbyte
  FRG      :=M1.0        //Sendeanstoss (1=Anstoss, geht auf 0
                          //wenn Senden abgeschlossen)
  GESE     :=MW16         //intern erforderlich
  ANZ_INT  :=MW18         //intern erforderlich
  ENDE_KOM :=M2.0        //intern erforderlich
  LETZTER_BLOCK:=M2.1    //intern erforderlich
  SENDEN_LAEUFT:=M2.2    //intern erforderlich
  FEHLER_KOM :=M2.3      //intern erforderlich

CALL "RECEIVE_ACII_STX_3964"
  ADR      :=256           //Eingangsadresse des Moduls
  _DB      :=DB11         //In diesem Datenbaustein wird das
                          //empfangene Telegramm abgelegt
  ABD      :=W#16#0       //Ab diesem Byte-Offset beginnt das Telegramm im _DB
  ANZ      :=MW22         //Telegrammlaenge (empfangene Laenge) in Byte
  EMFR     :=M1.1        //Empfangsstatus (1=Telegramm komplett empfangen)
  PAFE     :=MB34         //Fehlerbyte
  GEEM     :=MW36         //intern erforderlich
  ANZ_INT  :=MW38         //intern erforderlich
  EMPF_LAEUFT :=M3.0     //intern erforderlich
  LETZTER_BLOCK:=M3.1    //intern erforderlich
  FEHL_EMPF :=M3.2       //intern erforderlich

U      M      1.1        //solange Empfangsstatus=1 ist wird kein neues
                          //Telegramm eingetragen
R      M      1.1        //daher muss der Empfangsstatus mit 0 quittiert werden

```

Passen Sie noch ggf. die Adressen, die der CP in der CPU belegt, an die Adressen in Ihrer Parametrierung an und übertragen Sie die Hardware-Konfiguration in Ihre CPU 21x des Master-Systems.

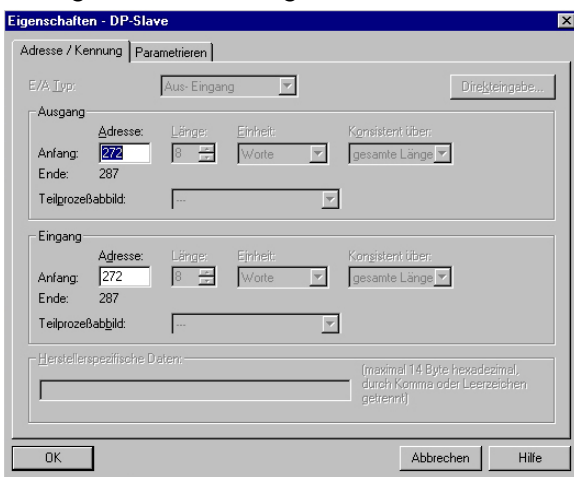
**Slave-Projektierung**

Für die Projektierung des Slave sind nur die Modbus-Parameter anzupassen. Ein SPS-Programm ist nicht erforderlich, da die Quell- und Zieldaten im Master-Telegramm mitgeliefert werden.

**Parametrierung**

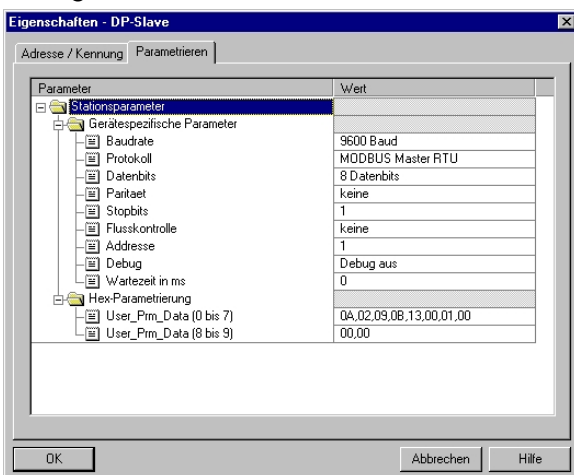
Zur Parametrierung des Slave-Moduls öffnen Sie das Beispielprojekt in Ihrem Hardware-Konfigurator. Wählen Sie das Modul 240-1CA10 an. Durch Doppelklick gelangen Sie in die Parametrierung.

**Dialog für Adress-Eingabe**



Hier können Sie angeben, ab welcher Adresse die 16Byte für Ein- und Ausgabe in der CPU abliegen.

**Dialog für Modbus-Parameter**



In diesem Teil der Parametrierung stellen Sie die Modbus-Parameter ein.

Folgende Parameter müssen bei allen Busteilnehmern gleich sein:

Baudrate, Datenbits, Parität, Stopbits und Flusskontrolle.

Geben Sie unter *Adresse* für den Slave eine gültige Modbus-Adresse an.

Übertragen Sie die Parametrierung in Ihre CPU des Slave-Systems.



**Telegramme senden und empfangen**

Öffnen Sie die Variablen-tabelle **Tabelle1** des Beispielprojekts und gehen Sie online.

	Operand	Anzei	Statuswert	Steuerwert
1	PEW 256	HEX	W#16#0000	
2	PEW 258	HEX	W#16#0000	
3	MW 12	DEZ	23	23
4	M 1.0	BOOL	false	true
5	MB 2	BIN	2#0000_0000	2#0000_0000
6	MW 22	DEZ	6	
7				
8	DB10.DBD 0	HEX	DW#16#05100000	DW#16#05100000
9	DB10.DBD 4	HEX	DW#16#000810A0	DW#16#000810A0
10	DB10.DBD 8	HEX	DW#16#A1A2A3A4	DW#16#A1A2A3A4
11	DB10.DBD 12	HEX	DW#16#A5A6A7A8	DW#16#A5A6A7A8
12	DB10.DBD 16	HEX	DW#16#A9AABAC	DW#16#A9AABAC
13	DB10.DBD 20	HEX	DW#16#ADAEAF00	DW#16#ADAEAF00
14				
15	DB11.DBD 0	HEX	DW#16#05100000	DW#16#00000000
16	DB11.DBD 4	HEX	DW#16#000810A0	DW#16#00000000
17	DB11.DBD 8	HEX	DW#16#00000000	DW#16#00000000
18	DB11.DBD 12	HEX	DW#16#00000000	DW#16#00000000
19	DB11.DBD 16	HEX	DW#16#00000000	DW#16#00000000
20				

**Sende-Baustein DB10**

DB10.DBD 0	<b>DW#16#05100000</b> mit 05 → 10 → 0000 →	<b>Kommandotelegramm</b> Slave-Adresse 05h Funktionscode 10h (write n words) Offset 0000h
DB10.DBD 4	<b>DW#16#000810A0</b> mit 0008 → 10 → A0 →	<b>Kommandotelegramm + 1 Datenbyte</b> Wordcount 0008h Bytecount 10h Beginn 16 Byte Daten mit A0h
DB10.DBD 8	<b>DW#16#A1A2A3A4</b>	<b>Daten Byte 2 ... 5</b>
DB10.DBD 12	<b>DW#16#A5A6A7A8</b>	<b>Daten Byte 6 ... 9</b>
DB10.DBD 16	<b>DW#16#A9AABAC</b>	<b>Daten Byte 10 ... 13</b>
DB10.DBD 20	<b>DW#16#ADAEAF00</b> mit ADAEAF → 00 →	<b>Daten Byte 14 ... 16 + 1 Byte nicht ben.</b> Daten Byte 14 ... 16 vom Modul nicht mehr belegt

**Empfangs-Baustein DB11**

DB11.DBD 0	<b>DW#16#05100000</b> mit 05 → 10 → 0000 →	<b>Antworttelegramm</b> Slave-Adresse 05h Funktionscode 10h (kein Fehler) Offset 0000h
DB11.DBD 4	<b>DW#16#000810A0</b> mit 0008 → 10 → 00 →	<b>Antworttelegramm + 1 Datenbyte</b> Wordcount 0008h Bytecount 10h Beginn 16 Byte Daten mit 00h (bei Schreibbefehl irrelevant)
DB11.DBD 8	<b>DW#16#00000000</b>	<b>Daten Byte 2 ... 5</b>
DB11.DBD 12	<b>DW#16#00000000</b>	<b>Daten Byte 6 ... 9</b>
DB11.DBD 16	<b>DW#16#00000000</b>	<b>Daten Byte 10 ... 13</b>
DB11.DBD 20	<b>DW#16#00000000</b> mit ADAEAF → 00 →	<b>Daten Byte 14 ... 16 + 1 Byte nicht ben.</b> Daten Byte 14 ... 16 vom Modul nicht mehr belegt

**Empfangs-Baustein mit Fehlerrückmeldung**

Bei der Kommunikation unter Modbus gibt es 2 Fehlerarten:

- Slave antwortet nicht auf Kommando von Master

Antwortet der Slave nicht innerhalb der vorgegebenen Time-out-Zeit, trägt der Master im Empfangs-Baustein folgende Fehlermeldung ein:

ERROR01 NO\_DATA. In der Hex-Darstellung werden folgende Werte eingetragen:

DB11.DBD 0	<b>DW#16#4552524F</b> mit 45 → 52 → 52 → 4F →	<b>Antworttelegramm</b> 45h: E 52h: R 52h: R → 4Fh: O
DB11.DBD 4	<b>DW#16#52000120</b> mit 52 → 0001 → 20 →	<b>Antworttelegramm</b> 52h: R 0001h:1 (als Wort) 52h: " "
DB11.DBD 8	<b>DW#16#4E4F2044</b> mit 4E → 4F → 20 → 44 →	<b>Antworttelegramm</b> 45h: N 52h: O 52h: " " 4Fh: D
DB11.DBD 12	<b>DW#16#41544100</b> mit 41 → 54 → 41 → 00 →	<b>Antworttelegramm</b> 45h: A 52h: T 52h: A 00h: leer

•  
•  
•

- Slave antwortet mit einer Fehlermeldung

Liefert der Slave einen Fehler zurück, so wird der Funktionscode mit 80h "verodert" zurückgesendet.

DB11.DBD 0	<b>DW#16#05900000</b> mit 05 → 90 → 0000 →	<b>Antworttelegramm</b> Slave-Adresse 05h Funktionscode 90h (Fehlermeldung da 10h OR 80h = 90h) Die Restdaten sind irrelevant da Fehler rückgemeldet wurde
------------	---	---

## Kommunikation über Standardhantierungsbausteine

Die Kommunikation wird mittels der mitgelieferten Hantierungsbausteine abgewickelt.



### Hinweis!

Für Baugruppen mit Firmwarestand ab V1.06 ist der Hantierungsbaustein SYNCHRON im Programm mit aufzurufen. Bevor der Hantierungsbaustein nicht durchlaufen wurde, ist eine Kommunikation mit dem CP 240 nicht möglich, da er eine Synchronkennung im Schreib- und Lesezeiger ausgibt und diese von der CPU zu quittieren ist.

Unter Modbus ist der SYNCHRON-Baustein nicht erforderlich.

Folgende Bausteine befinden sich im Lieferumfang:

#### für CPU 24x

Name	FBs	Kurzbeschreibung
SCP240	FB3	Sende-Baustein für ASCII, STX/ETX, 3964(R) und Modbus
RCP240	FB4	Receive-Baustein für ASCII, STX/ETX, 3964(R) und Modbus
FETCH	FB20	Fetch-Baustein nur für RK512
SEND	FB22	SEND nur für RK512
S/R_ALL	FB23	SEND - RECEIVE_ALL nur für RK512
SYNCHRON	FB25	Synchronisation des CP 240 Ab Firmwarestand V1.06 (bei Modbus nicht erforderlich)

#### für CPU 21x

Name	FCs	Kurzbeschreibung
SEND_ASCII_STX_3964	FC0	Sende-Baustein für ASCII, STX/ETX, 3964(R) und Modbus
RECEIVE_ASCII_3964	FC1	Receive-Baustein für ASCII, STX/ETX, 3964(R) und Modbus
FETCH_RK512	FC2	Fetch-Baustein nur für RK512
SEND_RK512	FC3	Send nur für RK512
S/R_ALL_RK512	FC4	Send - Receive ALL nur für RK512
SYNCHRON_RESET	FC9	Synchronisation des CP 240 Ab Firmwarestand V1.06 (bei Modbus nicht erforderlich)

## Standardhantierungsbausteine für CPU 24x

### Kommunikation unter ASCII, STX/ETX oder 3964(R)

Die Kommunikation unter diesen Prozeduren verläuft aktiv, d.h. beide Partner müssen sich aktiv um das Senden und Empfangen der Daten kümmern. Beim Senden wird das Ziel nicht mitgereicht und beim Empfangen nicht der Sendeort.

Zulässige Befehle unter ASCII, STX/ETX oder 3964(R):

SCP240	FB3	(Daten senden)
RCP240	FB4	(Daten empfangen)

### Kommunikation unter RK512

Bei Einsatz der Rechnerkopplung RK512 wird das Master-Slave-Verfahren angewendet. Der Master kann über FETCH Daten vom "Slave"-Partner anfordern und mit SEND Daten an den Partner schicken. Hierbei werden im Telegrammkopf Ziel bzw. Quelle mitgeliefert.

Damit der Slave auf Master-Aufgaben reagieren kann, ist im Slave ein S/R\_ALL (Send/Receive\_All) zyklisch auszuführen.

Zulässige Befehle unter RK512:

FETCH	FB20	(Daten anfordern)
SEND	FB22	(Daten senden)
S/R_ALL	FB23	(Reaktion auf Anforderung im Slave)

### Kommunikation unter Modbus

Abhängig von der ausgewählten Betriebsart können Sie den Modbus-CP im Master oder im Slave-Modus betreiben.

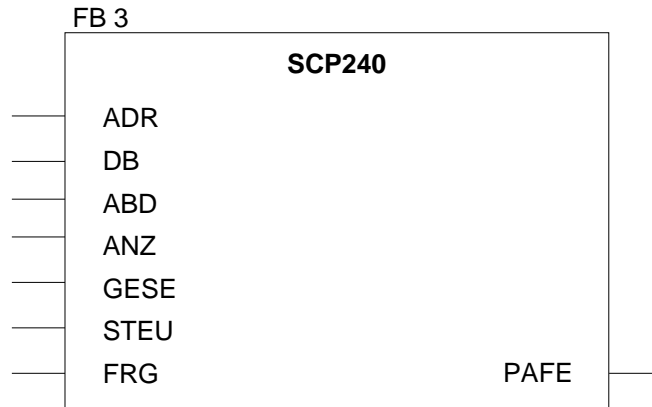
Im Slave-Betrieb sind keine Hantierungsbausteine erforderlich. Im Master-Modus erfolgt die Kommunikation mit dem Slave über SEND- und RECEIVE-Anweisungen. In beiden Fällen belegt der CP in der CPU einen 16Byte großen Bereich für Ein- und Ausgabe.

Zulässige Befehle unter Modbus:

SCP240	FB3	(Daten senden)
RCP240	FB4	(Daten empfangen)

**SCP240  
SEND (FB3)**

Dieser FB dient zur Datenausgabe an ein Peripheriegerät unter ASCII, STX/ETX, 3964(R) und Modbus.



- ADR** Peripherieadresse unter der das CP 240-Modul angesprochen wird. Im DB1 bestimmen Sie die Peripherieadresse unter der der CP 240 in das System einzubinden ist. Die Adresse liegt im Bereich PY000 ... PY240. Näheres hierzu finden Sie im CPU 24x-Handbuch HB99 von VIPA.
- DB** Nummer des Datenbausteins, der die zu sendenden Daten enthält.
- ABD** Wort-Variable, die die Nummer des Datenwortes enthält, ab dem die auszugebenden Zeichen abgelegt sind.
- ANZ** Wort-Variable, die die Anzahl der Bytes enthält, die übertragen werden sollen.
- GESE** Diese Variable wird intern verwendet und steuert den Sendevorgang. Geben Sie hier ein Merkerwort an, in dem der Hantierungsbaustein seine internen Daten ablegen kann.

**STEU** Hier legt der Hantierungsbaustein interne Steuerbits ab. Geben Sie ein Merkerbyte an, in dem der Hantierungsbaustein seine Steuerbits ablegen kann.

**FRG** Wenn dieses Merker-Bit auf "1" gesetzt ist, wird die in ANZ angegebene Anzahl von Daten einmalig ausgegeben. Nach der Übertragung wird das Bit wieder auf "0" gesetzt. Ist dieses Bit beim Aufruf des FB 3 auf "0" gesetzt, 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:

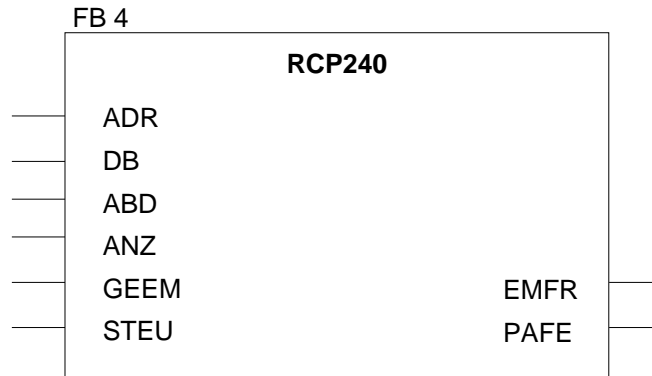
1 = Datenbaustein nicht vorhanden

2 = Datenbaustein zu kurz

3 = Datenbausteinnummer nicht im gültigen Bereich

**RCP240  
RECEIVE (FB4)**

Dieser FB dient zum Datenempfang von einem Peripheriegerät unter ASCII, STX/ETX, 3964(R) und Modbus.



- ADR** Peripherieadresse unter der das CP 240-Modul angesprochen wird. Im DB1 bestimmen Sie die Peripherieadresse unter welcher der CP 240 in das System einzubinden ist. Die Adresse liegt im Bereich PY000 ... PY240. Näheres hierzu finden Sie im CPU 24x-Handbuch HB99 von VIPA.
- DB** Nummer des Datenbausteins, in den die empfangenen Daten abzulegen sind.
- ABD** Wort-Variable, die die Nummer des Datenwortes enthält, ab dem die empfangenen Zeichen abzulegen sind.
- ANZ** Wort-Variable, die die Anzahl der Bytes enthält, die übertragen werden sollen.
- GEEM** Diese Variable wird intern verwendet und steuert den Receive-Vorgang. Geben Sie hier ein Merkwort an, in dem der Hantierungsbaustein seine internen Daten ablegen kann.

**STEU** Hier legt der Hantierungsbaustein interne Steuerbits ab. Geben Sie ein Merkerbyte an, in dem der Hantierungsbaustein seine Steuerbits ablegen kann.

**EMFR** Sobald ein Telegramm komplett empfangen wurde und im Empfangs-DB abgelegt ist, wird das Merker-Bit EMFR (Empfangen fertig) gesetzt. Dieses Bit wird nicht automatisch zurückgesetzt.

**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



## **FETCH** Daten anfordern über RK512 (FB20)

Dieser FB dient zur Datenanforderung von einem Peripheriegerät unter RK512. Dieser FB ist nur unter RK512 zulässig.



### **Hinweis!**

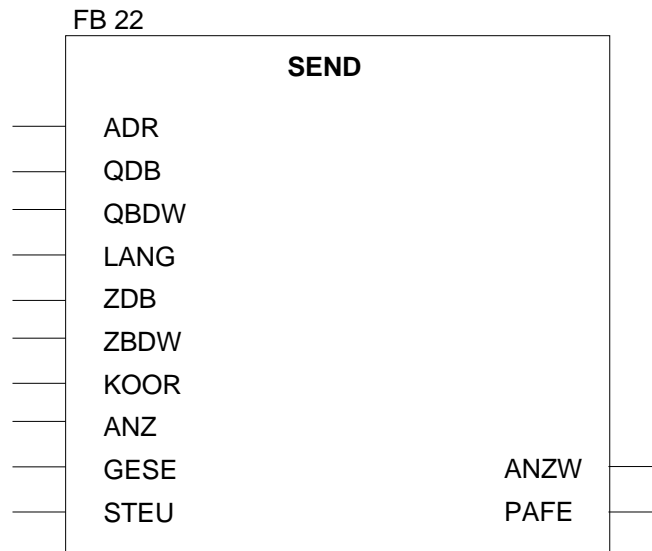
Damit der Slave auf die Master-Anforderung reagieren kann, ist im Slave ein S/R\_ALL (Send/Receive\_All) zyklisch auszuführen.

- ADR** Peripherieadresse unter der das CP 240-Modul angesprochen wird. Im DB1 bestimmen Sie die Peripherieadresse unter der der CP 240 in das System einzubinden ist. Die Adresse liegt im Bereich PY000 ... PY240. Näheres hierzu finden Sie im CPU 24x-Handbuch HB99 von VIPA.
- QDB** Nummer des Datenbausteins, der die zu sendenden Daten enthält.
- QBDW** Wort-Variable, die die Nummer des Datenwortes enthält, ab dem die auszugebenden Zeichen abgelegt sind.
- LANG** Wort-Variable, die die Anzahl der Bytes enthält, die übertragen werden sollen.
- ZDB** Nummer des Datenbausteins, in den die gesendeten Daten abzulegen sind.

<b>ZBDW</b>	Wort-Variable, die die Nummer des Datenwortes enthält, ab dem die empfangenen Zeichen abzulegen sind.
<b>KOOR</b>	<p>Hier wird die Verwendung eines Koordinierungsmerkers parametrierbar. Im High-Byte muss die Byte-Nummer, im Low-Byte muss die Bit-Nummer des Koordinierungsmerkers parametrierbar werden. Soll kein Koordinierungsmerker verwendet werden, so ist in High- und Low-Byte der Wert 255 vorzugeben.</p> <p>Mit dem Koordinierungsmerker wird der Zugriff auf den Quellbereich geregelt:</p> <p>Durch Setzen des Merkers können Sie im Partner-AG Ihre gesendeten Daten vor dem Überschreiben schützen. Durch Rücksetzen des Merkers können die Daten wieder überschrieben werden.</p>
<b>ANZW</b>	Anzeigenwort. Das Anzeigenwort belegt ein Merker-Wort. Im rechten Byte werden Statusbits abgelegt. Im linken Byte wird, wenn im rechten Byte die Kennung "Fertig mit Fehler" steht, eine Fehlernummer abgelegt.
<b>STEU</b>	Hier legt der Hantierungsbaustein interne Steuerbits ab. Geben Sie ein Merkerbyte an, in dem der Hantierungsbaustein seine Steuerbits ablegen kann.
<b>ANZ</b>	Wort-Variable, die die Anzahl der Bytes enthält, die übertragen werden sollen.
<b>GESE</b>	Diese Variable wird intern verwendet und steuert den Sendevorgang. Geben Sie hier ein Merkerwort an, in dem der Hantierungsbaustein seine internen Daten ablegen kann.
<b>PAFE</b>	<p>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.</p> <p>Folgende Fehler sind möglich:</p> <ul style="list-style-type: none"><li>1 = Datenbaustein nicht vorhanden</li><li>2 = Datenbaustein zu kurz</li><li>3 = Datenbausteinnummer nicht im gültigen Bereich</li></ul>

**SEND**  
**Daten senden über**  
**RK512 (FB22)**

Dieser FB dient zur Datenausgabe an ein Peripheriegerät unter RK512.  
 Dieser FB ist nur unter RK512 zulässig.



**Hinweis!**

Damit der Slave die vom Master gesendeten Daten verarbeiten kann, ist im Slave ein S/R\_ALL (Send/Receive\_All) zyklisch auszuführen.

**ADR** Peripherieadresse unter der das CP 240-Modul angesprochen wird. Im DB1 bestimmen Sie die Peripherieadresse unter welcher der CP 240 in das System einzubinden ist. Die Adresse liegt im Bereich PY000 ... PY240. Näheres hierzu finden Sie im CPU 24x-Handbuch HB99 von VIPA.

**QDB** Nummer des Datenbausteins, der die zu sendenden Daten enthält.

**QBDW** Wort-Variable, die die Nummer des Datenwortes enthält, ab dem die auszugebenden Zeichen abgelegt sind.

**LANG** Wort-Variable, die die Anzahl der Bytes enthält, die übertragen werden sollen.

**ZDB** Nummer des Datenbausteins, in den die gesendeten Daten abzulegen sind.

**ZBDW** Wort-Variable, die die Nummer des Datenwortes enthält, ab dem die empfangenen Zeichen abzulegen sind.

**KOOR** Hier wird die Verwendung eines Koordinierungsmerkers parametrieren. Im High-Byte muss die Byte-Nummer, im Low-Byte muss die Bit-Nummer des Koordinierungsmerkers parametrieren werden. Soll kein Koordinierungsmerker verwendet werden, so ist in High- und Low-Byte der Wert 255 vorzugeben.

Mit dem Koordinierungsmerker wird der Zugriff auf den Quellbereich geregelt:

Durch Setzen des Merkers können Sie im Partner-AG Ihre gesendeten Daten vor dem Überschreiben schützen. Durch Rücksetzen des Merkers können die Daten wieder überschrieben werden.

**STEU** Hier legt der Hantierungsbaustein interne Steuerbits ab. Geben Sie ein Merkerbyte an, in dem der Hantierungsbaustein seine Steuerbits ablegen kann.

**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

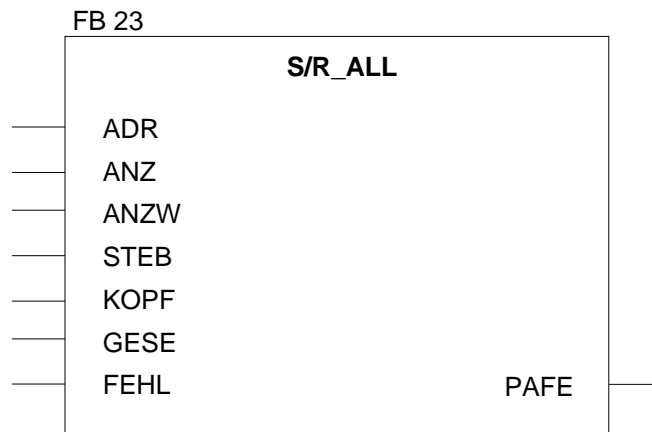
3 = Datenbausteinnummer nicht im gültigen Bereich

---

**S/R\_ALL**  
**Reaktion auf**  
**Master-Aufgaben**  
**über RK512 (FB23)**

Wird das System mit dem CP 240 als Slave eingesetzt, ist dieser FB in der Slave-CPU zyklisch aufzurufen. Nur dann kann der CP 240 auf die Master-Aufgaben reagieren. Bei FETCH werden die Daten bereitgestellt und an den Master übertragen. Die mit SEND vom Master empfangenen Daten werden übernommen, abgelegt und bestätigt.

Dieser FB ist nur unter RK512 zulässig.



- ADR** Peripherieadresse unter der das CP 240-Modul angesprochen wird. Im DB1 bestimmen Sie die Peripherieadresse unter welcher der CP 240 in das System einzubinden ist. Die Adresse liegt im Bereich PY000 ... PY240. Näheres hierzu finden Sie im CPU 24x-Handbuch HB99 von VIPA.
- ANZ** Wort-Variable, die die Anzahl der Bytes enthält, die übertragen werden sollen.
- ANZW** Anzeigenwort. Das Anzeigenwort belegt ein Wort. Im rechten Byte werden Statusbits abgelegt. Im linken Byte wird, wenn im rechten Byte die Kennung "Fertig mit Fehler" steht, eine Fehlernummer abgelegt.
- STEB** Steuerbyte, das intern verwendet wird.
- KOPF** Anfang des 10Byte großen Merkerbereichs, in dem der RK512-Telegrammkopf abgelegt wird.

**GESE** Diese Variable wird intern verwendet und steuert den Sendevorgang. Geben Sie hier ein Merkerwort an, in dem der Hantierungsbaustein seine internen Daten ablegen kann.

**FEHL** Steuerbit, das intern verwendet wird.

**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

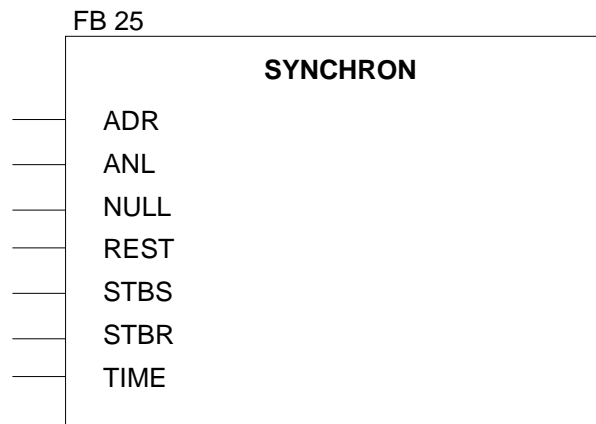
2 = Datenbaustein zu kurz

3 = Datenbausteinnummer nicht im gültigen Bereich

## **SYNCHRON** **Synchronisation** **(FB25)**

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.

Der SYNCHRON-Baustein ist bei Einsatz unter Modbus nicht erforderlich.



**ADR** Peripherieadresse unter der das CP 240-Modul angesprochen wird. Im DB1 bestimmen Sie die Peripherieadresse unter welcher der CP 240 in das System einzubinden ist. Die Adresse liegt im Bereich PY000 ... PY240. Näheres hierzu finden Sie im CPU 24x-Handbuch HB99 von VIPA.

**ANL** Ein Anlauf wurde durchgeführt.  
Über dieses Bit wird dem HTB mitgeteilt, dass an der CPU STOP/START bzw. NETZ-AUS/NETZ-EIN erfolgt ist und nun eine Synchronisation erfolgen muss.  
Nachdem die Synchronisation erfolgt ist, wird das Bit vom HTB gelöscht.

**NULL** Ein intern verwendetes Bit zur Datenübertragung mit dem CP.

**REST** Rücksetzen des CP 240.  
Durch Setzen dieses Bits kann der Anwender aus dem SPS-Programm heraus den CP 240 rücksetzen. Wird dieses Bit gesetzt, so schreibt der Hantierungsbaustein die Reset-Kennung auf den CP und wartet bis diese quittiert wird. Danach wird wie beim Anlauf verfahren.

<b>STBS</b>	Steuerbit SEND Hier ist das Merkerbyte anzugeben in dem die Steuerbits für den Sende-FB abgelegt sind.
<b>STBR</b>	Steuerbit RECEIVE Hier ist das Merkerbyte anzugeben in dem die Steuerbits für den Receive-FB abgelegt sind.
<b>TIME</b>	Timer für die Wartezeit bis die Quittierung des Reset erfolgt.



**Beispiel FB25  
SYNCHRON:**

Es ist im OB21 und OB22 das Bit auf "1" zu setzen, das am Bezeichner ANL angegeben ist. Erkennt der Baustein das ein Anlauf erfolgt ist, so quittiert er die Synchronkennung und löscht die Steuerbits der Hantierungsbausteine für SEND und RECEIVE bzw. FETCH und WRITE. Nachdem die Synchronisation abgeschlossen ist setzt der Baustein das Bit ANL auf "0".

Mit dem FB25 kann zusätzlich ein Reset des CP 240 durchgeführt werden. Dazu ist das am Bezeichner REST angegebene Bit zu setzen. Daraufhin schreibt der FB die Reset-Kennung auf den CP 240 und wartet bis diese quittiert wird.

**Parameter-  
beschreibung  
FB25**

```

BAUSTEIN#FB25
BSTNAME #SYNCHRON
BIB      #102
BEZ      #ADR    D:KF    BAUGRUPPENADRESSE
BEZ      #ANL    E:BI    ANLAUF WURDE DURCHGEFUEHRT
BEZ      #NULL   E:BI    0 NACH SZ/LZ SCHREIBEN
BEZ      #REST   E:BI    RESET AUSLOESEN
BEZ      #STBS   E:BY    STEUERBITS FUER SEND
BEZ      #STBR   E:BY    STEUERBITS FUER RECEIVE
BEZ      #TIME   T      WARTENZEIT FUER ANLAUF QUITT.

```

**Programmier-  
beispiel**

```

BAUSTEIN#OB21
BIB      #6100
00000    :
00002    :UN  M 101.0    Bit Anlauf ist 0
00004    :S   M 101.0    dann auf 1 setzen
00006    :U   M 101.1    Nullsenden ist 1
00008    :R   M 101.1    dann ruecksetzen
0000A    :
0000C    :
0000E    :BE

```

```

BAUSTEIN#OB22
BIB      #6100
00000    :
00002    :UN  M 101.0    Bit Anlauf ist 0
00004    :S   M 101.0    dann auf 1 setzen
00006    :U   M 101.1    Nullsenden ist 1
00008    :R   M 101.1    dann ruecksetzen
0000A    :
0000C    :
0000E    :BE

```

```

BAUSTEIN#FB223 (zyklisch bearbeiteter FB)
BSTNAME #P3964
BIB      #17100
0000A    :
0000C    :SPA FB 25
          NAME #SYNCHRON
          ADR  =KF +128      Baugruppen Adresse
          ANL  =M 101.0      Anlauf wurde durchgeführt
          NULL =M 101.1      Nullsenden an CP (interne
                              Auswertung)
          REST =M 101.2      Reset des CP
          STBS =MB 107       Steuerbits für Sende-FB
          STBR =MB 109       Steuerbits für Receive-FB
          TIME =T 19         Wartezeit Quittung
0001E    :
00020    :
00022    :U   M 101.0      Solange Synchron nicht
                              Quittiert
00024    :BEB              Programm Ende
00026    :                Nach Synchron kann
                              Kommunikation erfolgen
00028    :L   KB 0
0002A    :T   MW 102
0002C    :
0002E    :SPA FB 3
          NAME #SCP240
          ADR  =KF +128
          DB   =DB 10
          ABD  =MW 102
          ANZ  =MW 104
          GESE =MW 106
          STEU =MB 107
          FRG  =M 101.7
          PAFE =MB 108
00042    :
00044    :
00046    :SPA FB 4
          NAME #RCP240
          ADR  =KF +128
          DB   =DB 11
          ABD  =MW 102
          ANZ  =MW 114
          GEEM =MW 116
          STEU =MB 109
          EMFR =M 101.6
          PAFE =MB 110
0005A    :
0005C    :BE

```

## Standardhantierungsbausteine für CPU 21x

### Kommunikation unter ASCII, STX/ETX oder 3964(R)

Die Kommunikation unter diesen Prozeduren verläuft aktiv, d.h. beide Partner müssen sich aktiv um das Senden und Empfangen der Daten kümmern. Beim Senden wird das Ziel nicht mitgereicht und beim Empfangen nicht der Sendeort.

Zulässige Befehle unter ASCII, STX/ETX oder 3964(R):

SEND_ACII_STX_3964	FC0	(Daten senden)
RECEIVE_ACII_STX_3964	FC1	(Daten empfangen)

### Kommunikation unter RK512

Bei Einsatz der Rechnerkopplung RK512 wird das Master-Slave-Verfahren angewendet. Der Master kann über FETCH Daten vom "Slave"-Partner anfordern und mit SEND Daten an den Partner schicken. Hierbei werden im Telegrammkopf Ziel bzw. Quelle mitgeliefert.

Damit der Slave auf Master-Aufgaben reagieren kann, ist im Slave ein S/R\_ALL (Send/Receive\_All) zyklisch auszuführen.

Zulässige Befehle unter RK512:

FETCH_RK512	FC2	(Daten anfordern)
SEND_RK512	FC3	(Daten senden)
S/R_ALL_RK512	FC4	(Reaktion auf Anforderung im Slave)

### Kommunikation unter Modbus

Abhängig von der ausgewählten Betriebsart können Sie den Modbus-CP im Master oder im Slave-Modus betreiben.

Im Slave-Betrieb sind keine Hantierungsbausteine erforderlich. Im Master-Modus erfolgt die Kommunikation mit dem Slave über SEND- und RECEIVE-Anweisungen. In beiden Fällen belegt der CP in der CPU einen 16Byte großen Bereich für Ein- und Ausgabe.

Zulässige Befehle unter Modbus:

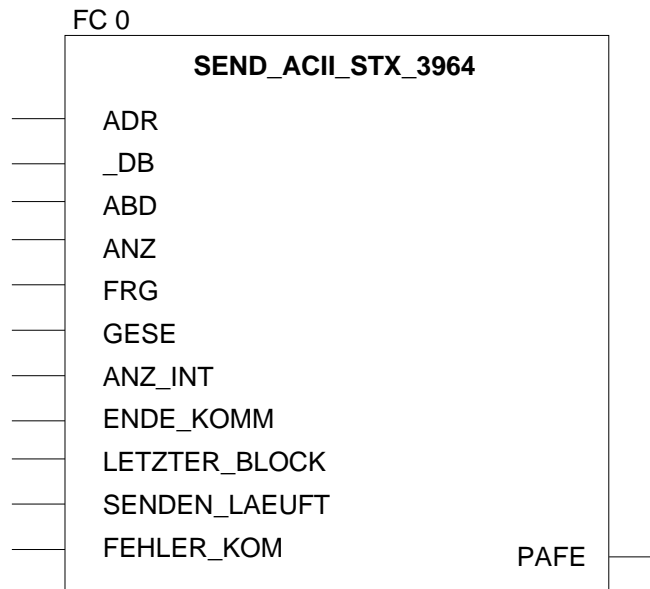
SEND_ASCII_STX_3964	FC0	(Daten senden)
RECEIVE_ASCII_STX_3964	FC1	(Daten empfangen)

**SEND\_ASCII\_  
STX\_  
3964 (FC 0)**

Dieser FC dient zur Datenausgabe an ein Peripheriegerät unter ASCII, STX/ETX., 3964(R) und Modbus.

Über die Bezeichner \_DB, ADB und ANZ wird das Sendefach festgelegt.

Über das Bit FRG wird der Sendeanstoß gesetzt und somit die Daten gesendet. Sind alle Daten gesendet, wird das Bit FRG vom HTB zurückgesetzt.



**ADR**

Int: Peripherieadresse unter der das CP 240-Modul angesprochen wird. In Ihrem Projektierool bestimmen Sie die Peripherieadresse unter der der CP 240 in das System einzubinden ist.

Die Adresse liegt im Bereich PY000 ... PY240.

**\_DB**

Block\_DB: Nummer des Datenbausteins, der die zu sendenden Daten enthält.

**ABD**

Word: Wortvariable, die die Nummer des Datenworts enthält, ab dem die auszugebenden Zeichen abgelegt sind.

**ANZ**

Word: Wortvariable, die die Anzahl der Bytes enthält, die übertragen werden sollen.

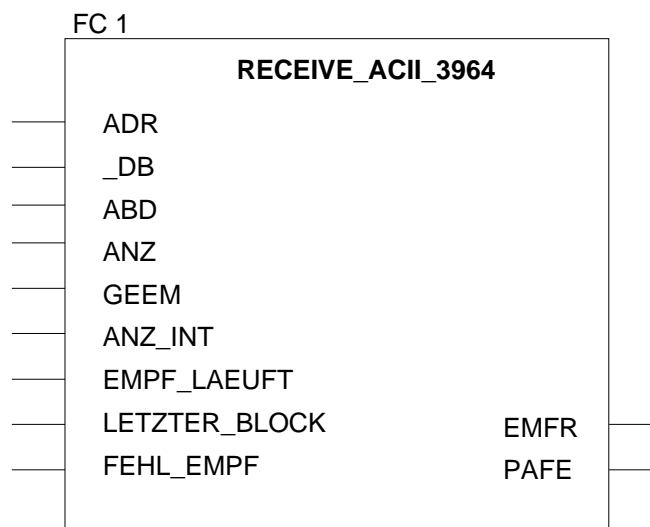
<b>FRG Sendefreigabe</b>	Bool: Wenn dieses Merker-Bit auf "1" gesetzt ist, wird die in ANZ angegebene Anzahl von Daten einmalig ausgegeben. Nach der Übertragung wird das Bit wieder auf "0" gesetzt. Ist dieses Bit beim Aufruf des FC 0 auf "0" gesetzt, wird der Baustein sofort wieder verlassen!
<b>GESE</b>	Word: Anzahl schon gesendeter Datenworte.
<b>ANZ_INT</b>	Word: Angabe der Sendelänge in Byte.
<b>ENDE_KOM</b>	Bool: Die Kommunikation ist fertig.
<b>LETZTER_BLOCK</b>	Bool: Letzter Block wird gesendet.
<b>SENDEN_LAEUFT</b>	Bool: Der Datenblock wird gesendet.
<b>FEHLER_KOM</b>	Bool: Fehler in der Kommunikation.
<b>PAFE</b>	<p>Byte: 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.</p> <p>Folgende Fehler sind möglich: 1 = Datenbaustein nicht vorhanden 2 = Datenbaustein zu kurz 3 = Datenbausteinnummer nicht im gültigen Bereich</p>

**RECEIVE\_ASCII\_3964 (FC 1)**

Dieser FC dient zum Datenempfang von einem Peripheriegerät unter ASCII, STX/ETX, 3964(R) und Modbus.

Über die Bezeichner `_DB` und `ADB` wird der Beginn des Empfangsfachs festgelegt.

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 wird dieses Bit vom Anwender zurückgesetzt. Solange dieses Bit auf "1" ist wird kein neues Telegramm in die SPS übernommen. Ankommende Telegramme werden je nach Pufferzahl auf der Baugruppe zwischengespeichert.



**ADR**                    Int: Peripherieadresse unter der das CP 240-Modul angesprochen wird. In Ihrem Projektiertool bestimmen Sie die Peripherieadresse unter der der CP 240 in das System einzubinden ist.  
Die Adresse liegt im Bereich PY000 ... PY240.

**\_DB**                    Block\_DB: Nummer des Datenbausteins, in den die empfangenen Daten abzulegen sind.

**ABD**                    Word: 1. Datenwort des Empfangsfachs.

**ANZ**                    Word: Wort-Variable, die die Anzahl der Bytes enthält, die empfangen werden sollen.

<b>GEEM</b>	Word: Anzahl schon empfangener Daten.
<b>ANZ_INT</b>	Word: Empfangslänge in Byte.
<b>EMPF_LAEUFT</b>	Bool: Der Empfang läuft.
<b>LETZTER_BLOCK</b>	Bool: Der letzte Block ist gesendet.
<b>FEHL_EMPF</b>	Bool: Fehler Kommunikation
<b>PAFE</b>	<p>Byte: 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.</p> <p>Folgende Fehler sind möglich:</p> <ul style="list-style-type: none"><li>1 = Datenbaustein nicht vorhanden</li><li>2 = Datenbaustein zu kurz</li><li>3 = Datenbausteinnummer nicht im gültigen Bereich</li></ul>

**FETCH\_RK512  
(FC 2)**

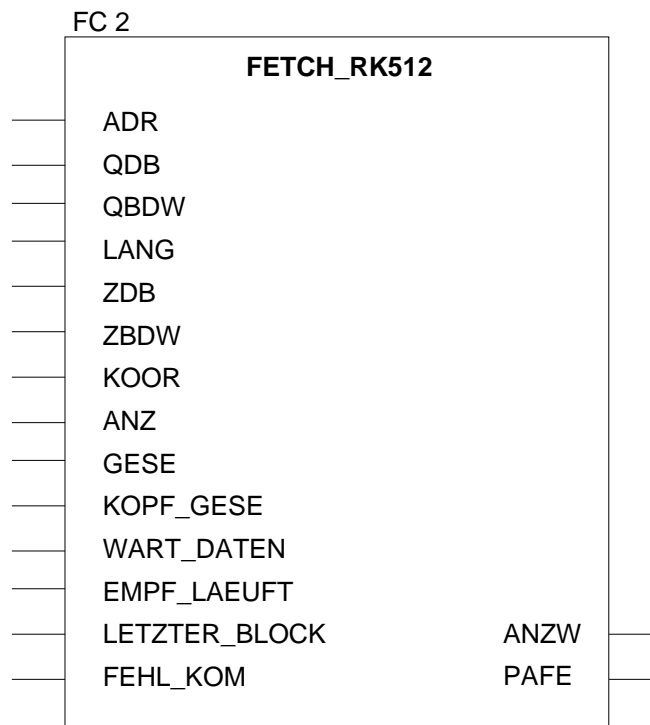
Daten anfordern über RK512

Dieser FC dient zur Datenanforderung von einem Peripheriegerät unter RK512. Dieser FC ist nur unter RK512 zulässig.

Der FETCH dient dazu Daten von einem Partner anzufordern. Hierzu wird ein Telegramm mit den Quell-Daten an den Partner gesendet. Daraufhin stellt der Partner die Daten zusammen und sendet diese zurück. Die empfangenen Daten werden nun in den angegebenen Ziel-Daten abgelegt. Die Daten werden beim Aufruf des FC an den Bezeichnern QDB, QBDW, LANG für den Quell-Bereich und an den Bezeichnern ZDB und ZBDW für den Ziel-Bereich angegeben.

Beim Aufruf des FCs wird anhand der Steuerbits geprüft, ob noch ein laufender Auftrag vorhanden ist. Sind alle Steuerbits "0", so wird ein neuer FETCH-Auftrag angestoßen. Hierzu wird der Telegrammkopf an den CP übergeben und anschließend auf die Quittung mit den Nutzdaten gewartet. Solange das Quittungstelegramm vom Partner nicht gesandt wurde, ist im Anzeigewort "Auftrag läuft" gesetzt. Erst nachdem der Empfang des Quittungstelegramms vom CP an die SPS gemeldet und die Nutzdaten übergeben wurden, wird "Auftrag fertig" im Anzeigewort gesetzt und die Kommunikation mit dem CP beendet. Wird ein Fehler in der Kommunikation festgestellt, so übergibt der CP eine Fehlernummer an die SPS. Daraufhin wird im Anzeigewort die Fehlernummer eingetragen und das Bit "Auftrag fertig mit Fehler" gesetzt.

Die Funktion ist solange im zyklischen Programm zu bearbeiten bis "Auftrag fertig - mit/ohne - Fehler" im Anzeigewort gesetzt ist.



**Hinweis!**

Damit der Slave auf die Master-Anforderung reagieren kann, ist im Slave ein S/R\_ALL (Send/Receive\_All) zyklisch auszuführen.





<b>ADR</b>	Int: Peripherieadresse unter der das CP 240-Modul angesprochen wird. In Ihrem Projektierool bestimmen Sie die Peripherieadresse unter welcher der CP 240 in das System einzubinden ist. Die Adresse liegt im Bereich PY000 ... PY240.
<b>QDB</b>	Int: Nummer des Datenbausteins, der die zu sendenden Daten enthält.
<b>QBDW</b>	Int: Erstes Datenwort im Quell-Datenbaustein.
<b>LANG</b>	Int: Länge der zu senden Daten in Worten.
<b>ZDB</b>	Block_DB: Ziel-Datenbaustein Nummer des Datenbausteins, in den die gesendeten Daten abzulegen sind.
<b>ZBDW</b>	Int: Erstes Datenwort im Ziel-Datenbaustein.
<b>KOOR</b>	Word: Koordinierungsbytebelegung: Hier wird die Verwendung eines Koordinierungsmerkers parametrieret. Im High-Byte muss die Byte-Nummer, im Low-Byte muss die Bit-Nummer des Koordinierungsmerkers parametrieret werden. Soll kein Koordinierungsmerker verwendet werden, so ist in High- und Low-Byte der Wert 255 vorzugeben. Mit dem Koordinierungsmerker wird der Zugriff auf den Quellbereich geregelt: Durch Setzen des Merkers können Sie im Partner-AG Ihre gesendeten Daten vor dem Überschreiben schützen. Durch Rücksetzen des Merkers können die Daten wieder überschrieben werden.
<b>ANZ</b>	Word: Anzahl empfangener Byte (intern).
<b>GESE</b>	Word: Anzahl empfangener Byte (intern).
<b>KOPF_GESE</b>	Bool: Kopf an Partner gesendet.

---

<b>WART_DATEN</b>	Bool: Warten auf Daten
<b>EMPF_LAEUFT</b>	Bool: Empfangen läuft.
<b>LETZTER_BLOCK</b>	Bool: Letzter Block gesendet.
<b>FEHL_KOM</b>	Bool: Es ist ein Fehler in der Kommunikation aufgetreten.
<b>ANZW</b>	Word: Anzeigenwort. Das Anzeigenwort belegt ein Merker-Wort. Im rechten Byte werden Statusbits abgelegt. Im linken Byte wird, wenn im rechten Byte die Kennung "Fertig mit Fehler" steht, eine Fehlernummer abgelegt.
<b>PAFE</b>	Byte: 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

### SEND\_RK512 Daten senden über RK512 (FC 3)

Dieser FC dient zur Datenausgabe an ein Peripheriegerät unter RK512. Dieser FC ist nur unter RK512 zulässig.

Der SEND\_RK512 dient dazu Daten aus einem definierten Quellbereich der SPS an einen Partner zu senden und diesem gleichzeitig mitzuteilen, wo die empfangenen Daten abzulegen sind. Der Quellbereich wird mit den Bezeichnern QDB, QBDW und LANG definiert. Der Zielbereich beim Partner wird mit den Bezeichnern ZDB und ZBDW festgelegt.

Beim Aufruf des FCs wird anhand der Steuerbits geprüft ob noch ein laufender Auftrag vorhanden ist. Sind alle Steuerbits "0", so wird ein neuer Sendeauftrag angestoßen. Hierzu wird das Telegramm bestehend aus Telegrammkopf und Nutzdaten an den CP übergeben und anschließend auf die Quittung gewartet.

Solange das Quittungstelegramm vom Partner nicht gesendet wurde, ist im Anzeigewort "Auftrag läuft" gesetzt. Erst nachdem der Empfang des Quittungstelegramms vom CP an die SPS gemeldet wurde, wird "Auftrag fertig" im Anzeigewort gesetzt und die Kommunikation mit dem CP beendet. Wird ein Fehler in der Kommunikation festgestellt, so übergibt der CP eine Fehlernummer an die SPS. Daraufhin wird im Anzeigewort die Fehlernummer eingetragen und das Bit "Auftrag fertig mit Fehler" gesetzt. Anschließend wird die Kommunikation mit dem CP beendet. Die Funktion ist solange im zyklischen Programm zu bearbeiten, bis "Auftrag fertig mit/ohne Fehler" im Anzeigewort gesetzt ist.



#### Hinweis!

Damit der Slave die vom Master gesendeten Daten verarbeiten kann, ist im Slave ein S/R\_ALL (Send/Receive\_All) zyklisch auszuführen.

---

<b>ADR</b>	Int: Peripherieadresse unter der das CP 240-Modul angesprochen wird. In Ihrem Projektierool bestimmen Sie die Peripherieadresse unter welcher der CP 240 in das System einzubinden ist. Die Adresse liegt im Bereich PY000 ... PY240.
<b>QDB</b>	Block_DB: Nummer des Datenbausteins, der die zu sendenden Daten enthält.
<b>QBDW</b>	Int: Erstes Datenwort des Sendefachs.
<b>LANG</b>	Int: Länge der zu sendenden Daten.
<b>ZDB</b>	Int: Nummer des Datenbausteins, in den die gesendeten Daten abzulegen sind.
<b>ZBDW</b>	Int: Erstes Datenwort des Empfangsfachs.
<b>KOOR</b>	Word: Hier wird die Verwendung eines Koordinierungsmerkers parametrisiert. Im High-Byte muss die Byte-Nummer, im Low-Byte muss die Bit-Nummer des Koordinierungsmerkers parametrisiert werden. Soll kein Koordinierungsmerker verwendet werden, so ist in High- und Low-Byte der Wert 255 vorzugeben. Mit dem Koordinierungsmerker wird der Zugriff auf den Quellbereich geregelt: Durch Setzen des Merkers können Sie im Partner-AG Ihre gesendete Daten vor dem Überschreiben schützen. Durch Rücksetzen des Merkers können die Daten wieder überschrieben werden.
<b>ANZ</b>	Word: Anzahl gesendeter Byte (intern).
<b>GESE</b>	Word: Anzahl gesendeter Byte (intern).
<b>KOPF_GESENET</b>	Bool: Kopf an Partner gesendet.

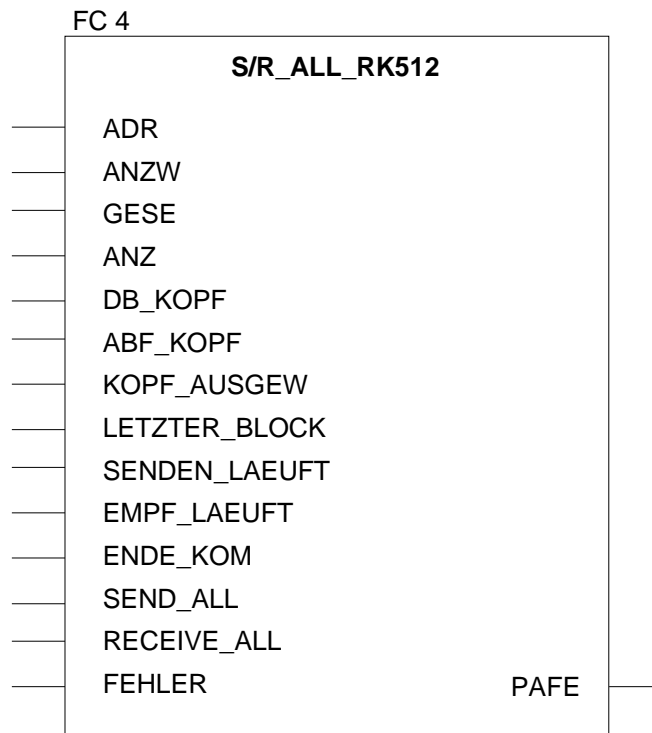
<b>ERSTER_BLOCK</b>	Bool: Erster Block der Daten wurde gesendet.
<b>SENDE_LAEUFT</b>	Bool: Senden läuft.
<b>SENDEN_FERTIG</b>	Bool: Senden der Daten abgeschlossen.
<b>LETZTER_BLOCK</b>	Bool: Letzter Block gesendet.
<b>FEHLER</b>	Es ist ein Fehler in der Kommunikation aufgetreten.
<b>ANZW</b>	Word: Anzeigenwort - Das Anzeigenwort belegt ein Merker-Wort. Im rechten Byte werden Statusbits abgelegt. Im linken Byte wird, wenn im rechten Byte die Kennung "Fertig mit Fehler" steht, eine Fehlernummer abgelegt.
<b>PAFE</b>	<p>Byte: 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.</p> <p>Folgende Fehler sind möglich:</p> <ul style="list-style-type: none"><li>1 = Datenbaustein nicht vorhanden</li><li>2 = Datenbaustein zu kurz</li><li>3 = Datenbausteinnummer nicht im gültigen Bereich</li></ul>

**S/R\_ALL\_RK512  
(FC 4)**

Der Baustein dient dazu FETCH- und SEND-Aufträge vom Partner zu bearbeiten.

Wird das System mit dem CP 240 als Slave eingesetzt, ist dieser FC in der Slave-CPU zyklisch aufzurufen. Nur dann kann der CP 240 auf die Master-Aufgaben reagieren. Bei FETCH werden die Daten bereitgestellt und an den Master übertragen. Die mit SEND vom Master empfangenen Daten werden übernommen, abgelegt und bestätigt.

Dieser FC ist nur unter RK512 zulässig.

**ADR**

Int: Peripherieadresse unter der das CP 240-Modul angesprochen wird. In Ihrem Projektierool bestimmen Sie die Peripherieadresse unter der der CP 240 in das System einzubinden ist.

Die Adresse liegt im Bereich PY000 ... PY240.

**ANZW**

Word: Anzeigenwort. Das Anzeigenwort belegt ein Wort. Im rechten Byte werden Statusbits abgelegt. Im linken Byte wird, wenn im rechten Byte die Kennung "Fertig mit Fehler" steht, eine Fehlernummer abgelegt.

**GESE**

Word: Anzahl empfangener Byte (intern).

**ANZ**

Word: Anzahl empfangener Byte (intern).

<b>ABF_KOPF</b>	Word: Erstes Datenwort im Empfangs/Sendefach.
<b>KOPF_AUSGEW</b>	Bool: Telegrammkopf ist ausgewertet.
<b>LETZTER_BLOCK</b>	Bool: Letzter Block gesendet.
<b>SENDEN_LAEUFT</b>	Bool: Senden läuft.
<b>EMPF_LAEUFT</b>	Bool: Das Empfangen der Daten läuft.
<b>ENDE_KOM</b>	Bool: Das Telegramm komplett gesendet/empfangen.
<b>SEND_ALL</b>	Bool: Der Baustein arbeitet als SEND_ALL.
<b>RECEIV_ALL</b>	Bool: Der Baustein arbeitet als RECEIVE_ALL.
<b>FEHLER</b>	Bool: Es ist ein Fehler in der Kommunikation aufgetreten.
<b>PAFE</b>	<p>Byte: 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.</p> <p>Folgende Fehler sind möglich:</p> <ul style="list-style-type: none"><li>1 = Datenbaustein nicht vorhanden</li><li>2 = Datenbaustein zu kurz</li><li>3 = Datenbausteinnummer nicht im gültigen Bereich</li></ul>

---

**SYNCHRON\_**  
**RESET**  
**Synchronisation**  
**und Rücksetzen**  
**(FC 9)**

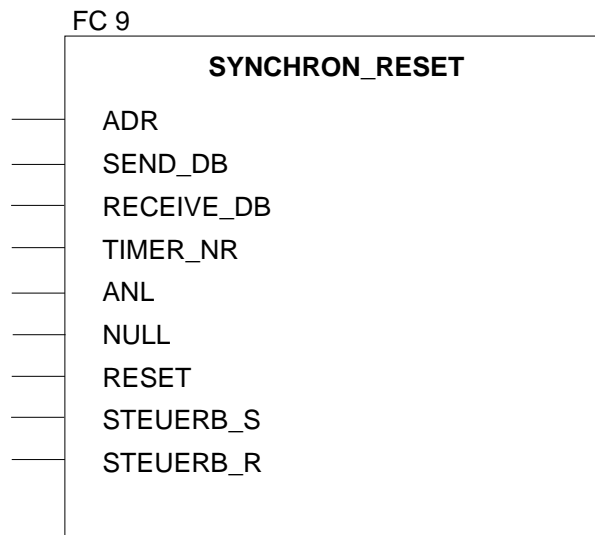
Der Baustein ist im zyklischen Programmteil aufzurufen. Mit dieser Funktion wird die Anlaufkennung des CP 240 quittiert, und so die Synchronisation zwischen CPU und CP hergestellt. Weiterhin kann bei einer Kommunikationsunterbrechung der CP rückgesetzt werden und so ein synchroner Anlauf erfolgen.

**Anlauf**

Bei einem Neustart oder Wiederanlauf ist einmalig das am Bezeichner ANL angegebene Bit auf "1" zu setzen und die Bearbeitung der SEND-/RECEIVE-Bausteine zu unterdrücken, solange dieses Bit nicht von der Funktion quittiert wurde.

**Reset**

Durch Setzen des Bits "RESET" wird eine Kennung an den CP übergeben und so der CP veranlasst, alle Puffer und Zeiger zu löschen. Hat der CP dies durchgeführt, setzt er die SYNCHRON-Kennung. Nachdem die CPU diese quittiert hat, kann die Kommunikation wieder erfolgen.

**ADR**

Int: Peripherieadresse unter der das CP 240-Modul angesprochen wird. In Ihrem Projektierool bestimmen Sie die Peripherieadresse unter der der CP 240 in das System einzubinden ist.

Die Adresse liegt im Bereich PY000 ... PY240.

**SEND\_DB**

Block\_DB: Datenbaustein für Datenübergabe an CP.

**RECEIVE\_DB**

Block\_DB: Datenbaustein für Datenübernahme von CP.



<b>TIMER_NR</b>	Timer: Nummer des Timers für die Wartezeit.
<b>ANL</b>	Bool: Ein Anlauf wurde durchgeführt. Über dieses Bit wird dem HTB mitgeteilt, dass an der CPU STOP/START bzw. NETZ-AUS/NETZ-EIN erfolgt ist und nun eine Synchronisation erfolgen muss. Nachdem die Synchronisation erfolgt ist, wird das Bit vom HTB gelöscht.
<b>NULL</b>	Bool: Null an CP senden (interne Auswertung).
<b>RESET</b>	Bool: Rücksetzen des CP 240. Durch Setzen dieses Bits kann der Anwender aus dem SPS-Programm heraus den CP 240 rücksetzen. Wird dieses Bit gesetzt, so schreibt der Hantierungsbaustein die Reset-Kennung auf den CP und wartet bis diese quittiert wird. Danach wird wie beim Anlauf verfahren.
<b>STUERB_S</b>	Byte: Steuerbit für SEND-FC und S/R_ALL-FC. Hier ist das Merkerbyte anzugeben, in dem die Steuerbits für den SEND-FC abgelegt sind.
<b>STUERB_R</b>	Byte: Steuerbit für RECEIVE-FC und FETCH-FC. Hier ist das Merkerbyte anzugeben in dem die Steuerbits für den RECEIVE-FC abgelegt sind.

**Programmier-  
beispiel**

Aufruf der Bausteine im OB1

```

CALL "DPRD_DAT"           //Daten von Baugruppe lesen
  LADDR :=W#16#100
  RET_VAL:=MW100
  RECORD :=P#DB11.DBX 0.0 BYTE 16

CALL FC      9           //Synchron aufrufen
  ADR        :=0         //1. DW im SEND/EMPF_DB
  SEND_DB    :=DB10      //Send_DB Baugruppe
  RECEIVE_DB:=DB11      //Empfang_DB Baugruppe
  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
SPB    schr

CALL FC      1           //Receive Daten
  ADR        :=0         //1. DW im SEND/EMPF_DB
  SEND_DB    :=DB10      //Send_DB Baugruppe
  EMPF_DB    :=DB11      //Empfang_DB Baugruppe
  _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
SEND_DB     :=DB10      //Send_DB Baugruppe
EMPF_DB     :=DB11      //Empfang_DB Baugruppe
_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

schr: CALL   "DPWR_DAT"   //Daten auf Baugruppe
                          schreiben
LADDR       :=W#16#100
RECORD      :=P#DB10.DBX 0.0 BYTE 16
RET_VAL     :=MW102

Programm im Anlauf OB100

UN         M         3.0
S          M         3.0           //Anlauf der CPU erfolgt
```

## Technische Daten

### CP 240 mit 20mA/RS232C- Schnittstelle

Elektrische Daten	VIPA 240-1BA00, VIPA 240-1BA10
Anzahl der Kanäle	1
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme	max. 200mA
ext. Spannungsversorgung	24V-Versorgung bei aktiver 20mA-Schnittstelle über Frontbuchse Pin 11 (+24V) und Pin 22 (Masse 24V) zur Erzeugung von 20mA.
Potenzialtrennung	keine
Statusanzeige (LEDs)	über LED auf der Frontseite
Anschlüsse / Schnittstellen	25pol. SubD-Buchse für RS232C und 20mA (umschaltbar) ASCII-Übertragung, 3964(R), 3964(R) mit RK512, Modbus (VIPA 240-1BA10)
Übertragungsrate -1BA00	150Baud bis 115kBaud
-1BA10	150Baud bis 38,4kBaud
Stopbit	1, 1.5, 2 (parametrierbar)
Parität	keine, gerade, ungerade (parametrierbar)
Flusskontrolle	keine, Hardware, XON/XOFF
ZVZ	Werte von 0 bis 5s
Programmierdaten	
Eingabedaten	16Byte
Ausgabedaten	16Byte
Parameterdaten	16Byte
Diagnosedaten	4Byte (keine bei Modbus)
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g

**CP 240 mit  
RS422/485-  
Schnittstelle**

Elektrische Daten	VIPA 240-1CA00, VIPA 240-1CA10
Anzahl der Kanäle	1
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme	max. 200mA
ext. Spannungsversorgung	-
Potenzialtrennung	>= AC 500V, nach DIN 19258
Statusanzeige (LEDs)	über LED auf der Frontseite
Anschlüsse / Schnittstellen	25pol. SubD-Buchse für RS422/RS485 ASCII-Übertragung, 3964(R), 3964(R) mit RK512, Modbus (VIPA 240-1CA10)
Übertragungsrate -1CA00	150Baud bis 115kBaud
-1CA10	150Baud bis 38,4kBaud
Stopbit	1, 1.5, 2 (parametrierbar)
Parität	keine, gerade, ungerade (parametrierbar)
Flusskontrolle	keine, Hardware, XON/XOFF
ZVZ	Werte von 0 bis 5s
Programmierdaten	
Eingabedaten	16Byte
Ausgabedaten	16Byte
Parameterdaten	16Byte
Diagnosedaten	4Byte (keine bei Modbus)
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g



## Teil 10 Zähler-Module

### Überblick

In diesem Kapitel finden Sie Informationen über den Anschluss und die Parametrierung des SSI-Moduls FM 250 S.

Vom Zähler-Modul FM 250 werden die einzelnen Betriebsarten der verschiedenen Zähler-Modi, d.h. das Verhalten der Zähler beim Anlegen der verschiedenen Eingangssignale beschrieben.

Nachfolgend sind beschrieben:

- SSI-Modul FM 250S
- Zähler-Modul FM 250
- Technische Daten

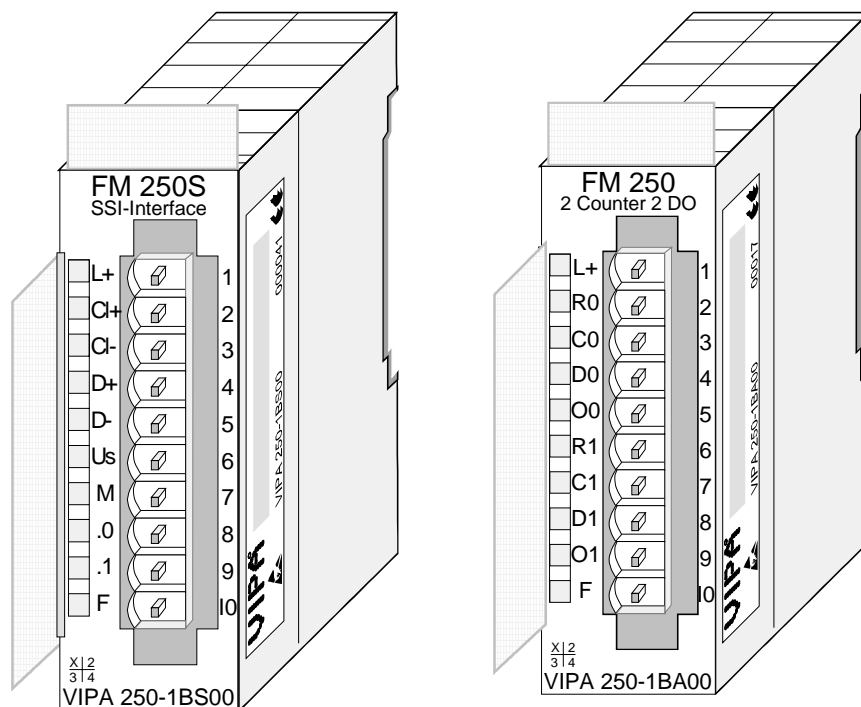
### Inhalt

Thema	Seite
<b>Teil 10 Zähler-Module .....</b>	<b>10-1</b>
Systemübersicht .....	10-2
FM 250S - SSI-Interface - Aufbau.....	10-3
FM 250 - Zähler-Modul - Aufbau .....	10-9
Übersicht Zählermodi und Anschaltung .....	10-12
Zählermodi.....	10-14
Technische Daten.....	10-58

## Systemübersicht

Nachfolgend finden Sie eine Übersicht der bei VIPA zur Zeit erhältlichen Mess-Module:

SSI-Interface FM 250S, Zähler-Modul FM 250



### Bestelldaten

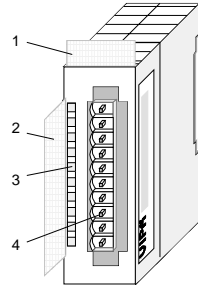
Typ	Bestellnummer	Beschreibung
FM 250S	VIPA 250-1BS00	SSI-Interface
FM 250	VIPA 250-1BA00	Zähler-Modul (2 Counter 2 DO)



## FM 250S - SSI-Interface - Aufbau

- Grundlagen**
- Eine SSI-Schnittstelle ist eine synchron getaktete serielle Schnittstelle. SSI ist die Abkürzung für **S**ynchron **S**erielles **I**nterface. Das SSI-Modul ermöglicht den Anschluss von absolut codierten Messwertaufnehmern mit einer SSI-Schnittstelle.
- Das Modul wandelt die serielle Information des Messwertaufnehmers in eine parallele Information um und stellt diese der Steuerung zur Verfügung. Es besteht die Möglichkeit, die Daten im Gray- oder Binär-Code zu übertragen.
- Schaltbare Ausgänge**
- Neben den SSI-Signalen Clock, Data und der Gebersversorgung stehen zusätzlich zwei Ausgänge zur Verfügung, die bei Überschreiten gesetzt bzw. rückgesetzt werden können.
- Der Ausgang 0 kann zusätzlich als Hold-Eingang parametrierbar werden. Der SSI-Geber-Wert wird in diesem Fall eingefroren, sobald an Ausgang 0 ein 24V-High-Pegel anliegt. Bei LOW-Pegel werden die aktuellen Werte des SSI-Gebers ausgegeben.
- Sie können die Ausgänge zudem so parametrieren, dass sie auch bei aktiviertem BASP-Signal gesetzt bleiben.
- Eigenschaften**
- Der Verkabelungsaufwand ist von der Datenwortlänge unabhängig. Es werden immer 4 Leitungen verwendet.
  - Maximale Datensicherheit durch Verwendung von symmetrischen Takt- und Datensignalen.
  - Sichere Datenerfassung durch Einsatz des einschrittigen Gray-Codes (wählbar).
  - Galvanische Trennung von Empfänger und Codierer durch Optokoppler.
  - 1 SSI-Kanal
  - Direkte Versorgung des SSI-Gebers über Frontstecker
  - Versorgungsspannung DC 24V
  - Baudrate von 100kBaude bis 600kBaude wählbar
  - 2 parametrierbare digitale Ausgänge, davon einer als Hold-Eingang zum Einfrieren des aktuellen SSI-Geber-Werts parametrierbar
  - Messwert in Gray- oder Binär-Code wählbar
  - 4Byte Parametrierdaten
  - 4Byte Eingabedaten
  - 4Byte Ausgabedaten
  - Parametrierung über Kontrollbyte

**Aufbau**

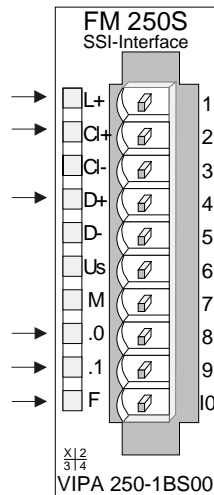


- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

- L+ LED (gelb)  
Versorgungsspannung liegt an
- Ci+ LED (grün)  
Clock-Ausgang
- D+ LED (grün)  
Geber-Daten Eingang
- .0 LED (grün)  
Ein-/Ausgang 0
- .1 LED (grün)  
Ein-/Ausgang 1
- F LED (rot)  
Fehler /Überlast



**Pin Belegung**

- 1 Versorgungsspannung +24V DC
- 2 CLK+ (Ausgang)
- 3 CLK- (Ausgang)
- 4 Data+ (Eingang)
- 5 Data- (Eingang)
- 6 DC 24V SSI-Geber-Versorgung
- 7 Masse SSI-Geber-Versorgung
- 8 Ein-/Ausgang .0 und Hold-Eing.
- 9 Ein-/Ausgang .1
- 10 Versorgungsspannung Masse

**LEDs**

Das SSI-Interface besitzt verschiedene LEDs. Die Verwendung dieser LEDs finden Sie in der nachfolgenden Tabelle:

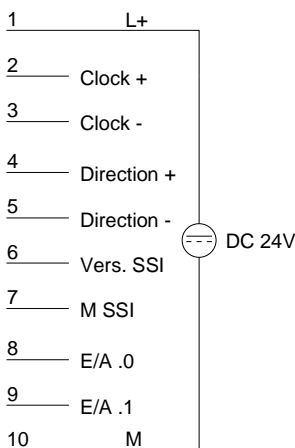
Bez.	Farbe	Bedeutung
L+	gelb	Signalisiert eine anliegende Betriebsspannung 24V
C+	grün	Leuchtet bei ausgehenden Clock-Impulsen
D+	grün	Leuchtet nicht bei aktivierter Hold-Funktion und 24V an E/A .0
.0	grün	Leuchtet wenn Daten vom Geber empfangen werden (Leitungstest)
.1	grün	Leuchtet bei anliegender 24V-Spannung an E/A .0
F	rot	Leuchtet bei anliegender 24V-Spannung an E/A .1
		Leuchtet bei Kurzschluss oder Überlast an einem der beiden E/A .0/.1

**Leitungslängen**

Die Baudrate ist von der Leitungslänge und vom SSI-Geber abhängig. Die Leitungen müssen paarig verdreht und abgeschirmt sein. Die hier gemachten Angaben sollen nur als Richtlinie dienen.

- < 400m: → 100kBaud
- < 100m: → 300kBaud
- < 50m: → 600kBaud

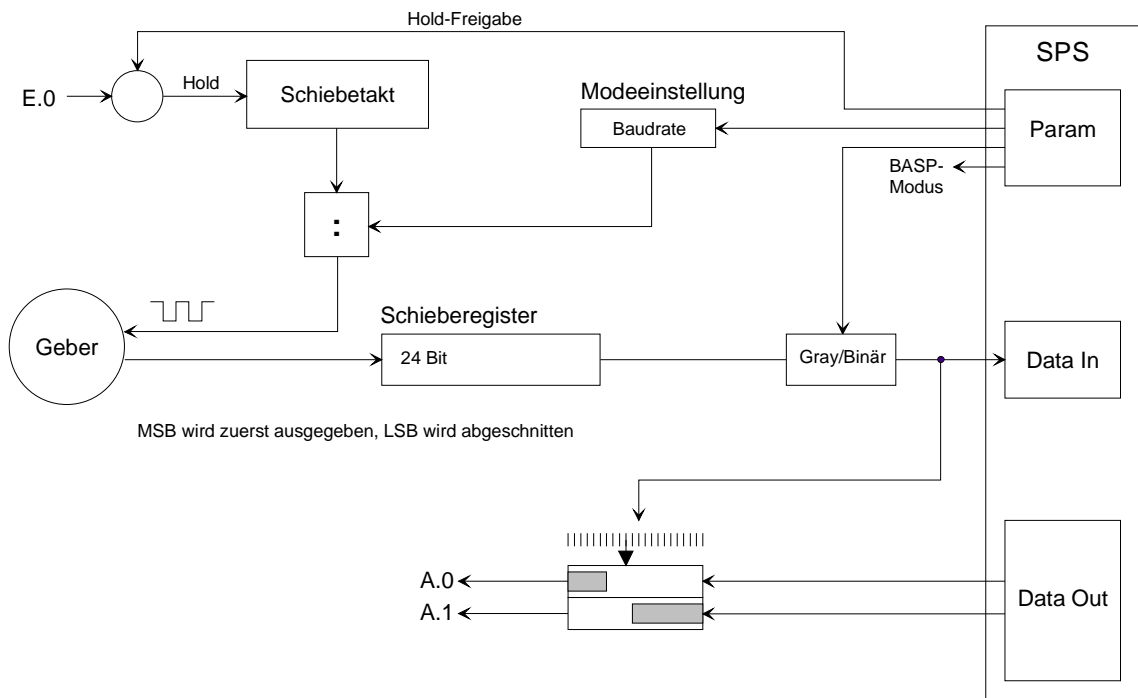
**Anschlussbild**



Das SSI-Interface besitzt ein eingebautes Netzteil. Das Netzteil ist mit 24V-Gleichspannung über L+ und M zu versorgen.

Über die Versorgungsspannung wird neben der Interface-Elektronik auch ein angeschlossener SSI-Geber mit DC 24V an Klemme 6 und 7 versorgt.

**Prinzipschaltbild**



**Parametrierdaten** Als Parametrierdaten werden 4Byte übergeben. In diesen Bytes bestimmen Sie die Baudrate, die Art der Codierung, die Auswertung des kombinierten E/A .0 und des BASP-Signals.

Die Parametrierdaten haben folgenden Aufbau:

Byte	Bit 0 ... Bit 7
0	Bit 0 ... Bit 7: reserviert
1	Bit 0 ... Bit 7: reserviert
2	Baudrate 0: 300kBaud (Default) 1: 100kBaud 2: 300kBaud 3: 600kBaud 4...255: 300kBaud
3	Bit 0: Codierung 0: Binär-Code (Default) 1: Gray-Code Bit 2: SSI-Format 0: Multiturn (24Bit) 1: Singleturn (12Bit) Bit 4: Hold-Funktion 0: deaktivieren 1: aktivieren Bit 7: BASP-Signal 0: ignorieren 1: auswerten

## Parameter

### Baudrate

Der an das SSI-Interface angeschlossene Geber übergibt seine Daten in serieller Form. Er erhält hierzu vom SSI-Interface einen Takt. Diesen Takt geben Sie hier als Baudrate vor. Sie haben die Wahl zwischen 100, 300 und 600kBaud. Die Grundeinstellung ist 300kBaud.

**Codierung**

Der Gray-Code ist eine andere Darstellungsform des Binärcodes. Seine Grundlage besteht darin, dass sich zwei benachbarte Gray-Zahlen in genau einem Bit unterscheiden.

Bei Einsatz des Gray-Codes können Übertragungsfehler leicht erkannt werden, da sich benachbarte Zeichen ja nur in einer Stelle unterscheiden dürfen.

Regel für den Gray-Code in Tabellenform:

Dezimal	Gray-Code
0	0 0 0 0
1	0 0 0 1
2	0 0 1 1
3	0 0 1 0
4	0 1 1 0
5	0 1 1 1
6	0 1 0 1
7	0 1 0 0
8	1 1 0 0
9	1 1 0 1
10	1 1 1 1
11	1 1 1 0
12	1 0 1 0
13	1 0 1 1
14	1 0 0 1
15	1 0 0 0

D.h. die letzte Stelle der Zahl ergibt sich immer aus der vertikalen Wiederholung der Zahlenfolge "0 11 0", die zweitletzte aus der Wiederholung "00 1111 00", die drittletzte aus der Wiederholung von 4x"0", 8x"1" und wieder 4x"0" usw. (siehe Spalten in der Tabelle!).

**Hold-Funktion**

Hier können Sie einstellen, dass der E/A .0 als Hold-Eingang ausgewertet wird. Wenn Sie diese Funktion über die Parametrierung aktiviert haben, wird, sobald an E/A .0 24V anliegen, der aktuelle Geberwert gespeichert. Der Geberwert wird erst wieder aktualisiert, wenn die 24V an E/A .0 nicht mehr anliegen.

Bitte beachten Sie, dass dann E/A .0 nur noch als Eingang arbeitet.

**BASP-Signal**

BASP ist die Abkürzung für Befehlsausgabesperre, d.h. solange das BASP-Signal über den Rückwandbus anliegt, werden alle Ausgänge zurückgesetzt und gesperrt. Die Auswertung des BASP-Signals können Sie durch Setzen des Bits unterbinden, so dass eine Auswertung des BASP-Signals nicht erfolgt und die Ausgänge gesetzt bleiben.

**Zugriff auf das SSI-Interface****Eingabedaten (Data In)**

Die Eingabedaten, die Ihr SSI-Geber liefert sind 4Byte lang, wobei das Byte 0 zur Statusanzeige der E/A verwendet werden kann. Abhängig vom eingestellten Modus werden die Daten im Binär- oder im Gray-Code bereitgestellt.

Byte	Data In
0	Bit 0: Status E/A .0. Die Rückmeldung des Status erfolgt aber nur, wenn dies im Ausgabe-Doppelwort für den jeweiligen E/A vorgegeben wurde! Bit 1: Status E/A .0. Die Rückmeldung des Status erfolgt aber nur, wenn dies im Ausgabe-Doppelwort für den jeweiligen E/A vorgegeben wurde! Bit 2-7: reserviert
1	SSI-Geberwert: HB
2	SSI-Geberwert: MB
3	SSI-Geberwert: LB

**Ausgabedaten (Data Out)**

Über Data Out haben Sie die Möglichkeit die 2 E/As auf dem SSI-Interface abhängig von einem Geberwert anzusteuern. Die Ausgabedaten setzen sich aus 4Byte zusammen.

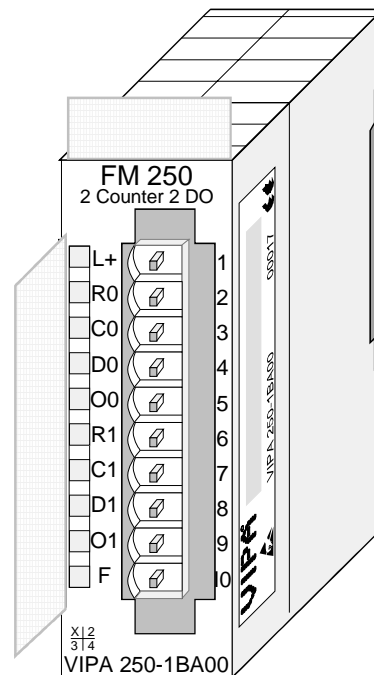
Im SSI-Geber werden immer 8Byte Ausgabedaten gehalten, d.h. Sie können zwei Vergleichswerte mit zugehörigem Control-Byte vorgeben.

Im Control-Byte können Sie genau spezifizieren, wie der Vergleichswert welchen Ausgang beeinflussen soll und ob der Status der E/As über die Eingangsbytes angezeigt werden soll.

Die Belegung der Ausgangs-Bytes finden Sie in der folgenden Tabelle.

Byte	Data Out
0	Bit 0-1: Sollwertvorgabe 00: keine Sollwertvorgabe 01: für Ausgang 0 10: für Ausgang 1 11: für beide Ausgänge Bit 2: Statusübergabe in Eingabedatenbereich 0: keine Statusübergabe 1: Status an Eingabe-Bereich übergeben Bit 3: Bedingung für Ausgang setzen 0: wenn Ist-Wert größer Vergleichswert 1: wenn Ist-Wert kleiner Vergleichswert Bit 4-7: reserviert
1	Vergleichswert: HB
2	Vergleichswert: MB
3	Vergleichswert: LB

## FM 250 - Zähler-Modul - Aufbau



### Hinweis!

Die hier gemachten Angaben gelten nur für das Zähler-Modul mit der Best.-Nr.: *VIPA 250-1BA00* mit dem Ausgabestand 5 und höher.

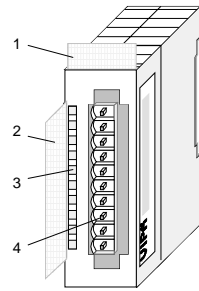
Der Zähler erfasst die Impulse der angeschlossenen Geber und verarbeitet diese Impulse entsprechend des gewählten Modus. Das Modul verfügt über 2 Kanäle bei einer Breite von je 32Bit.

Sie haben 24 Zähler-Modi und pro Kanal einen 24V-Ausgang, der je nach Modus angesteuert wird.

### Eigenschaften

- 2 Kanäle mit je 32Bit
- Versorgungsspannung DC 24V und über Rückwandbus
- 24V-Ausgänge (max. 0,5A) frei konfigurierbar
- Laden der Zähler- und Compare-Register über Control-Byte
- normaler Auf-, Abwärtszähler mit 32Bit oder 16Bit Breite
- Compare- und Auto-Reload-Funktionalität
- verschiedene Modi für Encoder-Impulse
- Periodendauermessung und Frequenzmessung

**Aufbau**

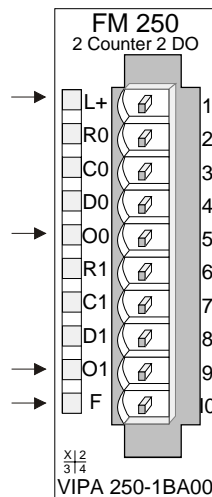


- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

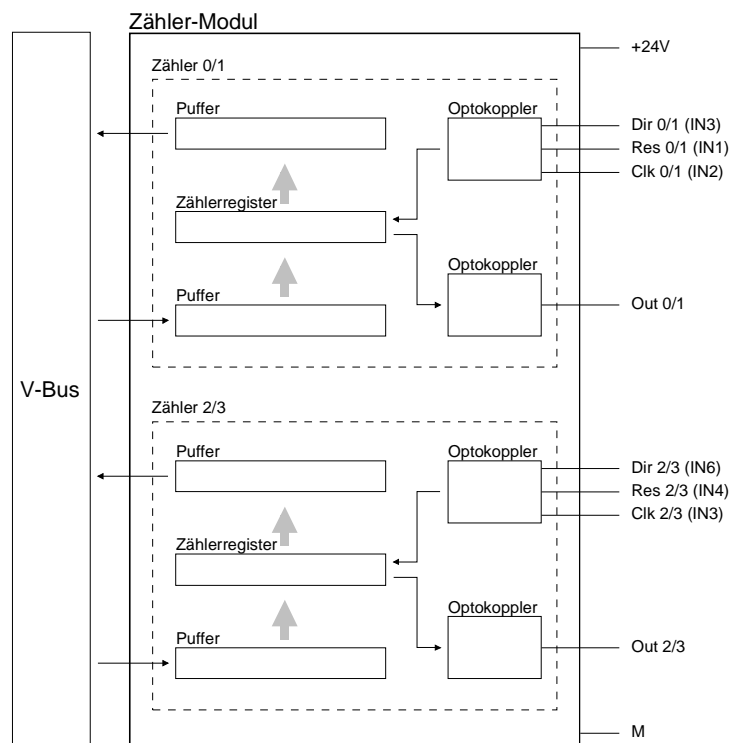
- L+** LED (gelb)  
Versorgungsspannung liegt an
- O0** LED (grün)  
Ausgang Zähler 0
- O1** LED (grün)  
Ausgang Zähler 1
- F** LED (rot)  
Fehler bei Überlast



**Pin Belegung**

- 1 Versorgungsspannung +24V
- 2 IN1 Eingang 1 Zähler 0/1
- 3 IN2 Eingang 2 Zähler 0/1
- 4 IN3 Eingang 3 Zähler 0/1
- 5 OUT0 Ausgang Zähler 0/1
- 6 IN4 Eingang 4 Zähler 2/3
- 7 IN5 Eingang 5 Zähler 2/3
- 8 IN6 Eingang 6 Zähler 2/3
- 9 OUT1 Ausgang Zähler 2/3
- 10 Versorgungsspannung Masse

**Prinzipschaltbild**







## Übersicht Zählermodi und Anschaltung

Mode	Kombinierbar	Funktion	IN1	IN2	IN3	IN4	IN5	IN6	OUT0	OUT1	Auto Re-load	Compare Load
			<b>Zähler 0/1</b>			<b>Zähler 2/3</b>						
0	ja	32 Bit Zähler	RES	CLK	DIR	RST	CLK	DIR	=0	=0	nein	=0
1	ja	Encoder 1 Flanke	RES	A	B	RST	A	B	=0	=0	nein	=0
3	ja	Encoder 2 Flanken	RES	A	B	RST	A	B	=0	=0	nein	=0
5	ja	Encoder 4 Flanken	RES	A	B	RST	A	B	=0	=0	nein	=0
			<b>Zähler 1 Zähler 0</b>			<b>Zähler 3 Zähler 2</b>						
8	ja	2x16 Bit Zähler up/up	-	CLK	CLK	-	CLK	CLK	-	-	nein	nein
9	ja	2x16 Bit Zähler down/up	-	CLK	CLK	-	CLK	CLK	-	-	nein	nein
10	ja	2x16 Bit Zähler up/down	-	CLK	CLK	-	CLK	CLK	-	-	nein	nein
11	ja	2x16 Bit Zähler down/down	-	CLK	CLK	-	CLK	CLK	-	-	nein	nein
			<b>Zähler 0/1</b>			<b>Zähler 2/3</b>						
12	ja	32 Bit Zähler up + gate	RES	CLK	Gate	RST	CLK	Gate	=comp	=comp	nein	ja
13	ja	32 Bit Zähler down + gate	RES	CLK	Gate	RST	CLK	Gate	=comp	=comp	nein	ja
14	ja	32 Bit Zähler up + gate	RES	CLK	Gate	RST	CLK	Gate	=comp	=comp	ja	ja
15	ja	32 Bit Zähler down + gate	RES	CLK	Gate	RST	CLK	Gate	=comp	=comp	ja	ja
			<b>Kombination aus Zähler 0 ... 3</b>									
16	nein	Frequenzmessung	RES	CLK	Start	Stop	-	-	Messung läuft	Messende	nein	ja
17	nein	Periodendauerermessung	RES	CLK	Start	Stop	-	-	Messung läuft	Messende	nein	ja
18	nein	Frequenzmessung mit Gate-Ausgang	RES	CLK	Start	Stop	-	-	Mess-gate	Gate	nein	ja
19	nein	Periodendauerermessung mit Gate-Ausgang	RES	CLK	Start	Stop	-	-	Mess-gate	Gate	nein	ja
			<b>Zähler 0/1</b>			<b>Zähler 2/3</b>						
6	ja	Pulse low, 50kHz with Direction Input	RES	Pulse	DIR	RES	Pulse	DIR	-	-		
20	ja	Pulse low, prog. timebase with Direction Input	RES	Pulse	DIR	RES	Pulse	DIR	-	-		
21	ja	Pulse low, up, prog. timebase with Gate	RES	Pulse	Gate	RES	Pulse	Gate	-	-		
22	ja	Pulse high, up, prog. timebase with Gate	RES	Pulse	Gate	RES	Pulse	Gate	-	-		
			<b>Zähler 0/1</b>			<b>Zähler 2/3</b>						
23	ja	One Shot, up, Set	RES	CLK	Gate	RES	CLK	Gate			nein	ja
24	ja	One Shot, down, Set	RES	CLK	Gate	RES	CLK	Gate			nein	ja
25	ja	One Shot, up, Reset	RES	CLK	Gate	RES	CLK	Gate			nein	ja
26	ja	One Shot, down, Reset	RES	CLK	Gate	RES	CLK	Gate			nein	ja
			<b>Zähler 0/1</b>			<b>Zähler 2/3</b>						
27	ja	32 Bit Zähler	Gate/R	CLK	DIR	Gate/R	CLK	DIR	=0	=0	nein	=0
28	ja	Encoder 1 Flanke	Gate/R	A	B	Gate/R	A	B	=0	=0	nein	=0
29	ja	Encoder 2 Flanken	Gate/R	A	B	Gate/R	A	B	=0	=0	nein	=0
30	ja	Encoder 4 Flanken	Gate/R	A	B	Gate/R	A	B	=0	=0	nein	=0

Aufgrund technischer Ergänzungen wurde mit jedem Ausgabestand die Funktionalität des Zählermoduls erweitert. Eine Zuordnung der Modi zum Ausgabestand sehen Sie hier:

- |           |                |               |                  |
|-----------|----------------|---------------|------------------|
| Mode 0-5  | Ausgabestand 3 | Mode 6, 20-26 | Ausgabestand 6/7 |
| Mode 0-17 | Ausgabestand 4 | Mode 27-30    | Ausgabestand 8   |
| Mode 0-19 | Ausgabestand 5 |               |                  |

**Begriffserklärung: RES**

RESET-Signal, das während des Messvorgangs auf LOW liegen muss. Ein HIGH-Pegel löscht abhängig vom verwendeten Modus einen oder beide Zähler.

**CLK**

Taktsignal eines angeschlossenen Gebers.

Start bzw. Stop

HIGH-Pegel löst Zählerstart aus bzw. stoppt den Zählvorgang. Nach einem Start-Pegel beginnt der Zählvorgang mit dem nächsten dem eingestellten Modus entsprechenden CLK-Pegel.

**DIR**

Der Signalpegel von DIR gibt im Modus 0 die Zählrichtung an.

LOW-Pegel: Aufwärtszähler

HIGH-Pegel: Abwärtszähler

**Auto Reload**

Die Auto-Reload-Funktion übergibt in den Zähler einen von Ihnen vorgegebenen Wert, sobald der Zählerstand mit dem Inhalt des Compare-Registers übereinstimmt.

**Compare Load**

Mit der Compare-Funktion können sie einen Zählerendwert angeben, der bei Erreichen je nach Modus einen Ausgang ansteuert oder über die Auto-Reload-Funktion die Zähler neu startet.

**Gate**

Torsignal, in dem der Zähler zählt.

**Gate/R**

Mit steigender Flanke dieses Signals wird der Zähler zurückgesetzt.

Während das Signal auf "1" liegt, ist der Zähler freigegeben.

**Messgate**

Zustandsanzeige der Zähleraktivität - bekommt HIGH-Pegel mit dem 1. CLK-Signal und LOW-Pegel mit dem letzten CLK-Signal (Mode 18 ... 19).

**Pulse**

Die Pulsbreite eines eingespeisten Signals wird mit einer internen Zeitbasis gemessen.

**Fref**

Referenz- bzw. Taktfrequenz, die im Mode 6 fest auf 50kHz eingestellt ist.

Die Taktfrequenz "Fref" für die Zähler-Modi 20, 21, 22 ist programmierbar:

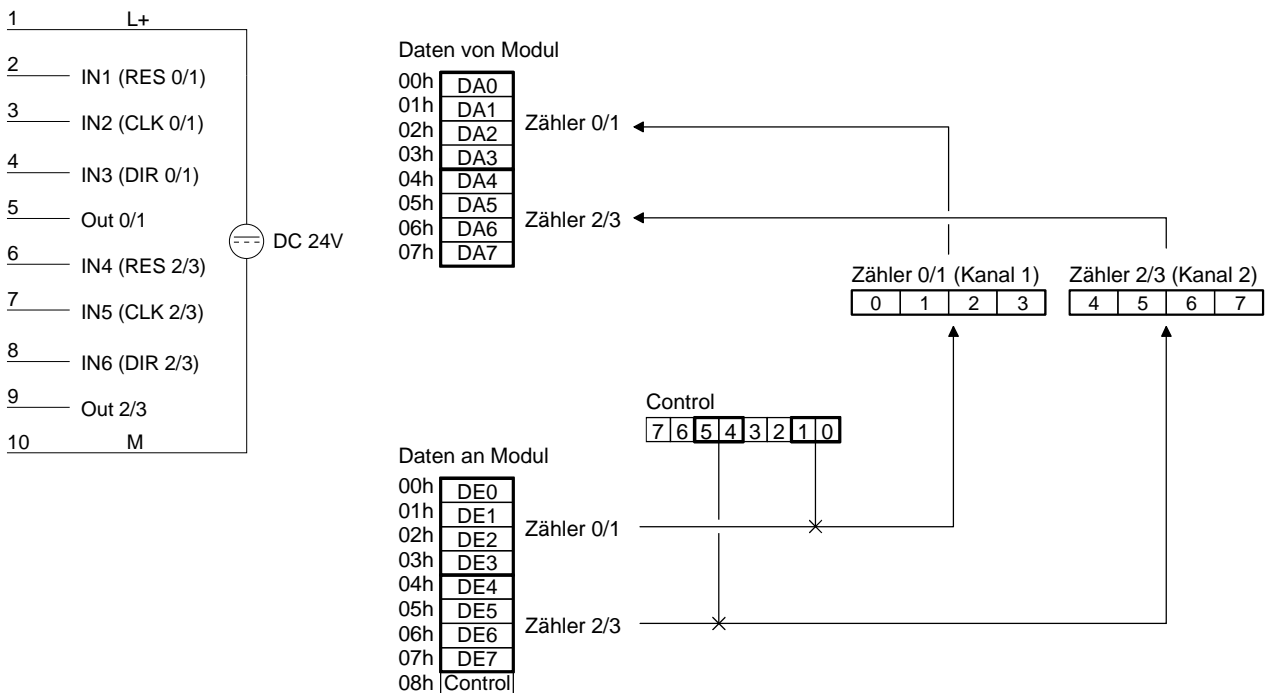
Parameter	Fref
0	10MHz
1	1MHz
2	100kHz
3	10kHz

# Zählermodi

## Mode 0 32Bit Zähler

Unter Mode 0 werden je 2 Zähler (16 Bit) zu einem 32 Bit Zähler zusammengefasst. Die Zählrichtung bestimmen Sie über DIR (IN3 bzw. IN6). Mit jeder steigenden Flanke des Eingangstakts wird der Zähler um 1 inkrementiert bzw. dekrementiert. RES muss während des Zählvorgangs auf LOW liegen. Hat RES HIGH-Pegel, wird der Zählerstand gelöscht. Bei Erreichen von Null wird der Ausgang OUT des entsprechenden Zählers mindestens für 100ms gesetzt, auch wenn der Zähler weiterzählt. Sollte der Zähler bei Null stoppen, bleibt der Ausgang gesetzt.

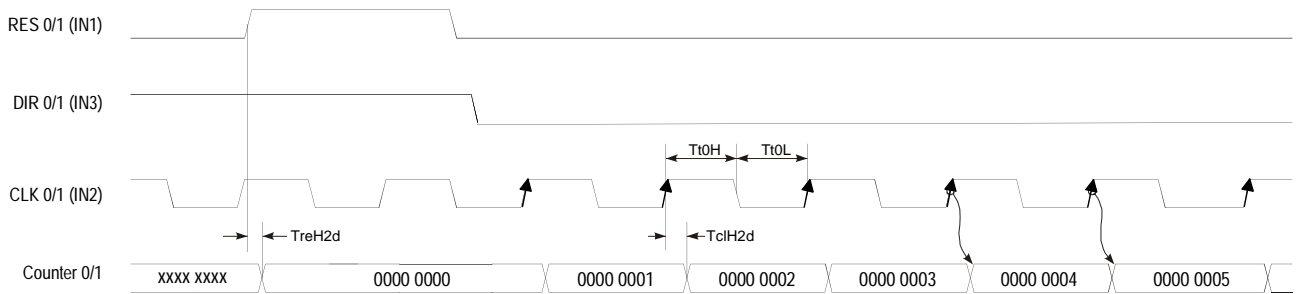
### Anschlussbelegung Zählerzugriff



**Aufwärtszähler**

Durch einen LOW-Pegel am Eingang DIR im Mode 0 wird der Zähler als Aufwärtszähler eingestellt.

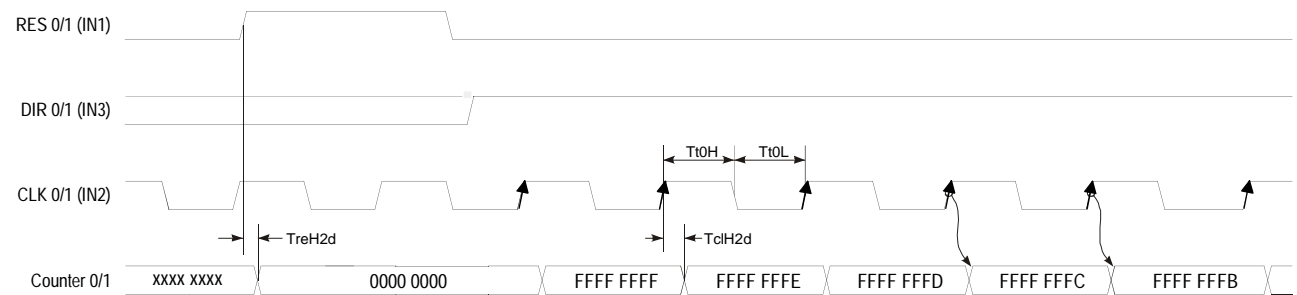
Timing-Diagramm am Beispiel von Zähler 0/1:



**Abwärtszähler**

Durch einen HIGH-Pegel am Eingang DIR im Mode 0 wird der Zähler als Abwärtszähler eingestellt.

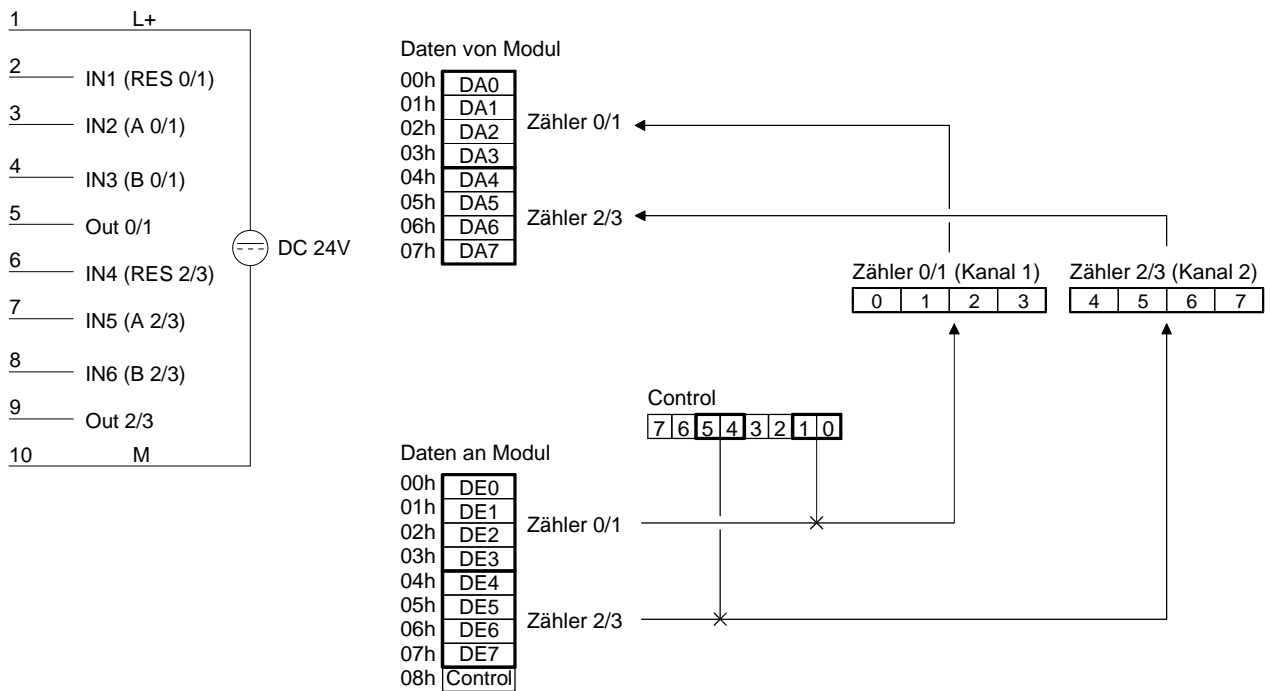
Timing-Diagramm am Beispiel von Zähler 0/1:



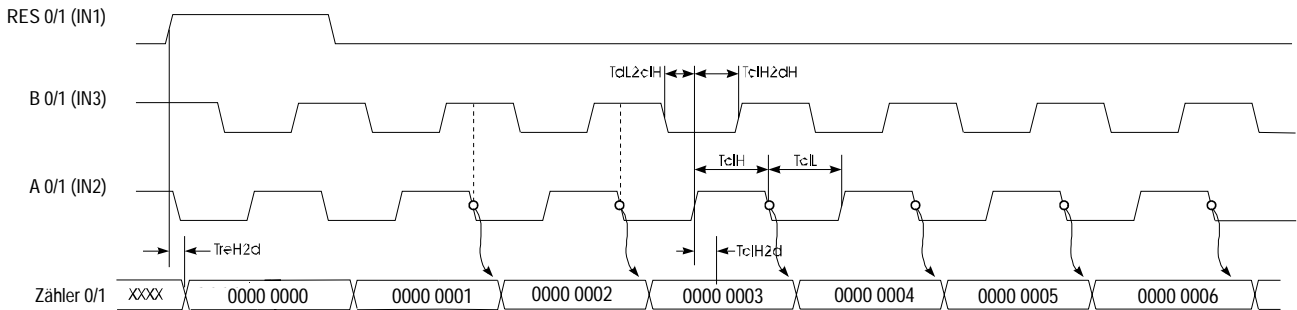
**Mode 1  
Encoder 1 Flanke**

Im Mode 1 können Sie für einen Kanal einen Encoder einstellen, der bei jeder fallenden Flanke entsprechend der Drehrichtung den internen Zähler um 1 inkrementiert bzw. dekrementiert. RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler. Bei Erreichen von Null wird der Ausgang OUT des entsprechenden Zählers mindestens für 100ms gesetzt auch, wenn der Zähler weiterzählt. Sollte der Zähler bei Null stoppen, bleibt der Ausgang gesetzt.

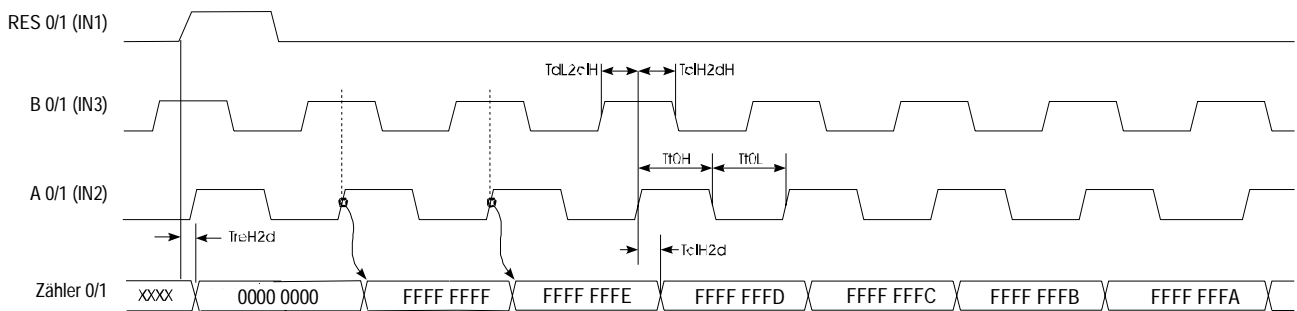
**Anschlussbelegung  
Zählerzugriff**



**Aufwärtszähler** Jede fallende Flanke an Eingang A inkrementiert den Zähler um 1, wenn zu diesem Zeitpunkt der Eingang B auf HIGH liegt.  
 Timing-Diagramm am Beispiel von Zähler 0/1:



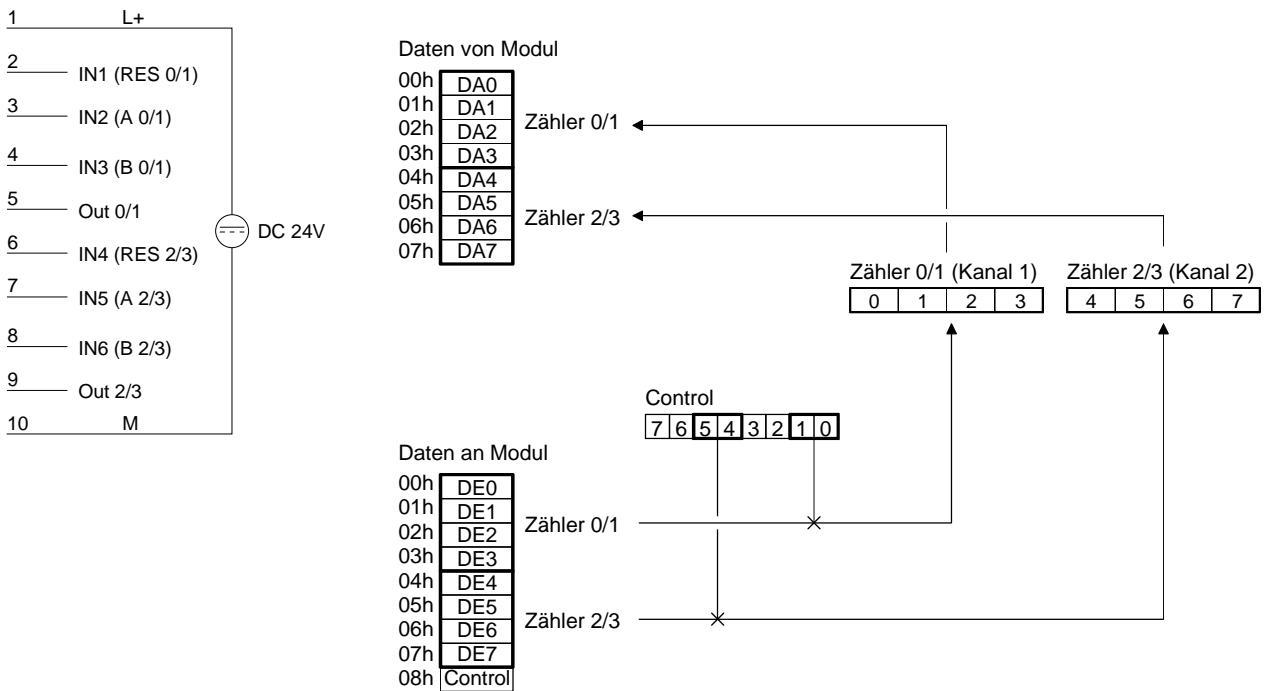
**Abwärtszähler** Jede steigende Flanke am Eingang A dekrementiert den internen Zähler um 1, wenn zu diesem Zeitpunkt der Eingang B auf HIGH liegt.  
 Timing-Diagramm am Beispiel von Zähler 0/1:



**Mode 3  
Encoder 2 Flanken**

Jede steigende bzw. fallende Flanke des Signals an Eingang A verändert den Zählerstand um 1. Die Zählrichtung ist vom aktuellen Pegelstand des Eingangs B abhängig. RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler. Bei Erreichen von Null wird der Ausgang OUT des entsprechenden Zählers mindestens für 100ms gesetzt auch wenn der Zähler weiterzählt. Sollte der Zähler bei Null stoppen, bleibt der Ausgang gesetzt.

**Anschlussbelegung  
Zählerzugriff**

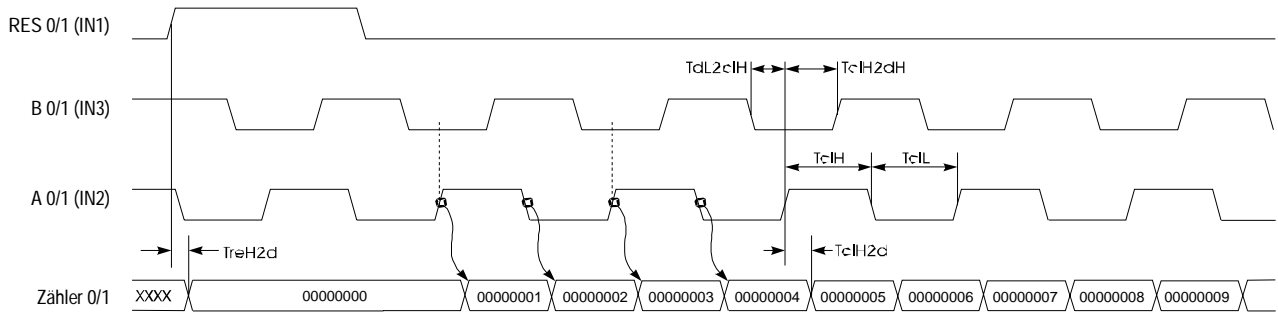




**Aufwärtszähler**

Der Zähler wird um 1 inkrementiert, wenn bei steigender Flanke des Signals A der Eingang B auf LOW liegt, oder bei fallender Flanke von A der Eingang B auf HIGH liegt.

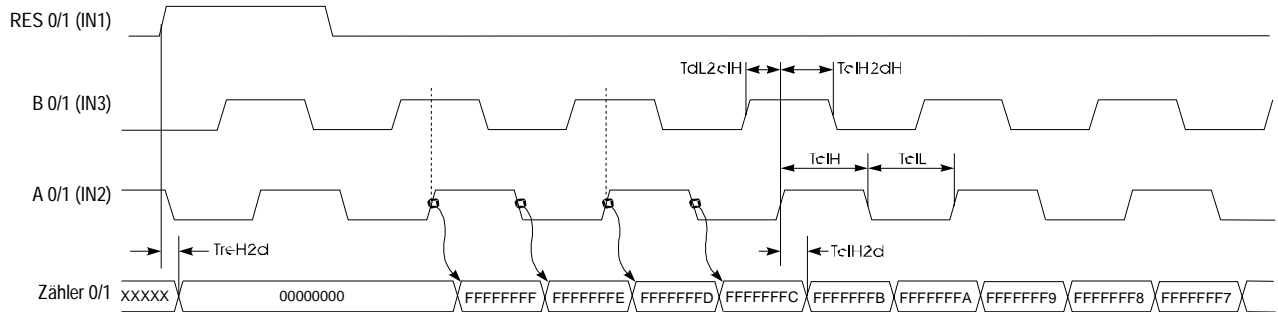
Timing-Diagramm am Beispiel von Zähler 0/1:



**Abwärtszähler**

Der Zähler wird um 1 dekrementiert, wenn bei steigender Flanke des Signals A der Eingang B auf HIGH liegt, oder bei fallender Flanke von A der Eingang B auf LOW liegt.

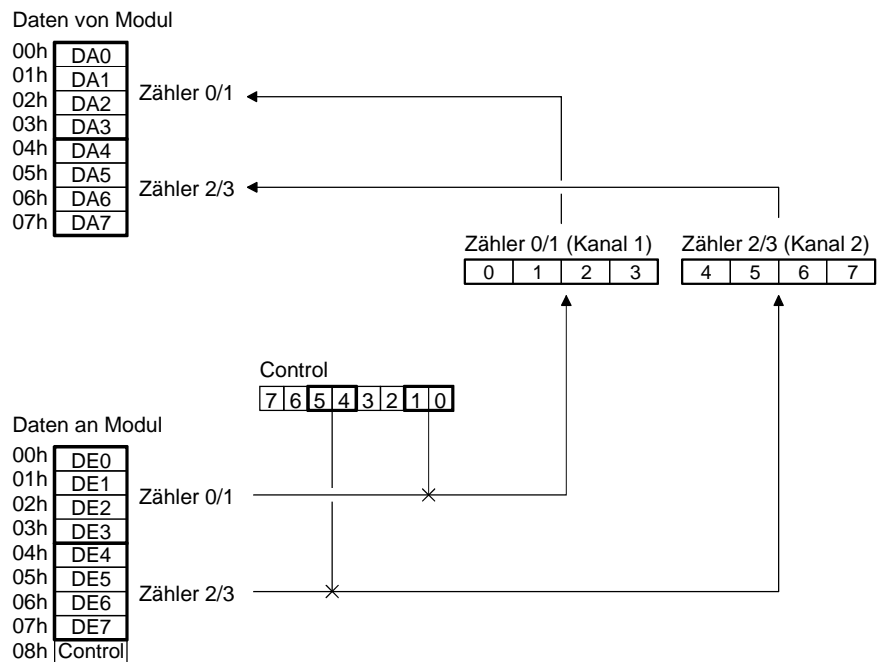
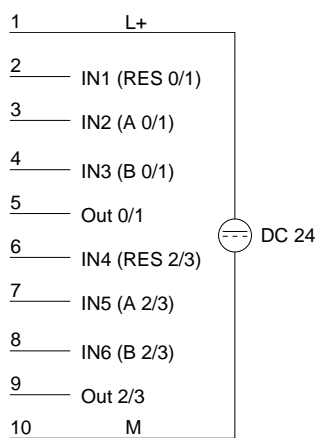
Timing-Diagramm am Beispiel von Zähler 0/1:



**Mode 5  
Encoder 4 Flanken**

Jede steigende bzw. fallende Flanke an einem der Eingänge A oder B verändert den Zählerstand um 1, wobei die Zählrichtung vom Pegel des anderen Eingangs (B bzw. A) abhängt. RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler. Bei Erreichen von Null wird der Ausgang OUT des entsprechenden Zählers mindestens für 100ms gesetzt, auch wenn der Zähler weiterzählt. Sollte der Zähler bei Null stoppen, bleibt der Ausgang gesetzt.

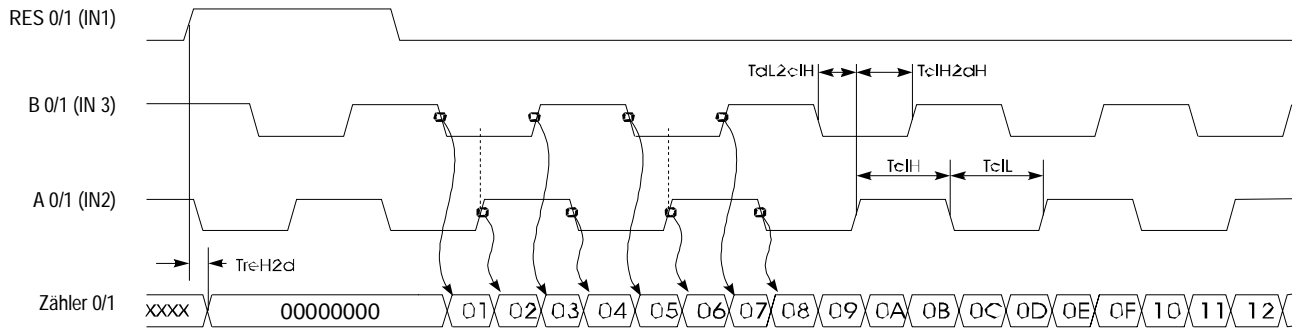
**Anschlussbelegung  
Zählerzugriff**



**Aufwärtszähler**

Der Zähler wird dann um 1 inkrementiert, wenn bei steigender Flanke an B der Eingang A auf HIGH liegt, bzw. wenn bei fallender Flanke an B der Eingang A auf LOW liegt, oder wenn bei steigender Flanke an A der Eingang B auf LOW liegt, bzw. bei fallender Flanke an A der Eingang B auf HIGH liegt.

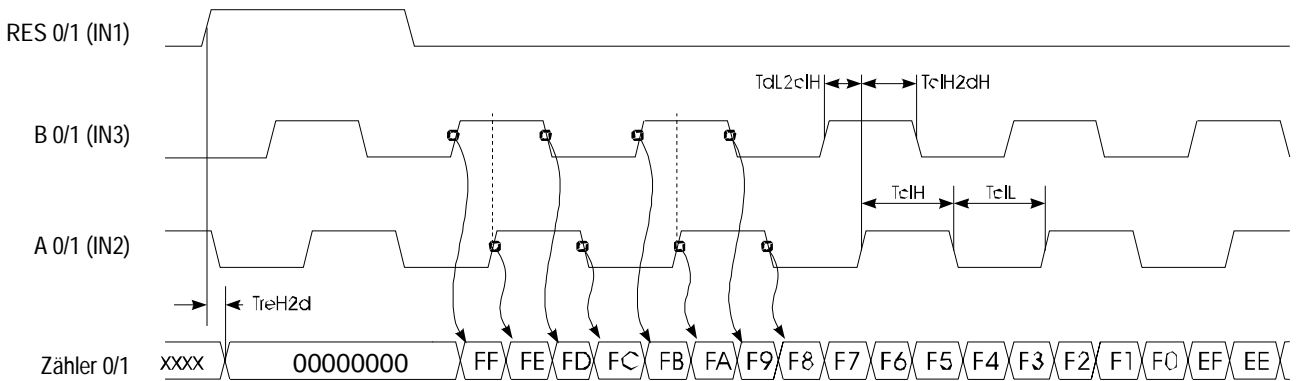
Timing-Diagramm am Beispiel von Zähler 0/1:



**Abwärtszähler**

Der Zähler wird dann um 1 dekrementiert, wenn bei steigender Flanke an B der Eingang A auf LOW liegt, bzw. wenn bei fallender Flanke an B der Eingang A auf HIGH liegt, oder wenn bei steigender Flanke an A der Eingang B auf HIGH liegt, bzw. bei fallender Flanke an A der Eingang B auf LOW liegt.

Timing-Diagramm am Beispiel von Zähler 0/1:



**Mode 6  
Pulsmessung,  
Pulse LOW, 50kHz  
mit Richtungs-  
angabe**

Die Pulsbreite eines am PULSE-Eingang eingespeisten Signals wird mit einer internen Zeitbasis gemessen. Der Messvorgang startet mit der fallenden Flanke des Eingangssignals und endet mit der steigenden. Die steigende Flanke des Messsignals speichert die Pulsbreite in der Einheit 20µs (entspricht  $f_{ref} = 50kHz$ ) als Ergebnis, welches dann ausgelesen werden kann.

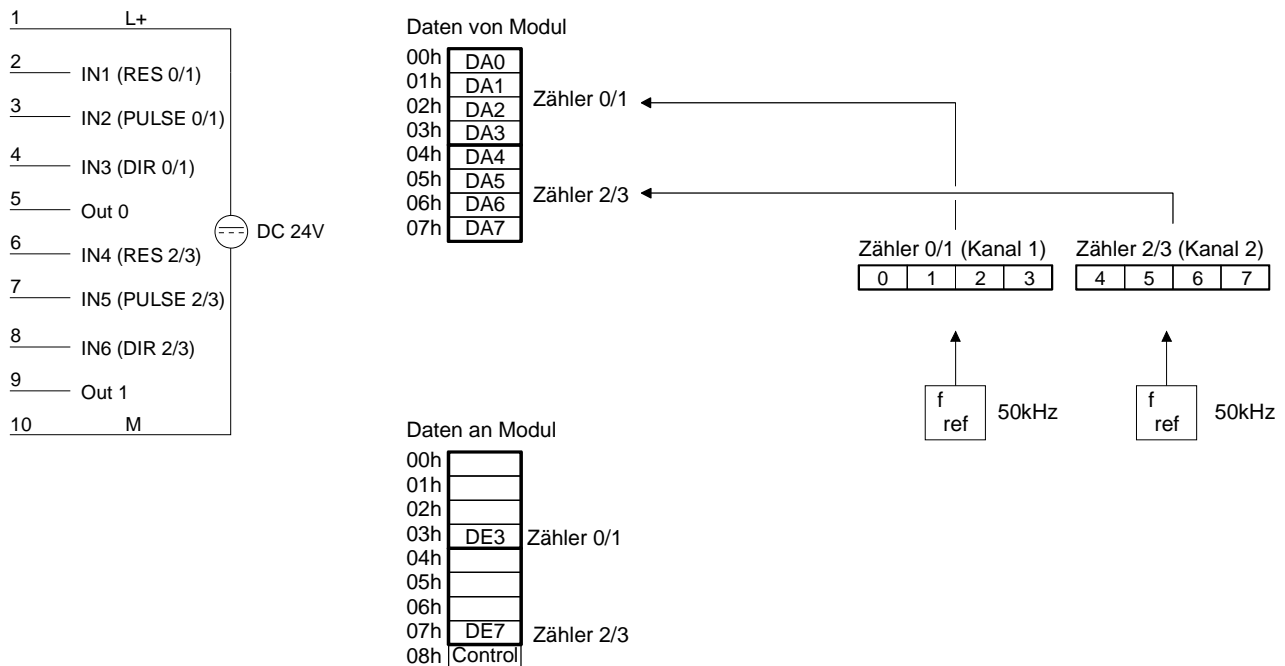
Der Eingang DIR bestimmt die Zählrichtung des Zählers. Wird DIR auf LOW gesetzt zählt der Messzähler aufwärts. Bei HIGH wird abwärts gezählt.

RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler.

Mit steigender Flanke des Signals Pulse steht ein Ergebnis im DA-Bereich zur Verfügung; das Ergebnis bleibt bis zum nächsten neuen Ergebnis stehen.

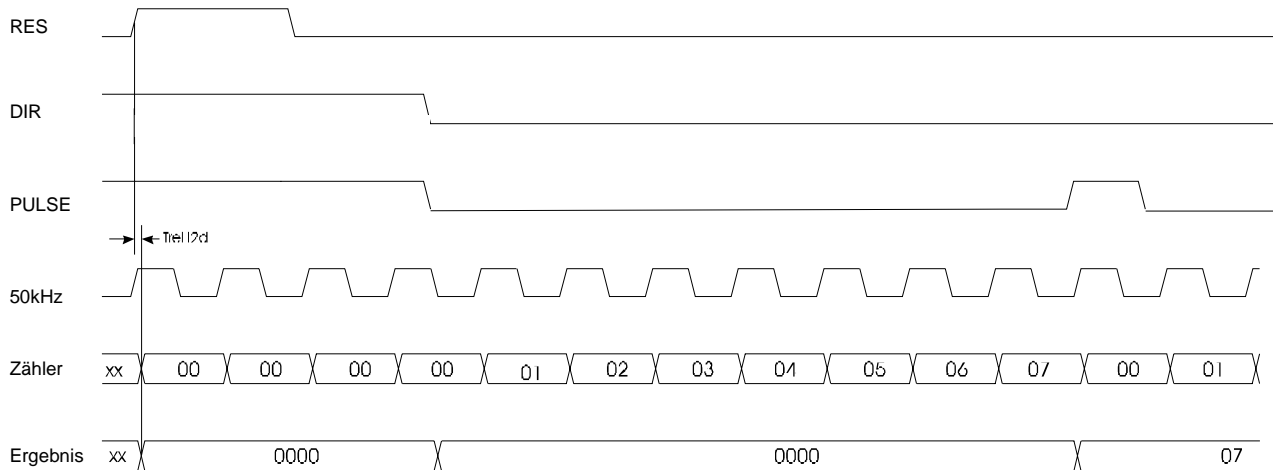
Die Signale Out 0 bzw. Out 1 werden nicht beeinflusst.

**Anschlussbelegung  
Zählerzugriff**



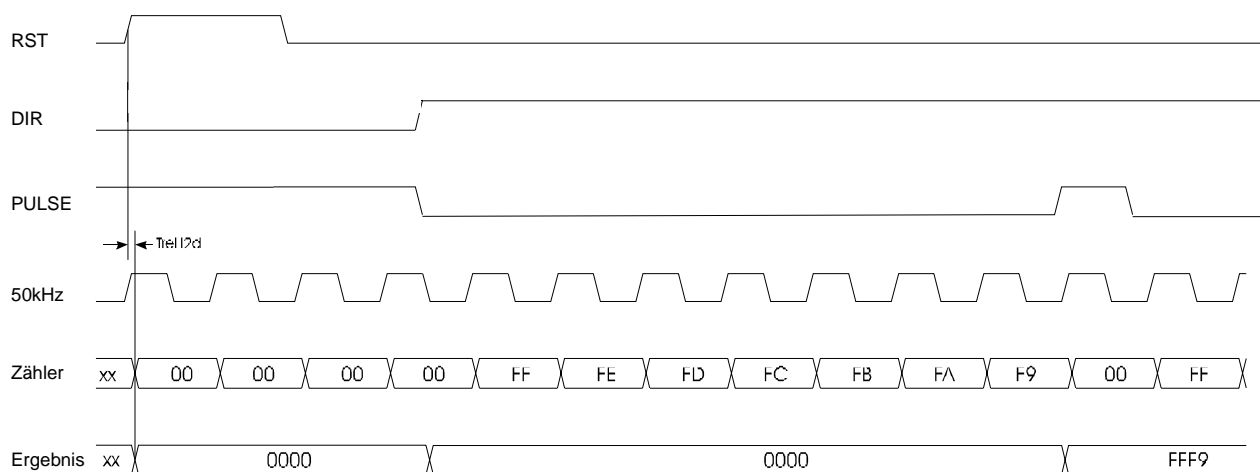
**Aufwärtszähler**

Das RES-Signal (R0) und das DIR-Signal (D0) werden auf Null gesetzt. Der Messvorgang wird nun mit der fallenden Flanke an PULSE (C0) gestartet und so der Zähler im 50kHz-Takt aufwärts gezählt. Mit der steigenden Flanke an PULSE (C0) wird der Zählvorgang beendet und der Zählwert in das Ergebnisregister übertragen. Das Ergebnisregister kann von der SPS gelesen werden. Der Wert bleibt solange im Ergebnisregister gespeichert, bis ein neuer Messvorgang beendet ist und dadurch das Register neu beschrieben wird.



**Abwärtszähler**

Das RES-Signal (R0) wird auf Null und das DIR-Signal (D0) wird auf HIGH gesetzt. Der Messvorgang wird nun mit der fallenden Flanke an PULSE (C0) gestartet und so der Zähler im 50kHz-Takt abwärts gezählt. Mit der steigenden Flanke an PULSE (C0) wird der Zählvorgang beendet und der Zählwert in das Ergebnisregister übertragen. Das Ergebnisregister kann von der SPS gelesen werden. Der Wert bleibt solange im Ergebnisregister gespeichert, bis ein neuer Messvorgang beendet ist und dadurch das Register neu beschrieben wird.

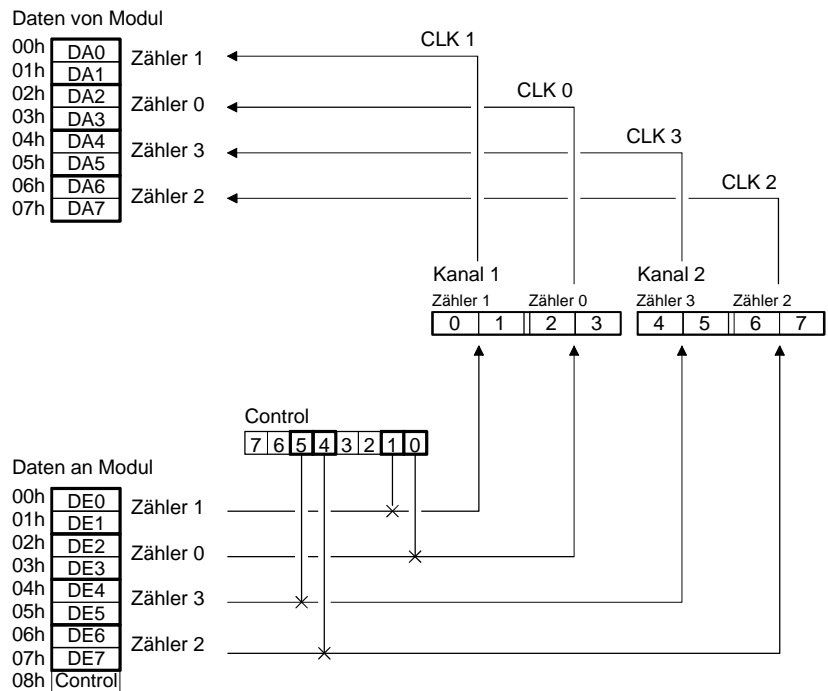
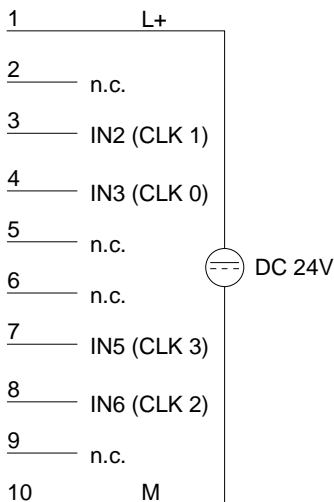


**Mode 8 ... 11  
Zählerfunktion mit  
zwei Eingängen**

In diesem Modus stehen je Kanal 2 Zähler mit je 16Bit Breite zur Verfügung. Mit jeder steigenden Flanke des Eingangstaktes CLK x wird der entsprechende Zähler hoch bzw. runtergezählt. Auch in diesem Modus kann jeder Zähler über ein Control-Bit mit einem Wert vorbelegt werden. Es werden keine Ausgänge angesteuert. Ein RESET steht nicht zur Verfügung. Folgende Kombinationen sind je Kanal möglich:

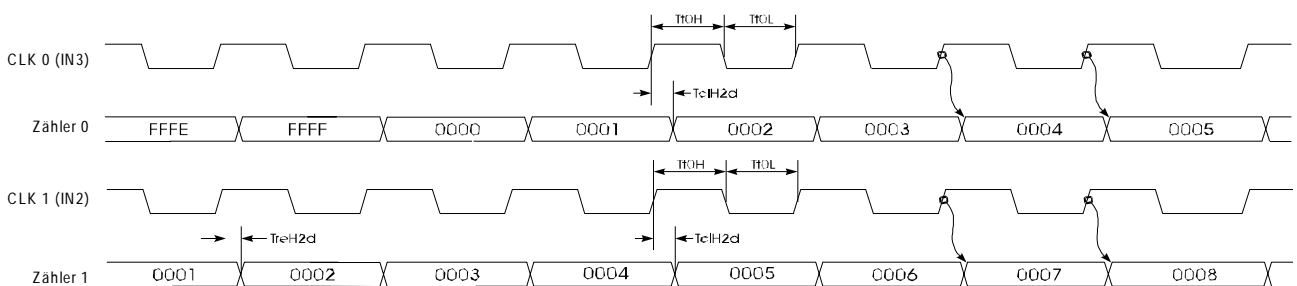
- Mode 8 - Zähler 0/2 aufwärts, Zähler 1/3 aufwärts**
- Mode 9 - Zähler 0/2 abwärts, Zähler 1/3 aufwärts**
- Mode 10 - Zähler 0/2 aufwärts, Zähler 1/3 abwärts**
- Mode 11 - Zähler 0/2 abwärts, Zähler 1/3 abwärts**

**Anschlussbelegung  
Zählerzugriff**



**Timing-Diagramm**

Nachfolgend ist ein Timing-Diagramm am Beispiel von Zähler 0 und Zähler 1 im Mode 8 aufgeführt:



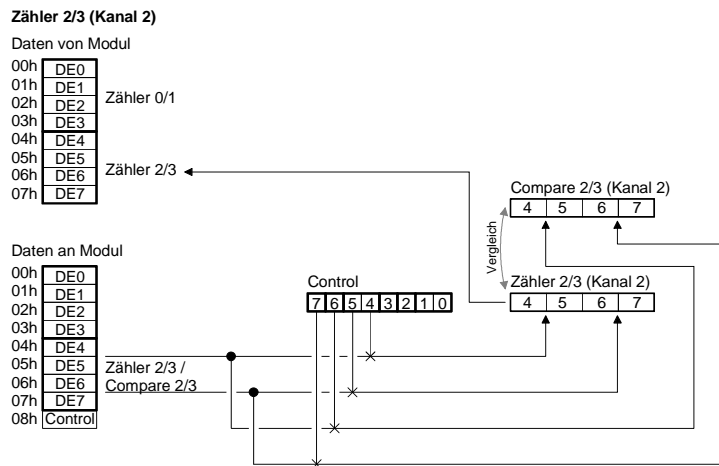
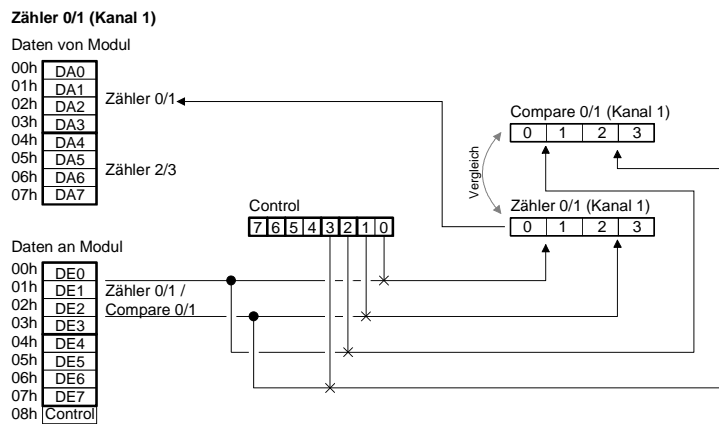
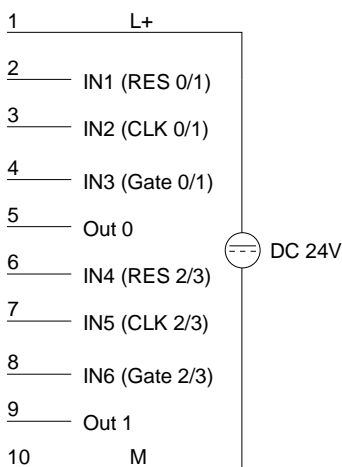
**Mode 12 und 13  
32Bit Zähler mit  
Gate**

Unter den Modi 12 und 13 können Sie je Kanal einen 32Bit Zähler realisieren, der über ein Torsignal (Gate) gesteuert wird. Die Zählrichtung hängt von dem eingestellten Modus ab. Mit jeder steigenden Flanke des Eingangstakts wird der Zähler um 1 inkrementiert bzw. dekrementiert, sofern Gate HIGH-Pegel hat. RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler. Bei Erreichen des in Compare geladenen Werts wird OUT für mindestens 100ms gesetzt wobei der Zähler weiterläuft.

**Mode 12 - 32Bit Zähler up + gate mit Compare**

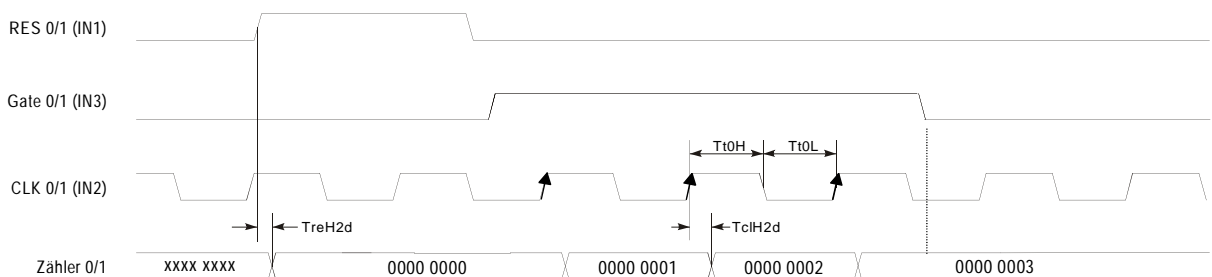
**Mode 13 - 32Bit Zähler down + gate mit Compare**

**Anschlussbelegung  
Zählerzugriff**



**Timing-Diagramm**

Nachfolgend ist ein Timing-Diagramm am Beispiel von Zähler 0/1 im Mode 12 aufgeführt:



**Mode 14 und 15  
32Bit Zähler mit  
Gate und Auto  
Reload**

Die Modi 14 und 15 bieten die gleiche Funktionalität wie die Modi 12 und 13, haben aber zusätzlich eine Auto-Reload-Funktion. Mit "Auto Reload" können Sie das Load-Register mit einem Wert belegen, der automatisch in den Zähler geladen wird, sobald dieser den eingestellten Compare-Wert erreicht.

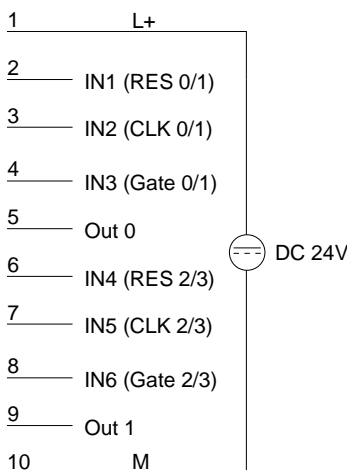
Ein HIGH-Impuls an RES setzt den Zähler auf 0000 0000. Mit einem HIGH-Pegel an GATE schalten Sie den Zähler frei, dass der Zähler mit jeder steigenden Flanke des CLK-Signals zählen kann. Solange Gate HIGH ist, zählt der Zähler mit jeder steigenden CLK-Flanke bis zu einem Wert, der mit dem nächsten Impuls den in Compare eingestellten Wert erreichen würde. Mit dem nächsten Impuls wird aber der Zählerstand mit dem im Load-Register eingestellten Wert überschrieben. Dies geschieht solange, bis GATE LOW-Pegel bekommt. Sobald ein Auto Reload erfolgt, wird der Zustand des zugehörigen Ausgangs geändert.

Das RES-Signal setzt nur den Zähler zurück, aber nicht die Ausgänge.

**Mode 14 - 32Bit Zähler up + gate mit Compare und Auto-Reload**

**Mode 15 - 32Bit Zähler down + gate mit Compare und Auto-Reload**

**Anschlussbelegung  
Zählerzugriff**



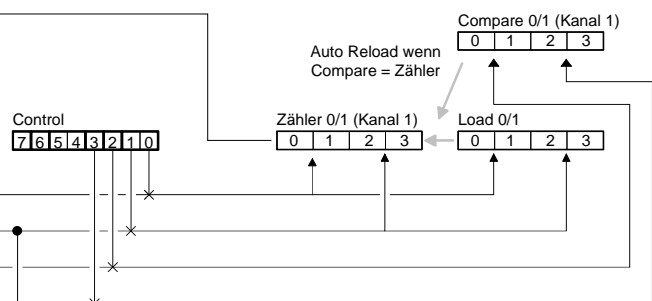
**Zähler 0/1 (Kanal 1)**

Daten von Modul

00h	DA0
01h	DA1
02h	DA2
03h	DA3
04h	DA4
05h	DA5
06h	DA6
07h	DA7

Daten an Modul

00h	DE0
01h	DE1
02h	DE2
03h	DE3
04h	DE4
05h	DE5
06h	DE6
07h	DE7
08h	Control



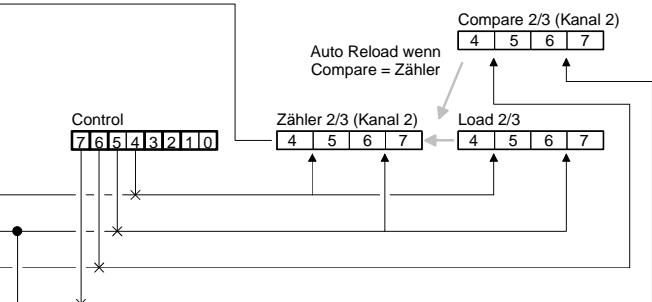
**Zähler 2/3 (Kanal 2)**

Daten von Modul

00h	DE0
01h	DE1
02h	DE2
03h	DE3
04h	DE4
05h	DE5
06h	DE6
07h	DE7

Daten an Modul

00h	DE0
01h	DE1
02h	DE2
03h	DE3
04h	DE4
05h	DE5
06h	DE6
07h	DE7
08h	Control





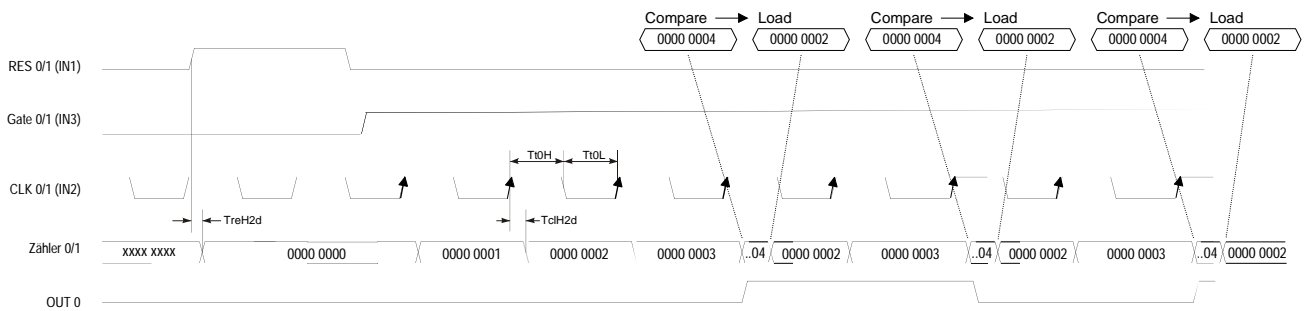
**Beispiel**

Das Beispiel soll die Funktionsweise der Zähler im Mode 14 und 15 nochmals verdeutlichen.

Ein HIGH-Impuls an RES setzt den Zähler auf 0000 0000. Mit einem HIGH-Pegel an GATE schalten Sie den Zähler frei. Solange GATE HIGH-Pegel hat, zählt der Zähler mit jeder steigenden Flanke an CLK bis zu einem Wert, der mit dem nächsten Impuls den in COMPARE eingestellten Wert erreichen würde. In diesem Beispiel zählt er bis 0000 0004 und führt sofort einen "Auto Reload" durch, d.h. der Zähler wird mit dem Inhalt des Load-Registers neu geladen (hier 0000 0002). Bei jedem Auto Reload ändert sich der Zustand des Ausgangs OUT 0.

In diesem Beispiel zählt der Zähler von 0000 0002 bis 0000 0004 solange GATE HIGH-Pegel hat.

Mit jedem Load-Vorgang wird der Zustand des Ausgangs OUT 0 geändert.



**Mode 16  
Frequenzmessung**

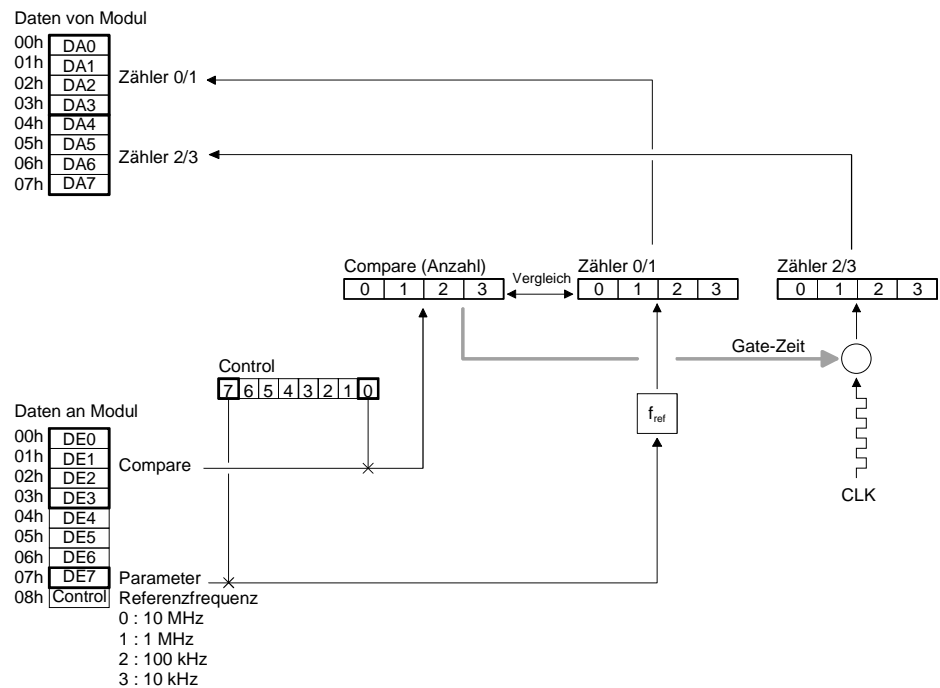
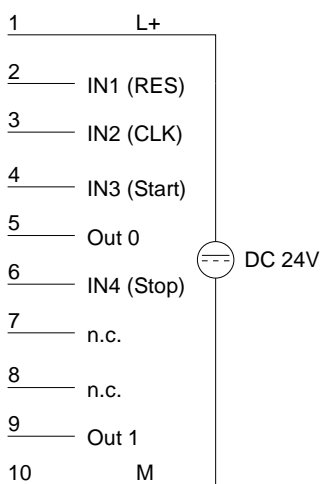
Mit diesem Mode bestimmen Sie die Frequenz eines Signals, das am CLK-Eingang eingespeist wird. Sie übergeben mit DE7 einen Referenztakt an Zähler 0/1 und bestimmen indirekt über die Anzahl n eine Gate-Zeit für deren Dauer Zähler 2/3 freigeschaltet wird. Die Anzahl n kann zwischen 1 und  $2^{32}-1$  liegen und wird in das Compare-Register geladen.

Während des Messvorgangs, der mit einer steigenden Flanke an Start beginnt, zählt Zähler 0/1 mit der ersten steigenden Flanke des CLK-Signals die Referenzimpulse, die der Referenztaktgenerator liefert.

Währenddessen zählt Zähler 2/3 jede steigende Flanke des CLK-Signals. Beide Zähler werden gestoppt, sobald der Zählerstand von Zähler 0/1 den Compare-Wert erreicht, oder wenn Stop HIGH-Pegel bekommt. Mit der unten aufgeführten Formel können Sie die Frequenz berechnen.

*Dieser Modus ist mit anderen Modi nicht kombinierbar!*

**Anschlussbelegung  
Zählerzugriff**



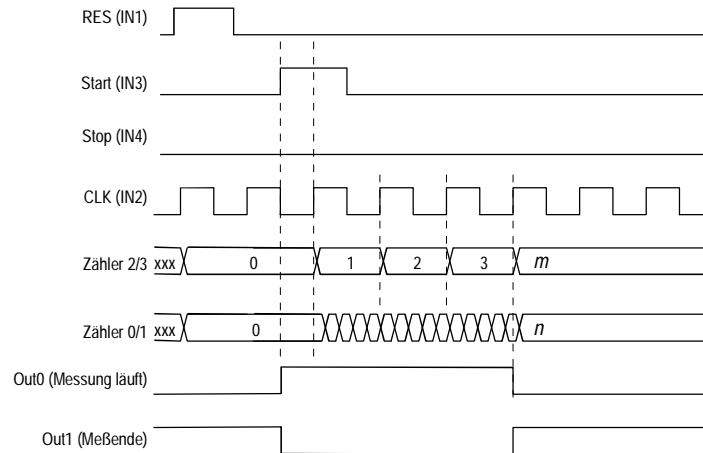
**Frequenz-  
berechnung**

Nach erfolgter Messung können Sie wie folgt die Frequenz berechnen:

$$Frequenz = \frac{fr \cdot m}{n}$$

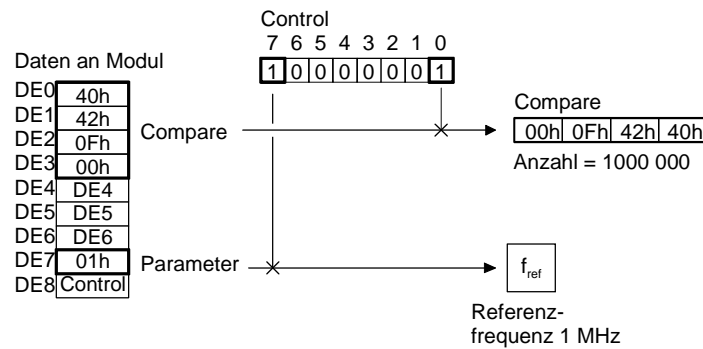
- mit  $fr$ : Referenzfrequenz (wird in DE7 mit dem Control-Bit 7 übergeben)
- $m$ : Inhalt Zähler 2/3 (Anzahl CLK-Impulse)
- $n$ : Anzahl Impulse der Referenzfrequenz in Zähler 0/1 (entspricht COMPARE, wenn nicht vorzeitig durch Stop abgebrochen wurde)

**Timing-Diagramm**



**Beispiel**

Anzahl = 1 000 000 Impulse  
Referenzfrequenz = 1 MHz



Bei einer Frequenz von 1MHz und einer Anzahl von 1 000 000 Impulsen entstehen 1Hz, d.h. nach einem Messvorgang beinhaltet Zähler 2/3 direkt die gesuchte Frequenz - eine Umrechnung entfällt.



**Hinweis!**

Wenn Sie  $f_r$  und  $n$  so wählen, dass in der Formel genau 1Hz entstehen, zeigt Zähler 2/3 direkt die Frequenz an.

**Mode 17  
Periodendauer-  
messung**

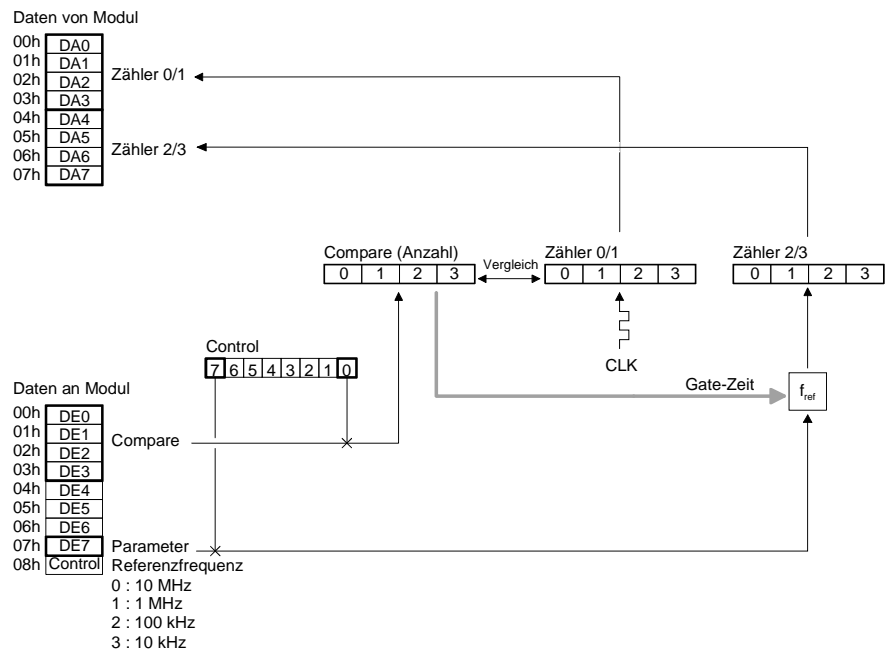
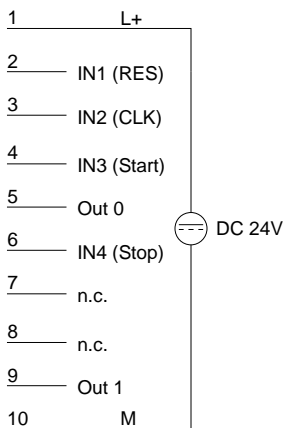
Mit diesem Mode bestimmen Sie die mittlere Periodendauer von n Messperioden eines Signals, das am CLK-Eingang eingespeist wird. Sie übergeben mit DE7 einen Referenztakt an Zähler 2/3 und bestimmen indirekt über die Anzahl n eine Gate-Zeit, für deren Dauer Zähler 2/3 freigeschaltet wird. Die Anzahl n kann zwischen 1 und  $2^{32}-1$  liegen und wird in das Compare-Register geladen.

Während des Messvorgangs, der mit einer steigenden Flanke an Start beginnt, zählt Zähler 2/3 mit der ersten steigenden Flanke des CLK-Signals die Referenzimpulse, die der Referenztaktgenerator liefert.

Währenddessen zählt Zähler 0/1 jede steigende Flanke des CLK-Signals. Beide Zähler werden gestoppt, sobald der Zählerstand von Zähler 0/1 den Compare-Wert erreicht, oder wenn Stop HIGH-Pegel bekommt. Mit der unten aufgeführten Formel können Sie die mittlere Periodendauer berechnen.

*Dieser Modus ist mit anderen Modi nicht kombinierbar!*

**Anschlussbelegung  
Zählerzugriff**

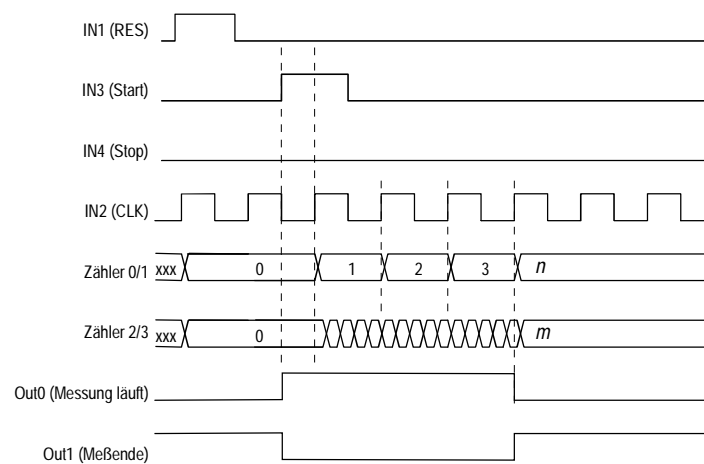


**Periodendauer-  
berechnung**

Nach erfolgter Messung können Sie wie folgt die mittlere Periodendauer berechnen:

$$Periodendauer = \frac{n}{fr \cdot m}$$

- mit *fr*: Referenzfrequenz (wird in DE7 mit dem Control-Bit 7 übergeben)  
*m*: Inhalt Zähler 2/3 (zählt Referenztakt-Impulse)  
*n*: Anzahl CLK-Impulse in Zähler 0/1 (entspricht Compare, wenn nicht durch Stop abgebrochen wurde)

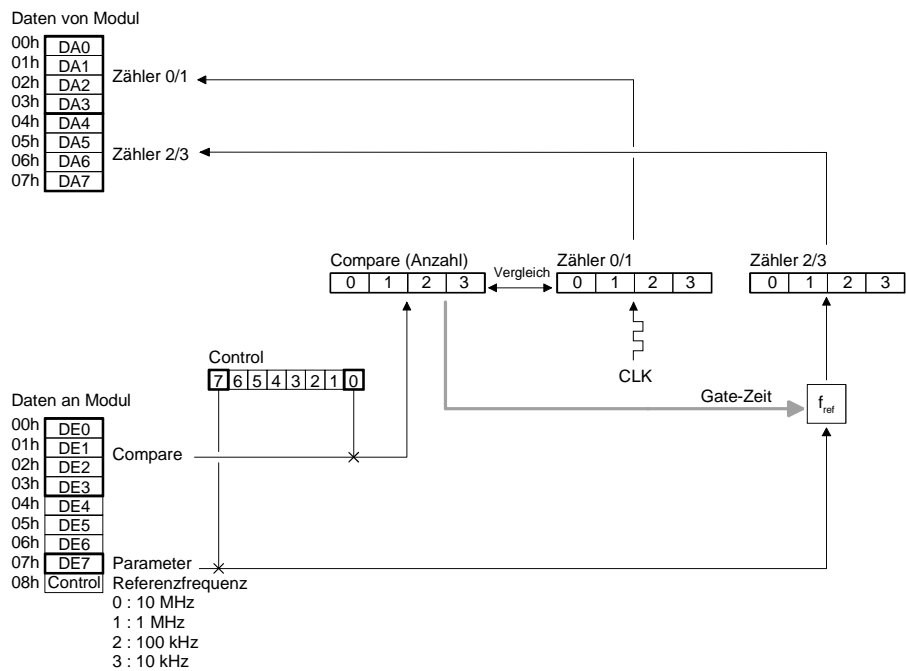
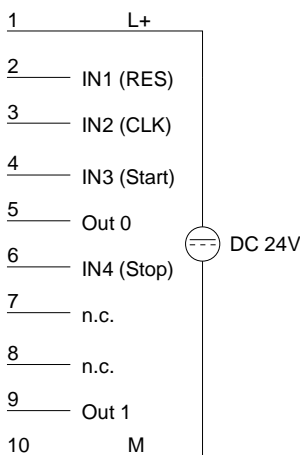
**Timing-Diagramm:**

**Mode 18  
Frequenzmessung  
mit Gate-Ausgang**

Der Modus 18 hat den gleichen Funktionsumfang wie Modus 16. Er unterscheidet sich aber in der Ansteuerung von OUT 0 und OUT 1. OUT 0 wird im Gegensatz zum Mode 16 erst aktiviert, wenn der Zählvorgang beginnt und deaktiviert, wenn der Zählvorgang endet, d.h. OUT 0 zeigt den Zustand des internen Gates an. OUT 1 zeigt den Gate-Zustand invers an.

*Dieser Modus ist mit anderen Modi nicht kombinierbar!*

**Anschlussbelegung  
Zählerzugriff**



**Frequenz-  
berechnung**

Nach erfolgter Messung können Sie wie folgt die Frequenz berechnen:

$$Frequenz = \frac{fr \cdot m}{n}$$

- mit *fr*: Referenzfrequenz (wird in DE7 mit dem Control-Bit 7 übergeben)
- m*: Inhalt Zähler 2/3 (Anzahl CLK-Impulse)
- n*: Anzahl Impulse der Referenzfrequenz in Zähler 0/1 (entspricht Compare, wenn nicht vorzeitig durch Stop abgebrochen wurde)

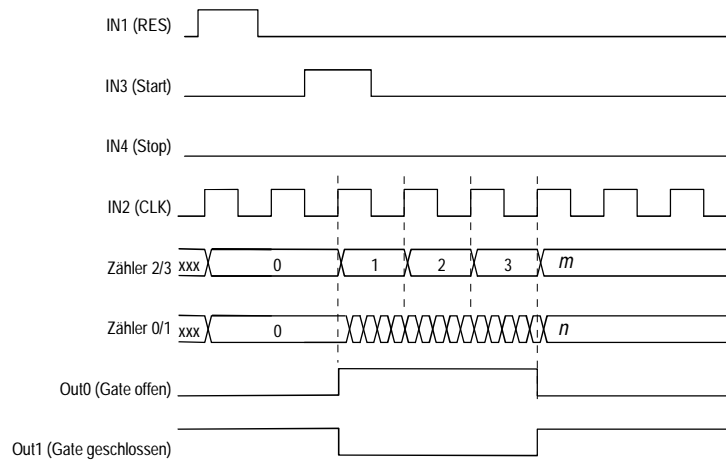


**Hinweis!**

Wenn Sie  $f_r$  und  $n$  so wählen, dass in der Formel genau 1Hz entstehen, zeigt Zähler 2/3 direkt die Frequenz an.

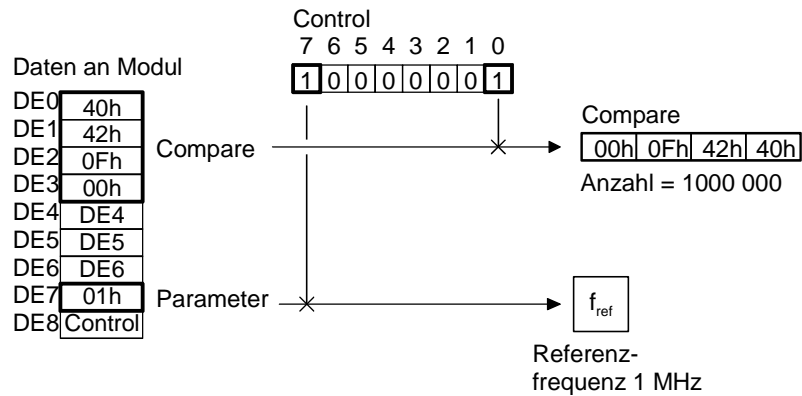
Z.B. bei einer Frequenz von 1MHz und einer Anzahl von 1 000 000 Impulsen entstehen 1Hz, d.h. nach einem Messvorgang beinhaltet Zähler 2/3 direkt die gesuchte Frequenz - eine Umrechnung entfällt.

**Timing-Diagramm:**



**Beispiel**

Anzahl = 1 000 000 Impulse  
 Referenzfrequenz = 1MHz

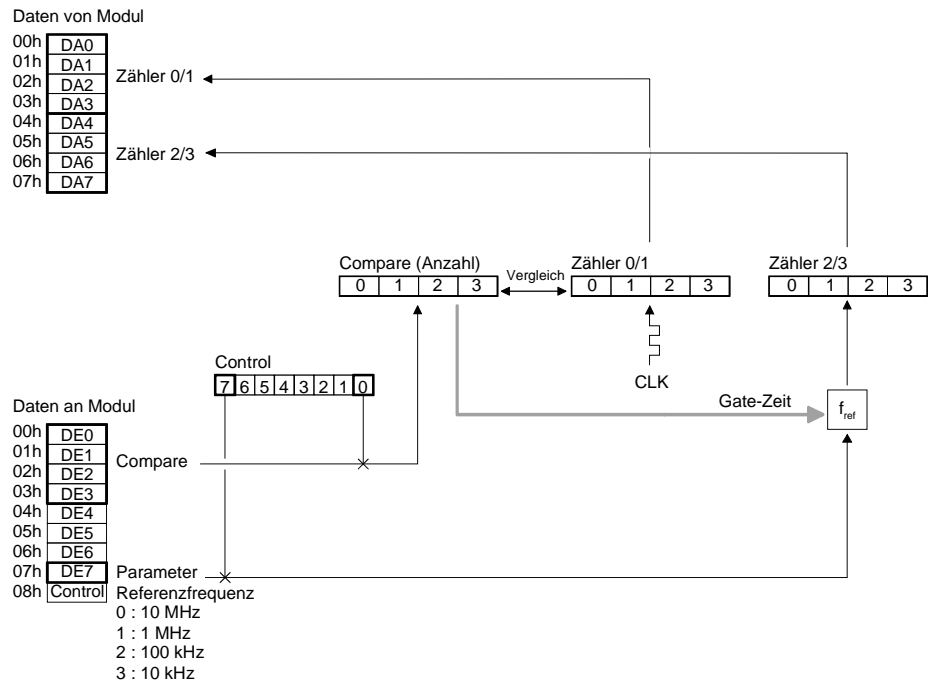
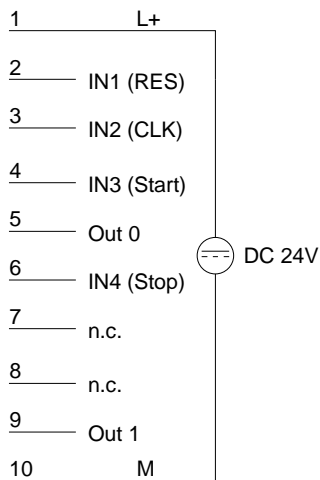


**Mode 19**  
**Periodendauer-**  
**messung mit**  
**Gate-Ausgang**

Der Mode 19 hat den gleichen Funktionsumfang wie Mode 17. Er unterscheidet sich aber in der Ansteuerung von OUT 0 und OUT 1. OUT 0 wird im Gegensatz zum Mode 17 erst aktiviert, wenn der Zählvorgang beginnt und deaktiviert, wenn der Zählvorgang endet, d.h. OUT 0 zeigt den Zustand des internen Gates an. OUT 1 zeigt den Gate-Zustand invers an.

*Dieser Modus ist mit anderen Modi nicht kombinierbar!*

**Anschlussbelegung**  
**Zählerzugriff**





**Periodendauer-  
berechnung**

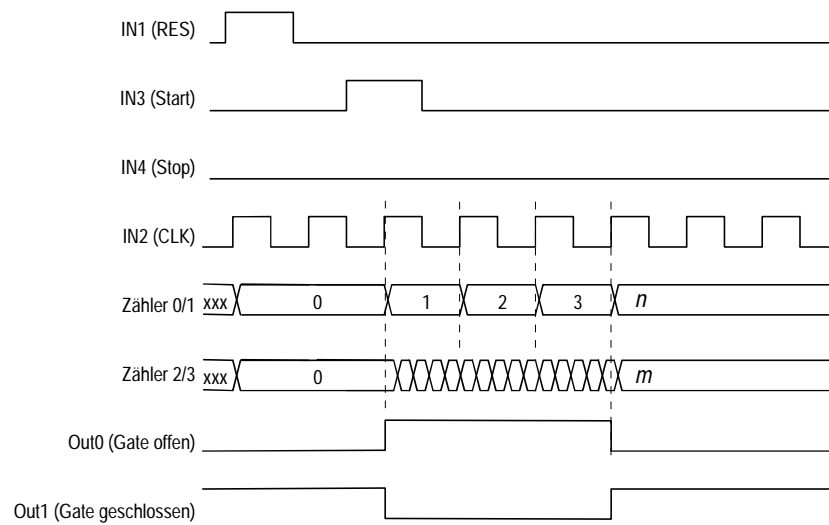
Nach erfolgter Messung können Sie wie folgt die mittlere Periodendauer berechnen:

$$\text{Periodendauer} = \frac{n}{fr \cdot m}$$

mit  $fr$ : Referenzfrequenz (wird in DE7 mit dem Control-Bit 7 übergeben)

$m$ : Inhalt Zähler 2/3 (zählt Referenztakt-Impulse)

$n$ : Anzahl CLK-Impulse in Zähler 0/1 (entspricht COMPARE, wenn nicht durch Stop abgebrochen wurde)

**Timing-Diagramm:**

**Mode 20  
Pulsmessung,  
Pulse down**

**prog. Zeitbasis, mit Richtungsangabe**

Die Pulsbreite eines am PULSE-Eingang eingespeisten Signals wird mit einer internen Zeitbasis gemessen. Der Messvorgang startet mit der fallenden Flanke des Eingangssignals und endet mit der steigenden. Die steigende Flanke des Messsignals speichert die Pulsbreite in der Einheit 1/Fref als Ergebnis, das dann ausgelesen werden kann.

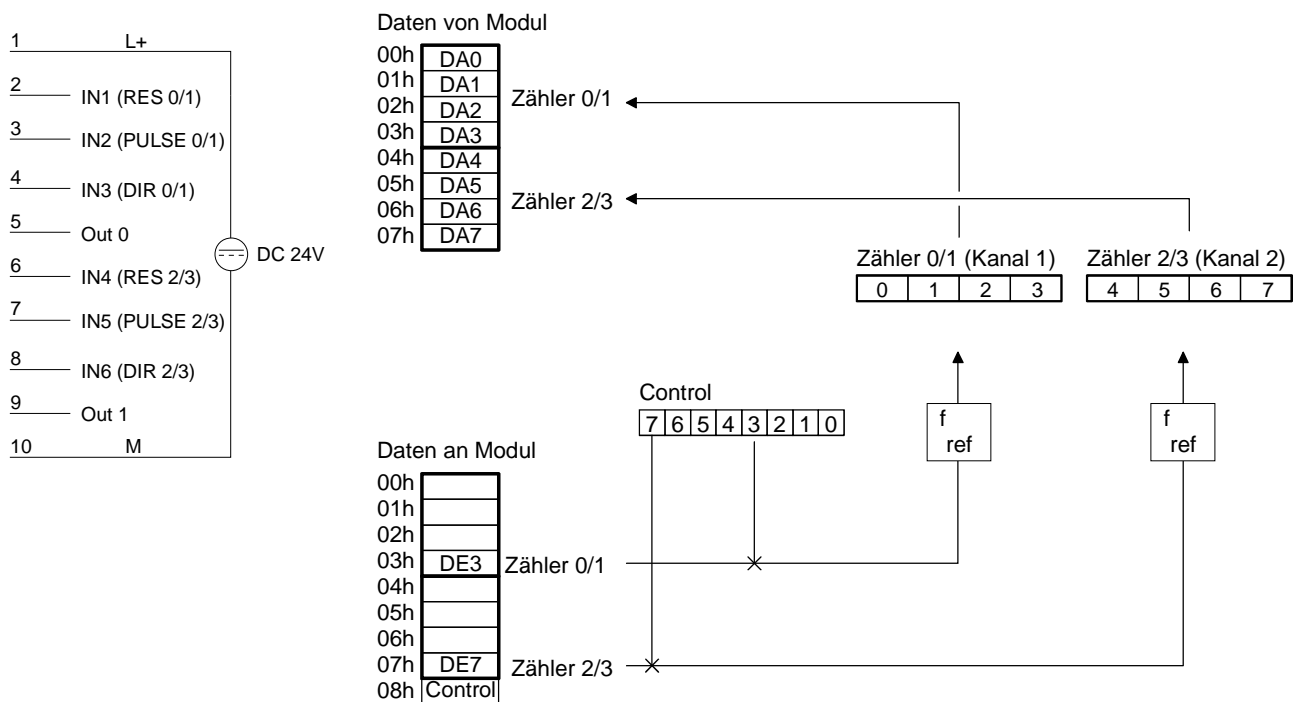
Der Eingang DIR bestimmt die Zählrichtung des Zählers. Wird DIR auf LOW gesetzt, zählt der Messzähler aufwärts. Bei HIGH wird abwärts gezählt.

RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler.

Fref ist programmierbar.

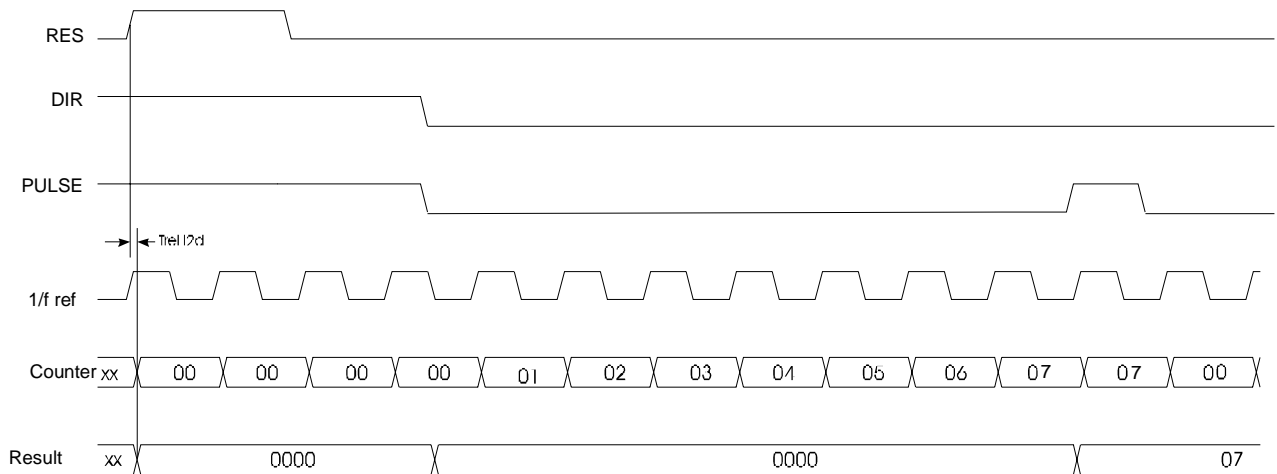
Das Signal OUT wird nicht beeinflusst.

**Anschlussbelegung  
Zählerzugriff**



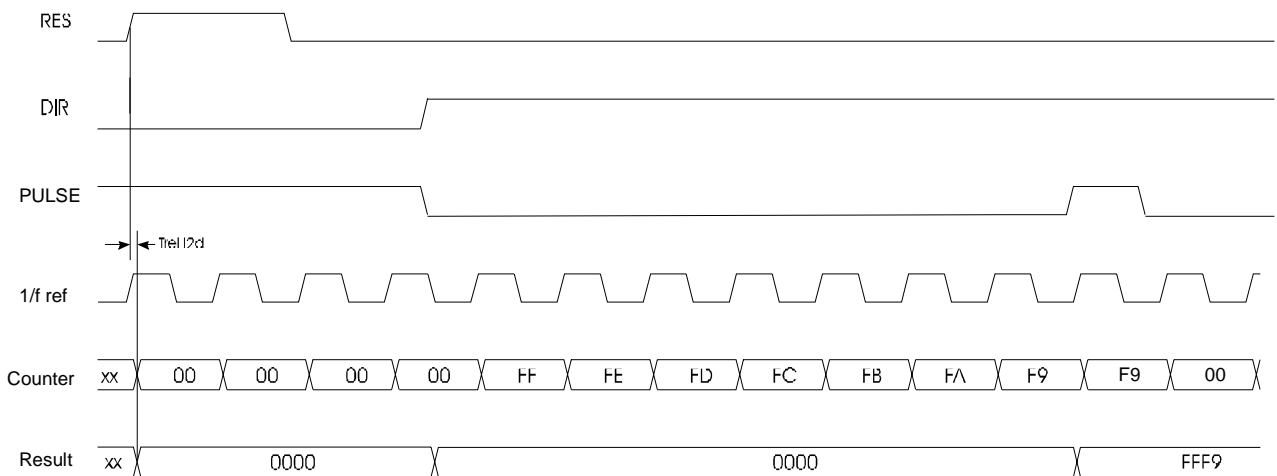
**Aufwärtszähler**

Das RES-Signal (R0) und das DIR-Signal (D0) werden auf Null gesetzt. Der Messvorgang wird nun mit der fallenden Flanke an PULSE (C0) gestartet und so der Zähler in der eingestellten Zeitbasis aufwärts gezählt. Mit der steigenden Flanke an PULSE (C0) wird der Zählvorgang beendet und der Zählwert in das Ergebnisregister übertragen. Das Ergebnisregister kann von der SPS gelesen werden. Der Wert bleibt solange im Ergebnisregister gespeichert, bis ein neuer Messvorgang beendet ist und dadurch das Register neu beschrieben wird.



**Abwärtszähler**

Das RES-Signal (R0) wird auf Null und das DIR-Signal (D0) wird auf HIGH gesetzt. Der Messvorgang wird nun mit der fallenden Flanke an PULSE (C0) gestartet und so der Zähler in der eingestellten Zeitbasis abwärts gezählt. Mit der steigenden Flanke an PULSE (C0) wird der Zählvorgang beendet und der Zählwert in das Ergebnisregister übertragen. Das Ergebnisregister kann von der SPS gelesen werden. Der Wert bleibt solange im Ergebnisregister gespeichert, bis ein neuer Messvorgang beendet ist und dadurch das Register neu beschrieben wird.



**Mode 21**  
**Pulsmessung,**  
**Pulse low**

**Zählrichtung aufwärts, prog. Zeitbasis, mit Freigabe**

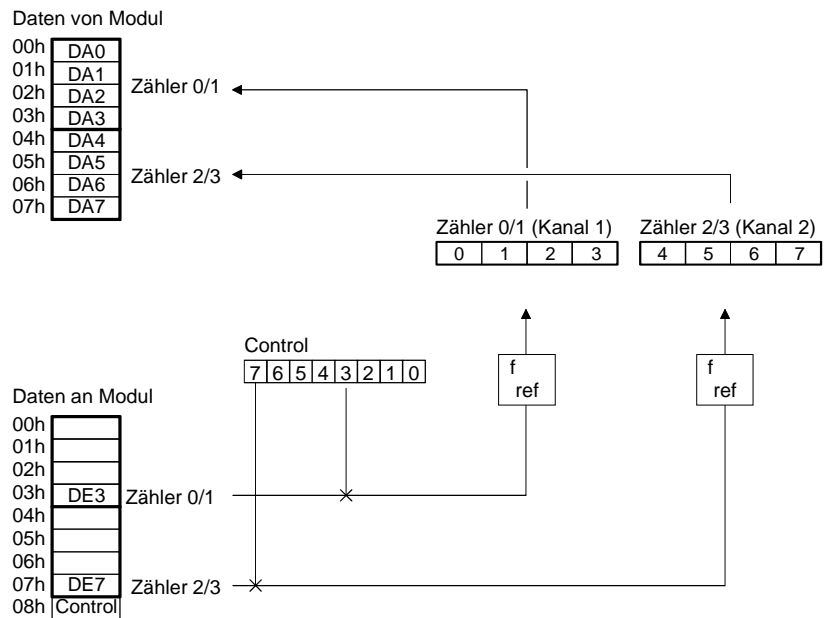
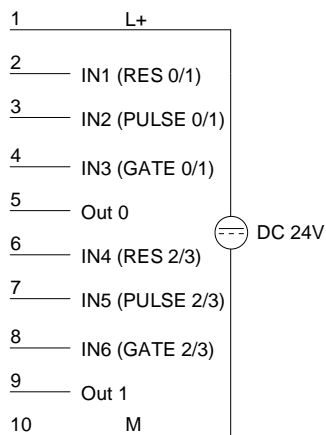
Die Pulsbreite eines am PULSE-Eingang eingespeisten Signals wird mit einer programmierbaren Zeitbasis (fref) gemessen. Der Messvorgang startet mit der fallenden Flanke des Eingangssignals und endet mit der steigenden. Die steigende Flanke des Messsignals speichert die Pulsbreite in der Einheit 1/fref als Ergebnis, welches dann ausgelesen werden kann.

Die Funktion wird nur ausgeführt bei HIGH-Signal am GATE-Eingang.

RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler.

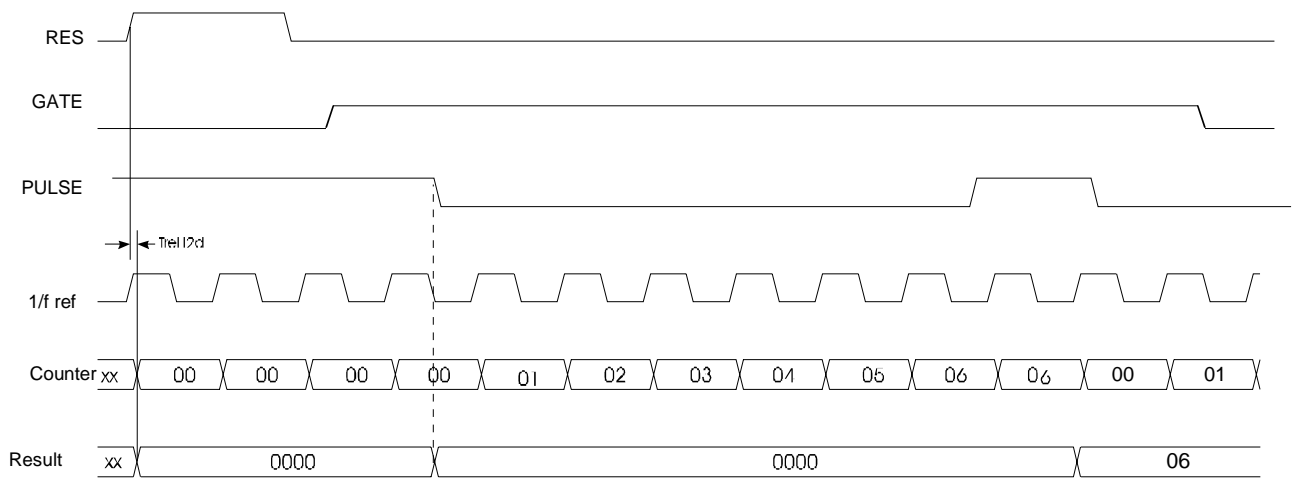
Das Signal OUT wird nicht beeinflusst.

**Anschlussbelegung**  
**Zählerzugriff**



**Aufwärtszähler**

Das RES-Signal (R0) wird auf Null gesetzt. Der Messvorgang kann nur gestartet werden, wenn das GATE-Signal auf HIGH gesetzt ist. Der Messvorgang wird nun mit der fallenden Flanke an PULSE (C0) gestartet und so der Zähler in der eingestellten Zeitbasis aufwärts gezählt. Mit der steigenden Flanke an PULSE (C0) wird der Zählvorgang beendet und der Zählwert in das Ergebnisregister übertragen. Das Ergebnisregister kann von der SPS gelesen werden. Der Wert bleibt solange im Ergebnisregister gespeichert, bis ein neuer Messvorgang beendet ist und dadurch das Register neu beschrieben wird. Es ist darauf zu achten, dass das GATE-Signal über den ganzen Messzyklus HIGH ist, da sonst die Messung nicht beendet wird.



**Mode 22**  
**Pulsmessung,**  
**Pulse high**

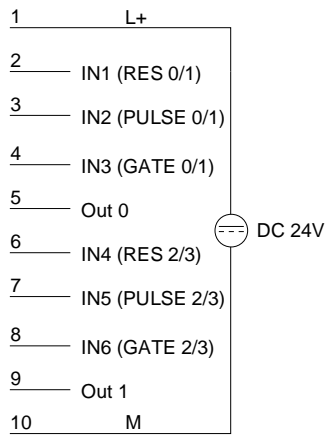
**Zählrichtung abwärts, prog. Zeitbasis, mit Freigabe**

Die Pulsbreite eines am PULSE-Eingang eingespeisten Signals wird mit einer programmierbaren Zeitbasis (fref) gemessen. Die steigende Flanke des Messsignals speichert die Pulsbreite in der Einheit 1/fref als Ergebnis, welches dann ausgelesen werden kann.

Die Funktion wird nur ausgeführt bei HIGH-Signal am GATE-Eingang  
 RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler.

Das Signal OUT wird nicht beeinflusst.

**Anschlussbelegung**  
**Zählerzugriff**



Daten von Modul

00h	DA0
01h	DA1
02h	DA2
03h	DA3
04h	DA4
05h	DA5
06h	DA6
07h	DA7

Zähler 0/1

Zähler 2/3

Zähler 0/1 (Kanal 1)				Zähler 2/3 (Kanal 2)			
0	1	2	3	4	5	6	7

Daten an Modul

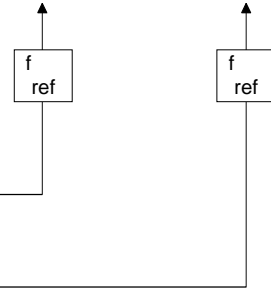
00h	
01h	
02h	
03h	DE3
04h	
05h	
06h	
07h	DE7
08h	Control

Control

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

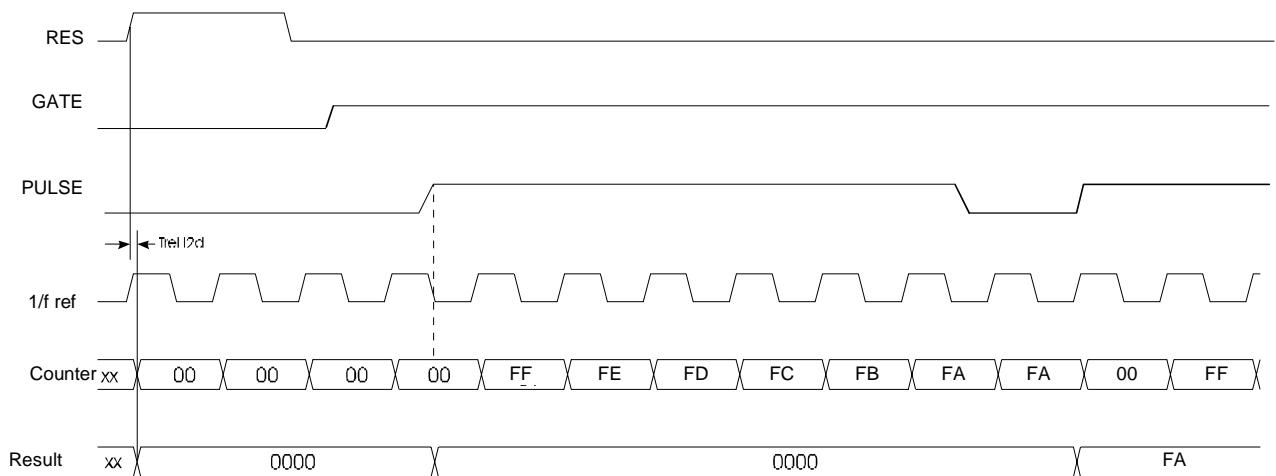
Zähler 0/1

Zähler 2/3



**Abwärtszähler**

Das RES-Signal (R0) wird auf Null gesetzt. Der Messvorgang kann nur gestartet werden, wenn das GATE-Signal auf HIGH gesetzt ist. Der Messvorgang wird nun mit der steigenden Flanke an PULSE (C0) gestartet und so der Zähler in der eingestellten Zeitbasis abwärts gezählt. Mit der fallenden Flanke an PULSE (C0) wird der Zählvorgang beendet und der Zählwert in das Ergebnisregister übertragen. Das Ergebnisregister kann von der SPS gelesen werden. Der Wert bleibt solange im Ergebnisregister gespeichert, bis ein neuer Messvorgang beendet ist, und dadurch das Register neu beschrieben wird. Es ist darauf zu achten, dass das GATE-Signal über den ganzen Messzyklus HIGH ist, da sonst die Messung nicht beendet wird.

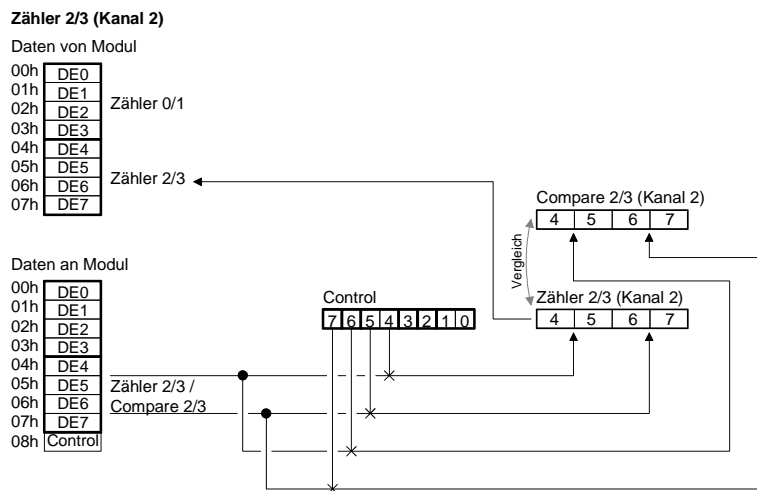
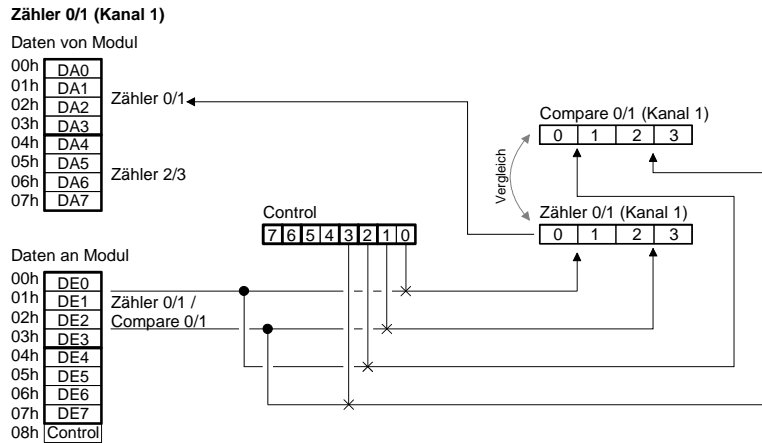
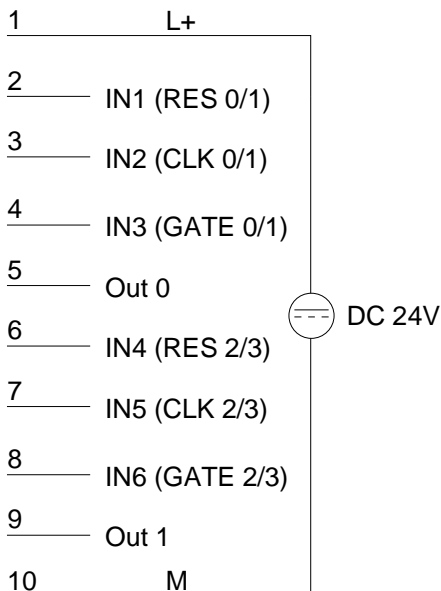


**Mode 23**  
**One Shot, Zähl-  
 richtung aufwärts,  
 mit Freigabe,  
 Ausgabesignal**

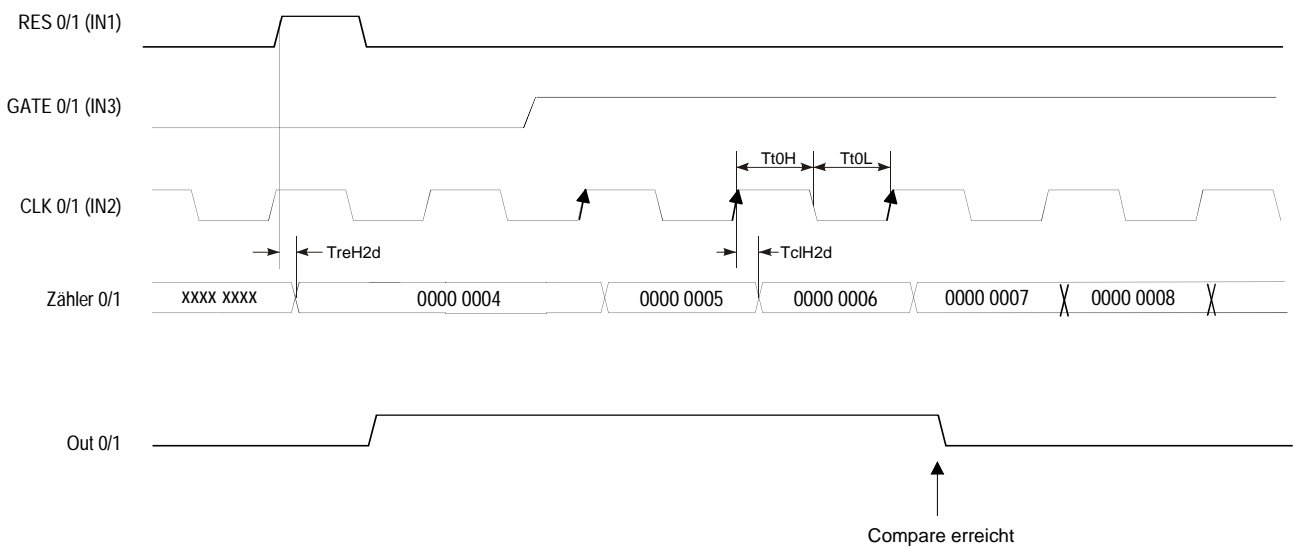
Unter dem Mode 23 können Sie je Kanal einen 32Bit Zähler realisieren, der über ein Torsignal (Gate) gesteuert wird. Mit jeder steigenden Flanke des Eingangstakts wird der Zähler um 1 aufwärts gezählt, sofern Gate HIGH-Pegel hat. RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler. Out wird mit Laden des Zählers HIGH. Bei Erreichen des in COMPARE geladenen Werts wird OUT gelöscht. Nach Erreichen des COMPARE-Wertes läuft der Zähler weiter.

**Mode 23 - One Shot, up mit Gate-Input, Output set**

**Anschlussbelegung  
 Zählerzugriff**





**Timing-Diagramm** Beispiel von Zähler 0/1 im Mode 23:

1. Das Signal RES wird LOW.
2. COMPARE einmalig laden.
3. Der Zähler (über Control gesteuert) wird geladen z.B. mit 0004.
4. Das Signal GATE ist aktiv.

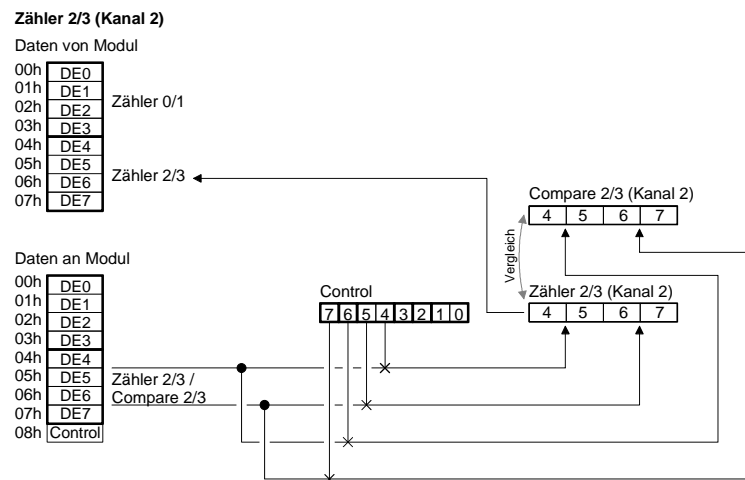
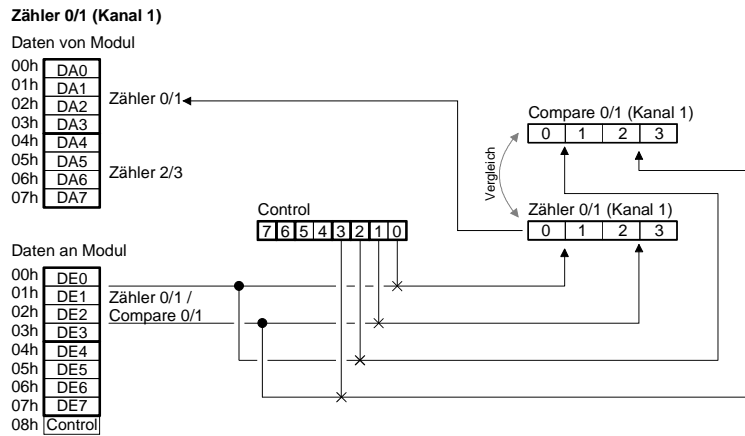
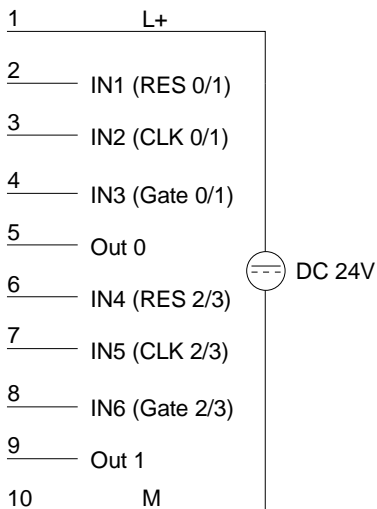
Stopp mit CONTROL = Abbruch

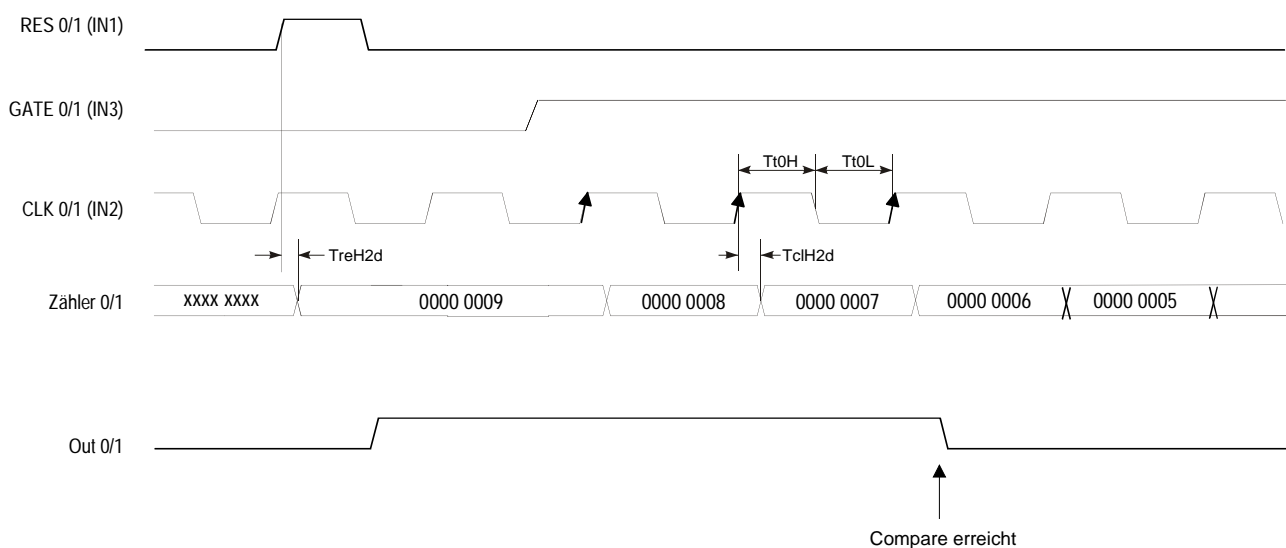
**Mode 24**  
**One Shot, Zähl-  
 richtung abwärts,  
 mit Freigabe,  
 Ausgabesignal**

Unter dem Mode 24 können Sie je Kanal einen 32Bit Zähler realisieren, der über ein Torsignal (Gate) gesteuert wird. Mit jeder steigenden Flanke des Eingangstakts wird der Zähler um 1 abwärts gezählt, sofern GATE HIGH-Pegel hat. RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler. OUT wird mit Laden des Zählers HIGH. Bei Erreichen des in COMPARE geladenen Werts wird OUT gelöscht. Nach Erreichen des COMPARE-Wertes läuft der Zähler weiter.

**Mode 24 - One Shot, down mit Gate-Input, Output set**

**Anschlussbelegung  
 Zählerzugriff**



**Timing-Diagramm** Beispiel von Zähler 0/1 im Mode 24:

1. Das Signal RES wird LOW.
2. COMPARE einmalig laden.
3. Der Zähler (über Control gesteuert) wird geladen z.B. mit 0009.
4. Das Signal GATE ist aktiv.

Stopp mit CONTROL = Abbruch

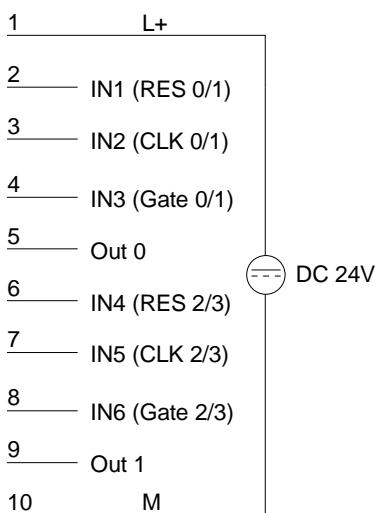
**Mode 25**

**One Shot, Zähl-  
richtung aufwärts,  
mit  
Rücksetzsignal**

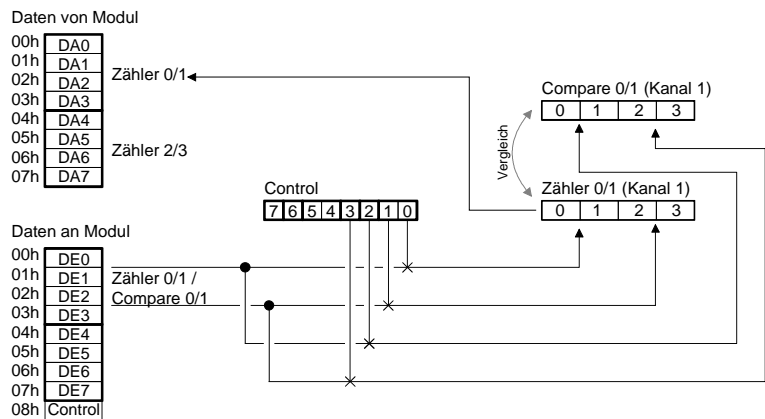
Unter dem Mode 25 können Sie je Kanal einen 32 Bit Zähler realisieren, der über ein Torsignal (Gate) gesteuert wird. Mit jeder steigenden Flanke des Eingangstakts wird der Zähler um 1 aufwärts gezählt, sofern Gate HIGH-Pegel hat. RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler. Das Signal OUT (aktiv mit 0) wird mit dem Laden des Zählers LOW; bei Erreichen des in COMPARE geladenen Wertes wird OUT wieder HIGH.

**Mode 25 One Shot, count up, Reset**

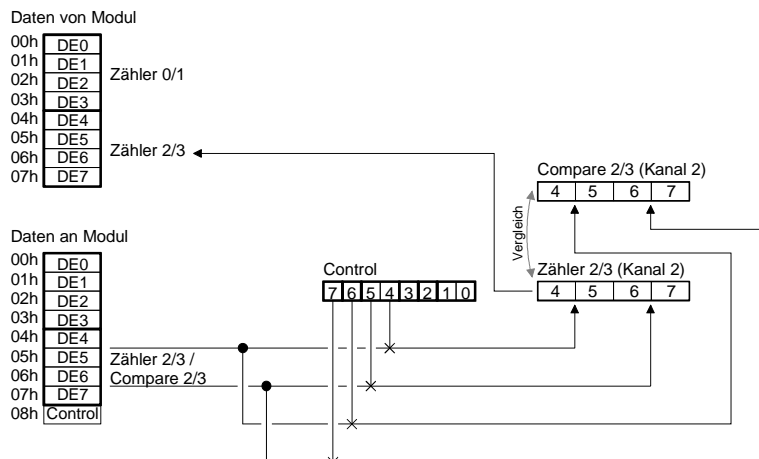
**Anschlussbelegung  
Zählerzugriff**



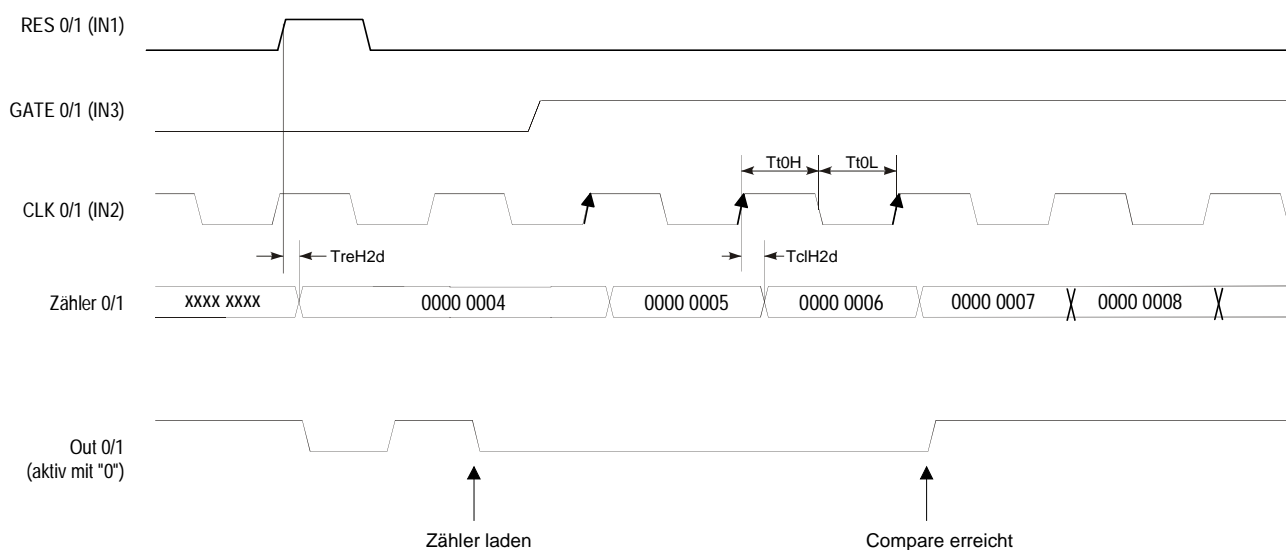
**Zähler 0/1 (Kanal 1)**



**Zähler 2/3 (Kanal 2)**



**Timing-Diagramm** Beispiel von Zähler 0/1 im Mode 25:

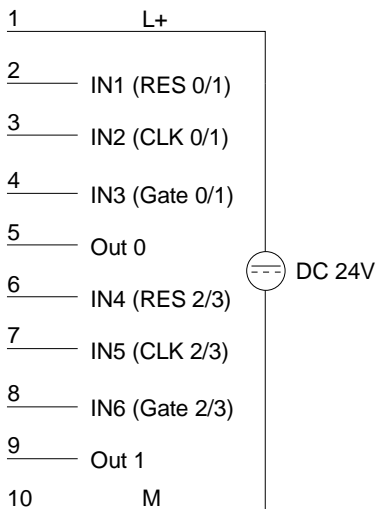


**Mode 26**  
**One Shot, Zähl-  
 richtung abwärts**  
**mit Rücksetz-  
 signal**

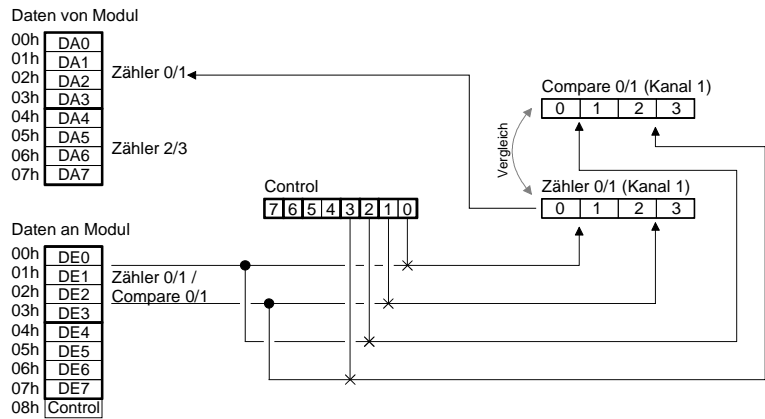
Unter dem Mode 26 können Sie je Kanal einen 32 Bit Zähler realisieren, der über ein Torsignal (Gate) gesteuert wird. Mit jeder steigenden Flanke des Eingangstakts wird der Zähler um 1 abwärts gezählt, sofern Gate HIGH-Pegel hat. RES muss während des Zählvorgangs auf LOW liegen. Ein HIGH-Pegel löscht den Zähler. Das Signal OUT (aktiv mit 0) wird mit dem Laden des Zählers LOW; bei Erreichen des in COMPARE geladenen Wertes wird OUT wieder HIGH.

**Mode 26 - One Shot, down, Reset**

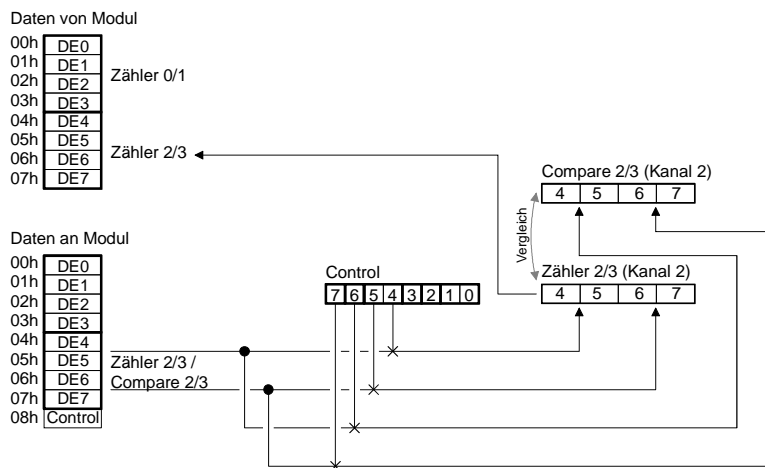
**Anschlussbelegung**  
**Zählerzugriff**



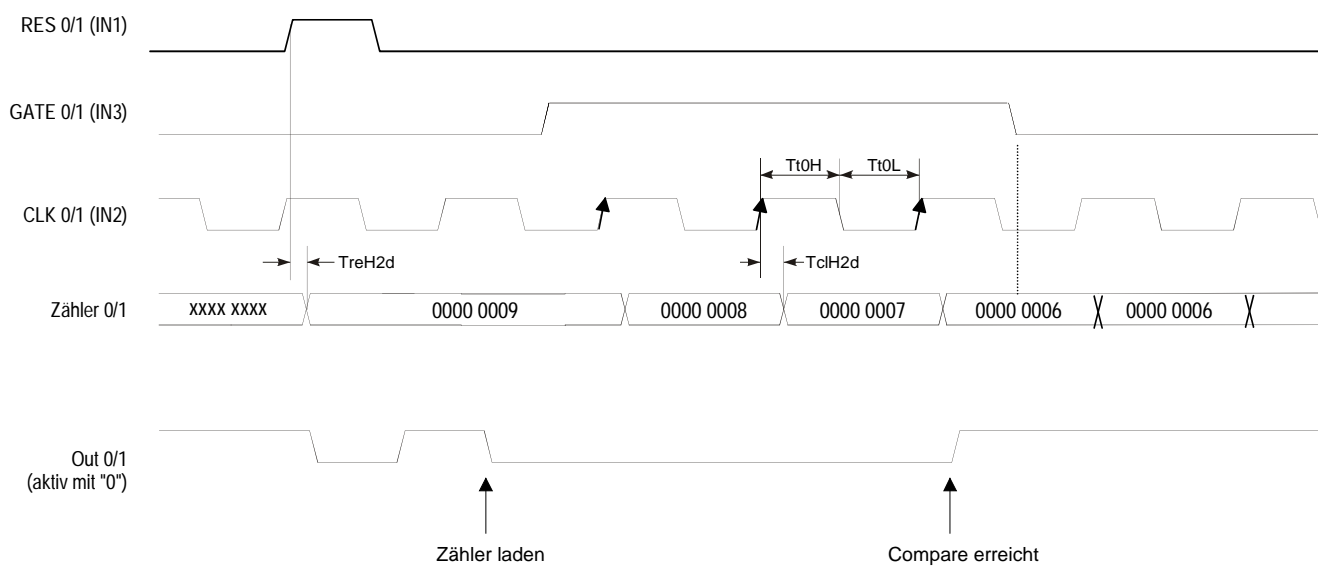
**Zähler 0/1 (Kanal 1)**



**Zähler 2/3 (Kanal 2)**



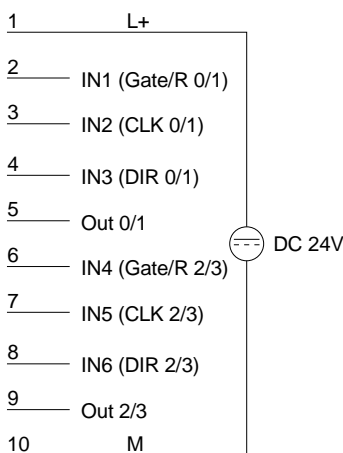
Timing-Diagramm Beispiel von Zähler 0/1 im Mode 26:



**Mode 27**  
**32Bit Zähler**

Unter Mode 27 werden je 2 Zähler (16 Bit) zu einem 32 Bit Zähler zusammengefasst. Die Zählrichtung bestimmen Sie über DIR (IN3 bzw. IN6). Mit jeder steigenden Flanke des Eingangstakts wird der Zähler um 1 inkrementiert bzw. dekrementiert. Mit steigender Flanke des Signals Gate/R wird der Zähler gelöscht. Während des Zählvorgangs muss das Signal Gate/R auf HIGH liegen. Wird das Signal Gate/R "0" bleibt der Zählerstand erhalten. Bei Erreichen von Null wird der Ausgang OUT des entsprechenden Zählers mindestens für 100ms gesetzt, auch wenn der Zähler weiterzählt. Sollte der Zähler bei Null stoppen, bleibt der Ausgang gesetzt.

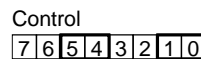
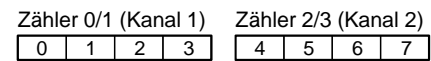
**Anschlussbelegung**  
**Zählerzugriff**



Daten von Modul

00h	DA0
01h	DA1
02h	DA2
03h	DA3
04h	DA4
05h	DA5
06h	DA6
07h	DA7

Zähler 0/1  
Zähler 2/3



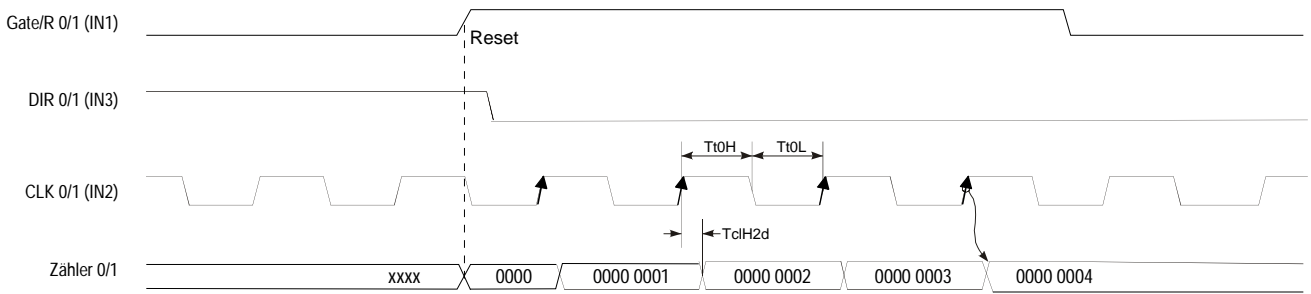
Daten an Modul

00h	DE0
01h	DE1
02h	DE2
03h	DE3
04h	DE4
05h	DE5
06h	DE6
07h	DE7
08h	Control

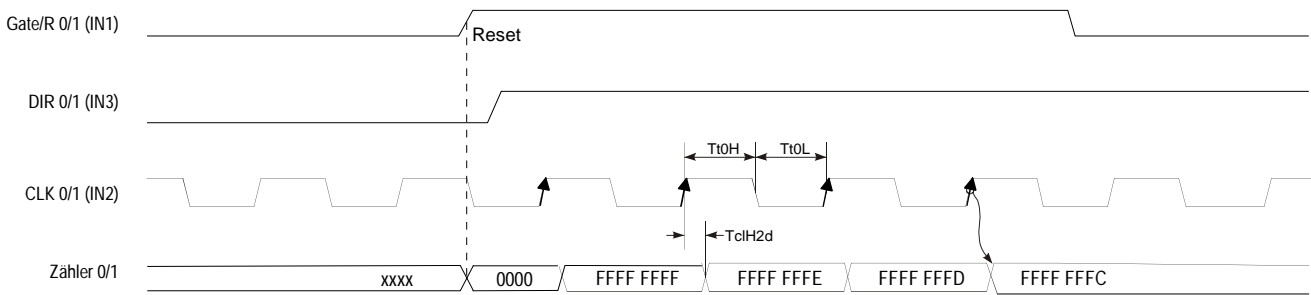
Zähler 0/1  
Zähler 2/3



**Aufwärtszähler** Durch einen LOW-Pegel am Eingang DIR im Mode 27 wird der Zähler als Aufwärtszähler eingestellt.  
 Timing-Diagramm am Beispiel von Zähler 0/1:



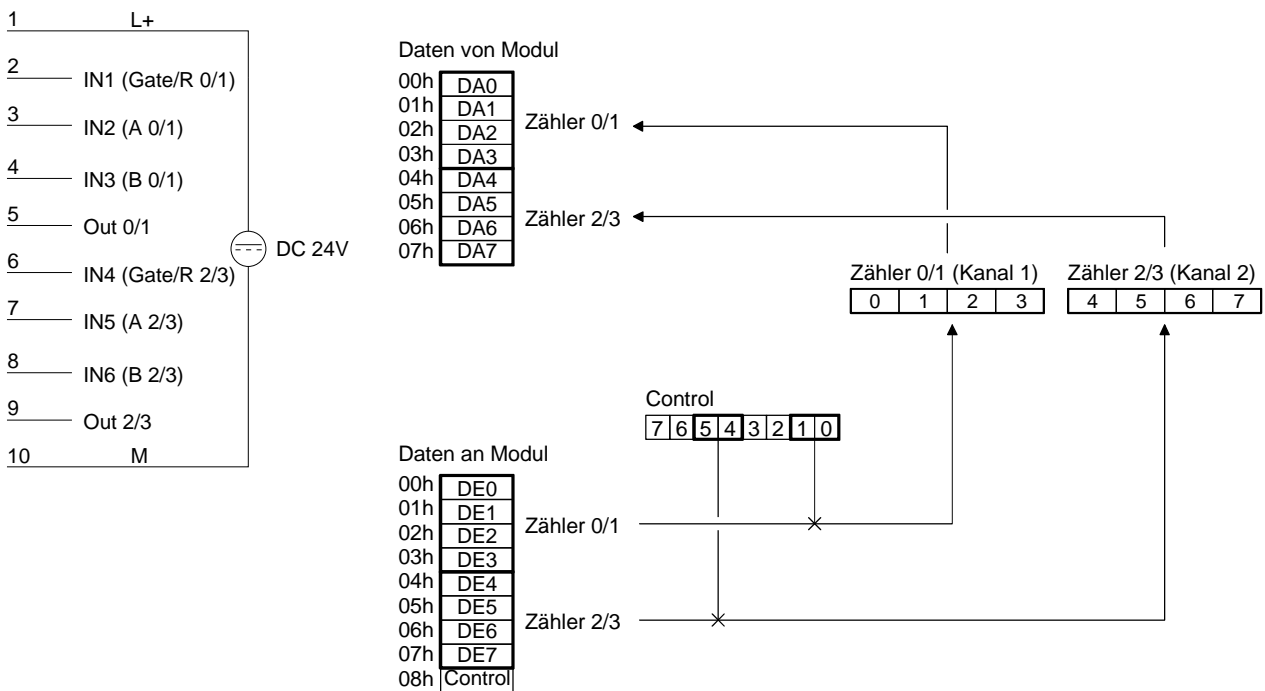
**Abwärtszähler** Durch einen HIGH-Pegel am Eingang DIR im Mode 27 wird der Zähler als Abwärtszähler eingestellt.  
 Timing-Diagramm am Beispiel von Zähler 0/1:



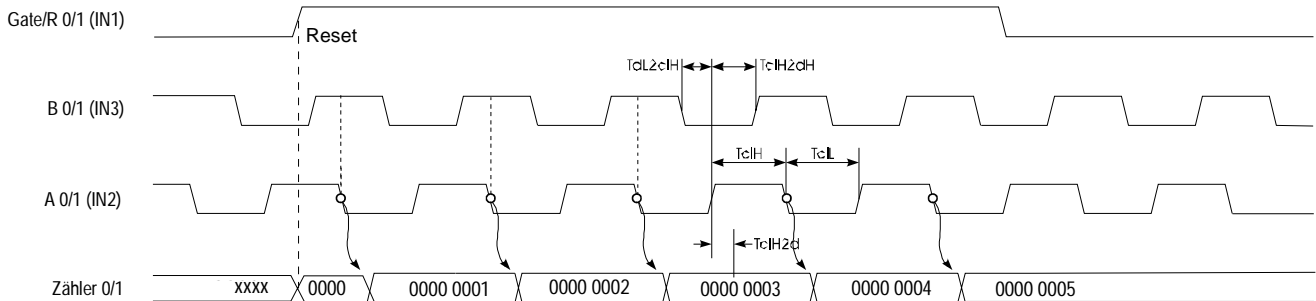
**Mode 28  
Encoder 1 Flanke**

Im Mode 28 können Sie für einen Kanal einen Encoder einstellen, der bei jeder fallenden Flanke entsprechend der Drehrichtung den internen Zähler um 1 inkrementiert bzw. dekrementiert. Mit steigender Flanke des Signals Gate/R wird der Zähler gelöscht. Während des Zählvorgangs muss das Signal Gate/R auf HIGH liegen. Wird das Signal Gate/R "0" bleibt der Zählerstand erhalten. Bei Erreichen von Null wird der Ausgang OUT des entsprechenden Zählers mindestens für 100ms gesetzt auch, wenn der Zähler weiterzählt. Sollte der Zähler bei Null stoppen, bleibt der Ausgang gesetzt.

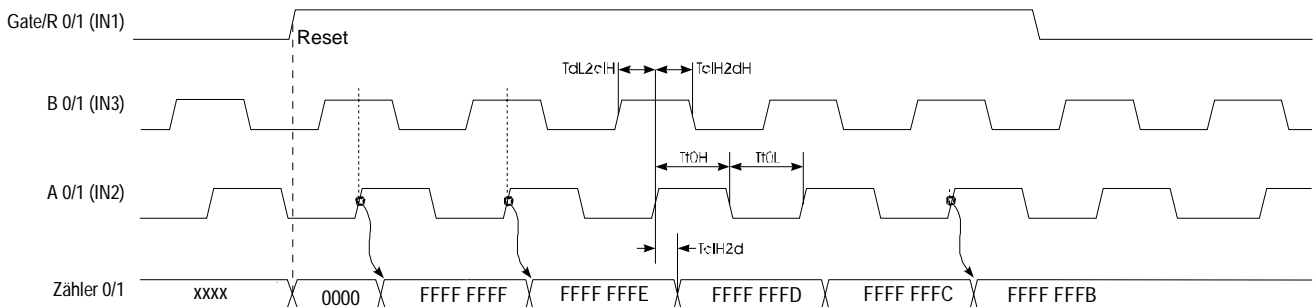
**Anschlussbelegung  
Zählerzugriff**



**Aufwärtszähler** Jede fallende Flanke an Eingang A inkrementiert den Zähler um 1, wenn zu diesem Zeitpunkt der Eingang B auf HIGH liegt.  
 Timing-Diagramm am Beispiel von Zähler 0/1:



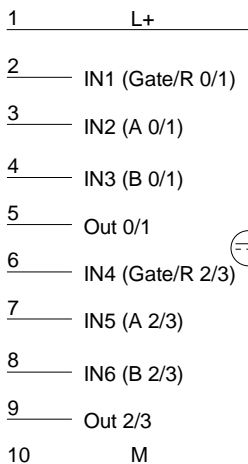
**Abwärtszähler** Jede steigende Flanke am Eingang A dekrementiert den internen Zähler um 1, wenn zu diesem Zeitpunkt der Eingang B auf HIGH liegt.  
 Timing-Diagramm am Beispiel von Zähler 0/1:



**Mode 29  
Encoder 2 Flanken**

Jede steigende bzw. fallende Flanke des Signals an Eingang A verändert den Zählerstand um 1. Die Zählrichtung ist vom aktuellen Pegelstand des Eingangs B abhängig. Mit steigender Flanke des Signals Gate/R wird der Zähler gelöscht. Während des Zählvorgangs muss das Signal Gate/R auf HIGH liegen. Wird das Signal Gate/R "0" bleibt der Zählerstand erhalten. Bei Erreichen von Null wird der Ausgang OUT des entsprechenden Zählers mindestens für 100ms gesetzt auch wenn der Zähler weiterzählt. Sollte der Zähler bei Null stoppen, bleibt der Ausgang gesetzt.

**Anschlussbelegung  
Zählerzugriff**

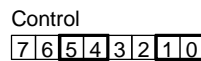
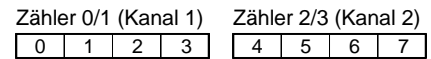


Daten von Modul

00h	DA0
01h	DA1
02h	DA2
03h	DA3
04h	DA4
05h	DA5
06h	DA6
07h	DA7

Zähler 0/1

Zähler 2/3



Daten an Modul

00h	DE0
01h	DE1
02h	DE2
03h	DE3
04h	DE4
05h	DE5
06h	DE6
07h	DE7
08h	Control

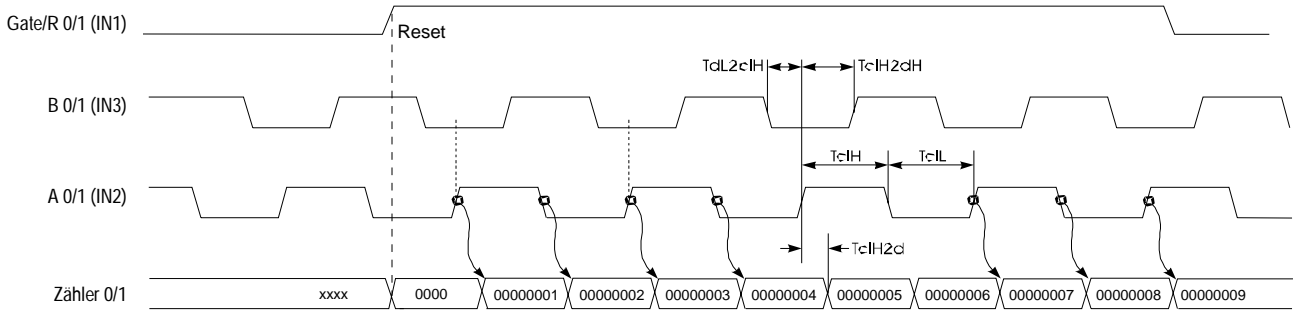
Zähler 0/1

Zähler 2/3

**Aufwärtszähler**

Der Zähler wird um 1 inkrementiert, wenn bei steigender Flanke des Signals A der Eingang B auf LOW liegt, oder bei fallender Flanke von A der Eingang B auf HIGH liegt.

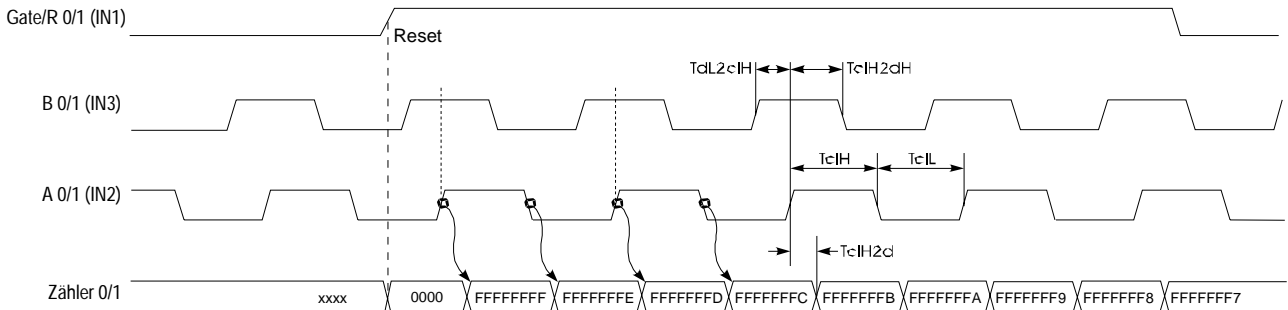
Timing-Diagramm am Beispiel von Zähler 0/1:



**Abwärtszähler**

Der Zähler wird um 1 dekrementiert, wenn bei steigender Flanke des Signals A der Eingang B auf HIGH liegt, oder bei fallender Flanke von A der Eingang B auf LOW liegt.

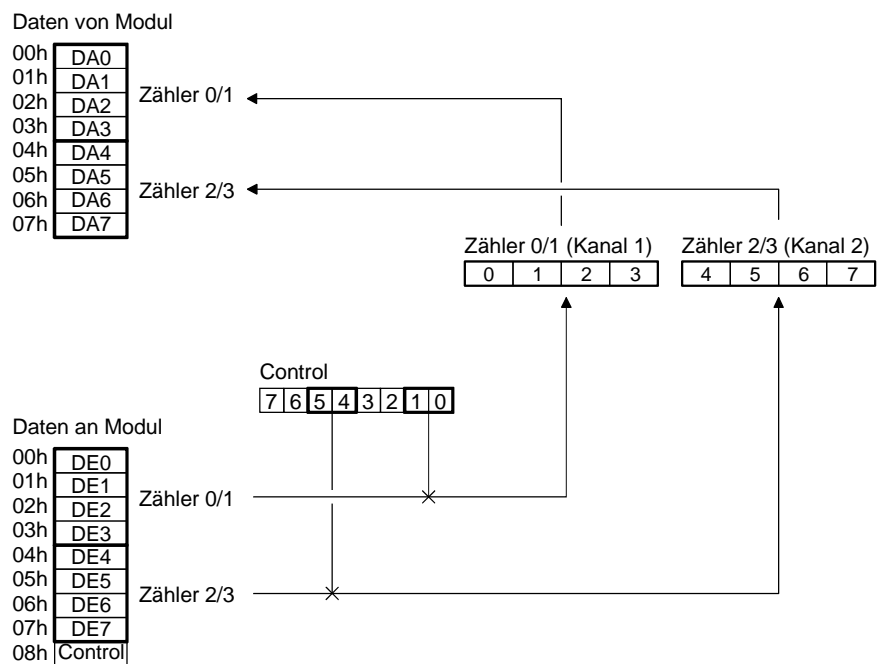
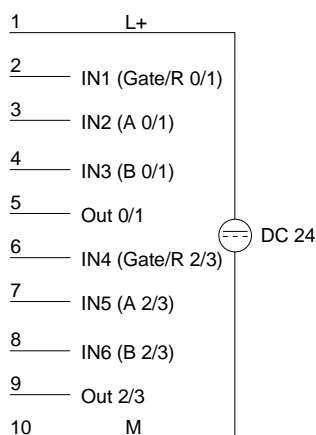
Timing-Diagramm am Beispiel von Zähler 0/1:



**Mode 30  
Encoder 4 Flanken**

Jede steigende bzw. fallende Flanke an einem der Eingänge A oder B verändert den Zählerstand um 1, wobei die Zählrichtung vom Pegel des anderen Eingangs (B bzw. A) abhängt. Mit steigender Flanke des Signals Gate/R wird der Zähler gelöscht. Während des Zählvorgangs muss das Signal Gate/R auf HIGH liegen. Wird das Signal Gate/R "0" bleibt der Zählerstand erhalten. Bei Erreichen von Null wird der Ausgang OUT des entsprechenden Zählers mindestens für 100ms gesetzt, auch wenn der Zähler weiterzählt. Sollte der Zähler bei Null stoppen, bleibt der Ausgang gesetzt.

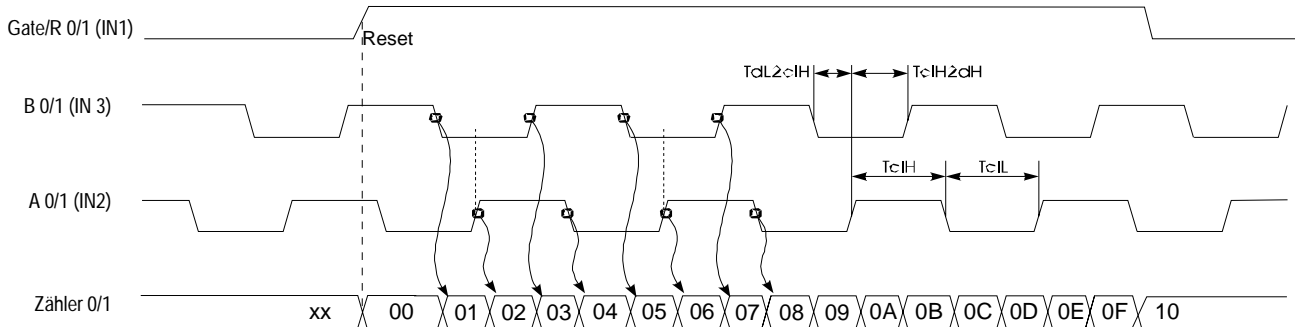
**Anschlussbelegung  
Zählerzugriff**



**Aufwärtszähler**

Der Zähler wird dann um 1 inkrementiert, wenn bei steigender Flanke an B der Eingang A auf HIGH liegt, bzw. wenn bei fallender Flanke an B der Eingang A auf LOW liegt, oder wenn bei steigender Flanke an A der Eingang B auf LOW liegt, bzw. bei fallender Flanke an A der Eingang B auf HIGH liegt.

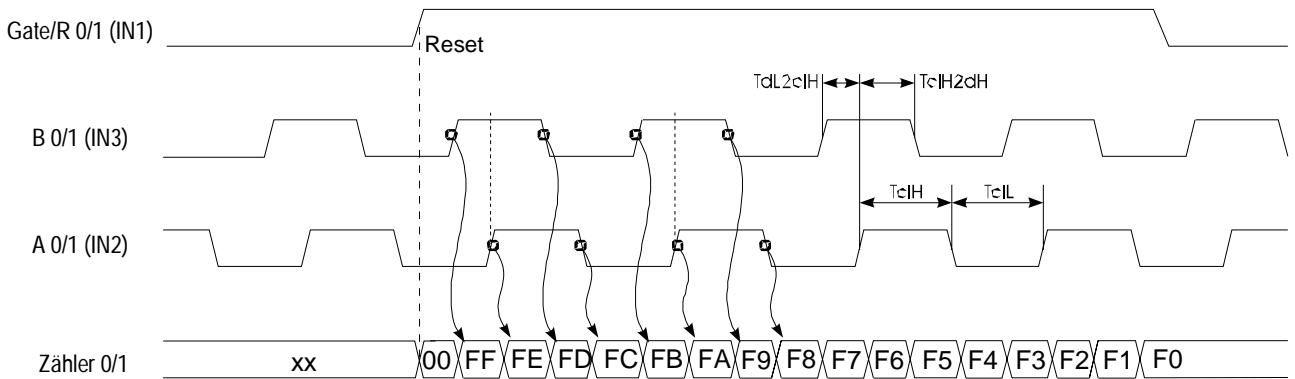
Timing-Diagramm am Beispiel von Zähler 0/1:



**Abwärtszähler**

Der Zähler wird dann um 1 dekrementiert, wenn bei steigender Flanke an B der Eingang A auf LOW liegt, bzw. wenn bei fallender Flanke an B der Eingang A auf HIGH liegt, oder wenn bei steigender Flanke an A der Eingang B auf HIGH liegt, bzw. bei fallender Flanke an A der Eingang B auf LOW liegt.

Timing-Diagramm am Beispiel von Zähler 0/1:



## Technische Daten

### SSI-Modul FM 250S

Elektrische Daten	VIPA 250-1BS00
Anzahl der Kanäle	1
Anzahl der Ausgänge	2
Stromaufnahme	200mA über Rückwandbus
Potenzialtrennung	ja
SSI-Schnittstelle	Geberversorgung
Datenleitung	RS422, potenzialgetrennt
Clock	RS422, potenzialgetrennt
Baudrate	parametrierbar: 100 / 300 / 600kBaud (Default: 300kBaud)
Signalspannung "0"	-5 ... 7V
Signalspannung "1"	13 ... 36V
Ausgangsstufe	DC 24V high side switch 0,5A
Ext. Spannungsversorgung	DC 24V (18 ... 28,8V)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	4Byte
Ausgabedaten	4Byte, 8Byte-Puffer im Modul
Parameterdaten	4Byte
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT)	25,4x76x76mm
Gewicht	100g



**Zähler-Modul****FM 250**

Elektrische Daten	VIPA 250-1BA00
Anzahl der Zähler	2 bzw. 4
Zählerbreite	32Bit bzw. 16Bit
Anzahl der Betriebsarten	26
Zählfrequenz	max. 1MHz
Stromaufnahme	80mA über Rückwandbus
Potenzialtrennung	ja
Ausgangsstufe	DC 24V high side switch 0,5A
Ext. Spannungsversorgung	DC 24V (18 ... 28,8V)
Signalspannung "0"	-30 ... 5V
Signalspannung "1"	13 ... 30V
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	10Byte
Ausgabedaten	10Byte
Parameterdaten	2Byte
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT)	25,4x76x76mm
Gewicht	100g



## Teil 11 MotionControl-Module

### Überblick

In diesem Kapitel finden Sie Informationen über den Anschluss, die Datenübertragung und die Betriebsarten der MotionControl-Module für Schritt- und Servo-Antriebe.

Nachfolgend sind beschrieben:

- Anschluss
- Parametrierung
- Datenübergabe
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 11 MotionControl-Module</b> .....	<b>11-1</b>
Systemübersicht .....	11-2
FM 253 - MotionControl Stepper .....	11-3
FM 253 - MotionControl Stepper - Aufbau .....	11-4
FM 253 - Anschluss eines Antriebs.....	11-6
FM 253 - Datenübergabe >> FM 253.....	11-8
FM 253 - Parametrierung.....	11-9
FM 253 - Betriebsarten .....	11-11
FM 253 - Datenübergabe > CPU .....	11-15
FM 253 - Hantierungsbausteine.....	11-17
FM 254 - MotionControl Servo .....	11-23
FM 254 - MotionControl Servo - Aufbau.....	11-24
FM 254 - Anschluss eines Antriebs mit Drehgeber .....	11-26
FM 254 - Übersicht der Parameter und Übergabewerte.....	11-28
FM 254 - Parametrierung.....	11-29
FM 254 - Datenübergabe >> FM 254.....	11-30
FM 254 - Betriebsarten .....	11-31
FM 254 - Datenübergabe >> CPU .....	11-37
Technische Daten.....	11-38

# Systemübersicht

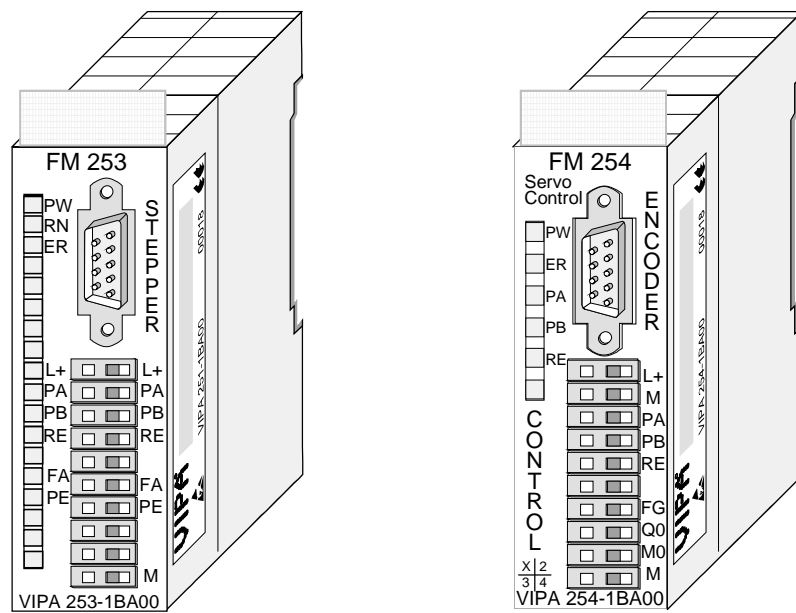
## Allgemeines

Die hier beschriebenen MotionControl-Module sind Module für Antriebe in Maschinen mit hohen Taktraten.

Die Module sind einsetzbar für Punkt-zu-Punkt-Positionierungen und für komplexe Verfahrprofile mit höchsten Ansprüchen an Genauigkeit, Dynamik und Geschwindigkeit.

Je nach Modul können Schrittmotoren oder Servoantriebe angesteuert werden.

## MotionControl-Module



## Bestelldaten

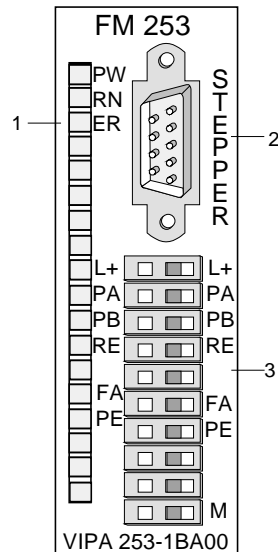
Typ	Bestellnummer	Beschreibung	Seite
FM 253	VIPA 253-1BA00	MotionControl-Modul Stepper	11-3
FM 254	VIPA 254-1BA00	MotionControl-Modul Servo	11-23

## FM 253 - MotionControl Stepper

- Eigenschaften** Das FM 253 ist ein Positionier-Modul für die Ansteuerung eines Schrittmotors.  
Das Modul arbeitet selbständig und wird über ein entsprechendes Anwenderprogramm in der CPU gesteuert.  
Das Modul hat folgende Eigenschaften:
- Mikroprozessorgesteuertes Positionier-Modul zur Ansteuerung eines 1-achsigen Antriebs mit Schrittmotor
  - Betrieb von Rund- und Linearachsen
  - Verschiedene Betriebsarten
  - Das Modul arbeitet selbständig und wird über ein Anwenderprogramm im System 200V gesteuert.
  - Die Parametrierdaten werden im internen Flash-Speicher abgelegt. Es ist keine Batterie erforderlich.
  - Das Modul besitzt zum Anschluss von Endschaltern 3 Eingänge und kann 2 Ausgänge ansteuern. Der Zustand der Ein-/Ausgänge wird zusätzlich auf LEDs ausgegeben.
- Einsatzgebiete** Das Modul können Sie sowohl für einfache Positionieraufgaben einsetzen als auch für komplexe Verfahraphile mit höchstem Anspruch an die Genauigkeit bei der Zielpositionierung.  
Schrittmotoren kommen zum Einsatz, wenn maximales Drehmoment bei niedrigen Drehzahlen gefordert ist und die Zielposition ohne Überschwingen erreicht und gehalten werden soll.
- Betriebsarten** Die Betriebsart geben Sie durch Ihr Anwenderprogramm vor. Das Modul verfügt über folgende Betriebsarten:
- Positionierbetrieb absolut
  - Positionierbetrieb relativ
  - Referenzfahrt
  - Dauerlauf Achse
  - Position setzen
  - Motorparameter bearbeiten
  - Fehler löschen
  - Eingänge lesen

## FM 253 - MotionControl Stepper - Aufbau

### Frontansicht



- [1] LED Statusanzeigen
- [2] Anschluss für Antrieb
- [3] Anschluss für Versorgungsspannung, Endschalter und Ausgänge

### Komponenten

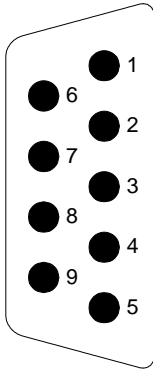
#### LEDs

Das FM 253 besitzt an der Front LEDs, die der Statusanzeige dienen. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
PW	Gelb	Versorgungsspannung über Rückwandbus liegt an
RN	Grün	RUN: Regelung aktiv
ER	Rot	interner Fehler
L+	Gelb	DC 24V Versorgungsspannung für die Ausgänge liegt an.
PA	Grün	Grenzwert A überfahren, Eingang PA gesetzt
PB	Grün	Grenzwert B überfahren, Eingang PB gesetzt
RE	Grün	Referenzpunkt überfahren
FA	Grün	Antrieb in Fahrt
PE	Grün	Antrieb hat Position erreicht

**"Stepper"-  
Schnittstelle**

Über diese Schnittstelle schließen Sie Ihren Schrittmotor an. Die Schnittstelle ist als 9-poliger SubD Stecker ausgeführt und arbeitet mit RS422-Pegel. Sie hat folgende Pinbelegung:

*9-poliger SubD Stecker*

Pin	Belegung
1	PULSE_P: (+) Taktausgang
2	DIR_P: (+) Richtungssignal
3	ENABLE_P: (+) Freigabesignal
4	READY+: (+) Bereitschaftsmeldung
5	GND: Masse
6	PULSE_N: (-) Taktausgang
7	DIR_N: (-) Richtungssignal
8	ENABLE_N: (-) Freigabesignal
9	READY-: (-) Bereitschaftsmeldung

**Control-  
Schnittstelle**

Die "Control"-Schnittstelle bietet Anschlussmöglichkeiten für Endschalter und Ausgabeelemente. Die Schnittstelle hat folgende Pinbelegung:



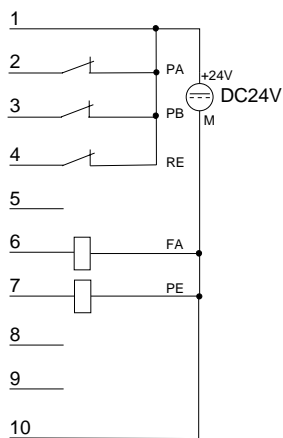
Pin	Belegung
1	Versorgungsspannung DC 24V für Ausgänge
2	Eingang: Endschalter PA
3	Eingang: Endschalter PB
4	Eingang: Referenzschalter
5	reserviert
6	Ausgang: Achse in Fahrt
7	Ausgang: Position erreicht
8	reserviert
9	reserviert
10	Masse 24V

## FM 253 - Anschluss eines Antriebs

### Anschluss eines Schrittmotors

Der Anschluss eines Schrittmotors erfolgt ausschließlich über die "Stepper"-Schnittstelle

### Anschluss von Versorgungsspannung, Endschalter und Ausgabeeinheiten



#### Spannungsversorgung

Das Modul selbst wird über den Rückwandbus versorgt. Der Einsatz der integrierten digitalen Ausgänge macht eine zusätzliche Spannungsversorgung erforderlich. Der Anschluss einer zusätzlichen DC 24V Versorgungsspannung erfolgt über Klemme 1 und 10 der "Control"-Schnittstelle.

#### Eingänge für Endschalter

An das Modul können Sie bis zu 3 Endschalter (Öffner) anschließen.

An Klemme 2 und 3 (PA und PB) schließen Sie die Endschalter an, mit denen Sie die Strecke begrenzen. Sobald einer dieser Schalter betätigt wird, wird der Antrieb sofort angehalten und kann dann nur noch in die Gegenrichtung gefahren werden.

An Klemme 4 schließen Sie den Referenzschalter an, der für die Abstimmung Ihres Antriebs auf das FM 253 Modul erforderlich ist.

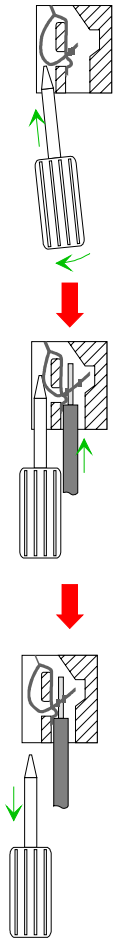
#### Ausgänge

Das Modul besitzt 2 Ausgänge, die nur vom Modul angesteuert werden können:

- FA - Antrieb in Fahrt (Klemme 6)
- PE - Antrieb hat Position erreicht (Klemme 7)

Die Zustände der Ausgänge werden über die entsprechenden LEDs ausgegeben.



**Verdrahtung**

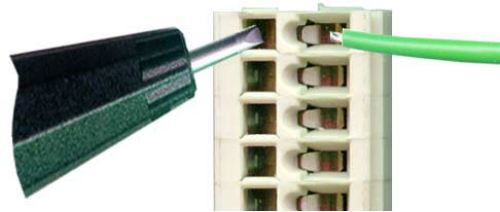
Die Endschalter und die Ausgänge sind an "Control" anzuschließen. Hierbei wird ein 10-poliger Stecker mit Federklemmtechnik der Firma WAGO eingesetzt. Die Verdrahtung mit Federklemmtechnik ermöglicht eine schnelle Verdrahtung. Im Gegensatz zu Schraubverbindungen ist diese Technik erschütterungssicher.

Sie können Drähte mit dem Querschnitt von  $0,08\text{mm}^2$  bis  $1,5\text{mm}^2$  anschließen.

Die Verdrahtung geschieht auf die gleiche Weise wie bei den großen Federklemmen des System 200V.

Drücken Sie mit einem passenden Schraubendreher die Federklemme in der quadratische Öffnung nach hinten und führen Sie durch die rechteckige Öffnung den Draht ein.

Durch Herausziehen des Schraubendrehers wird der Draht sicher gehalten.



## FM 253 - Datenübergabe >> FM 253

### Fahr-Daten

Das MotionControl Stepper Modul holt sich zyklisch einen Datenblock von der CPU und wertet diesen aus.

Der Datenblock ist 16Byte lang und hat folgenden Aufbau:

Byte-Nr.	Inhalt	Länge
0-3	Soll-Position	4Byte
4-7	Soll-Frequenz	4Byte
8-9	Reserviert	
10	Modus	1Byte
11	Index	1Byte
12-15	Variablen Parameter	4Byte

Über das "Modus"-Byte werden die Inhalte des Datenblocks spezifiziert. Folgende Funktionen können über das "Modus"-Byte ausgelöst werden:

### Modus (Byte 10)

Bit 7 ... Bit 0	Vorgabe in Byte	Rückgabe in Byte
00: Idle-Mode - es erfolgt keine Zustandsänderung des Antriebs, dient zur Parameteränderung	-	-
01: Positionieren relativ - Anfahren der vorgegebenen Schrittzahl	0-3: rel. Sollposition	-
02: Referenzfahrt - Kalibrierung des Antriebs	15: Parameter-Bits	-
03: Dauerlauf Achse - Antrieb dreht mit Sollfrequenz	4-7: Sollfrequenz	-
04: Eingänge lesen - gibt Zustände der Endschalter zurück	-	15: Zustand
05: Motor Parameter - übergibt je nach Index Parameter	11: Index, 12-15: Parameter	-
06: Position setzen - setzt in Modul die Ist-Position ohne Antrieb zu bewegen	0-3: Sollposition	-
07: Fehler löschen - löscht das mit 1 aktivierte Fehlerbit	14-15: Fehlerbit	-
08: Positionieren absolut - Anfahren der Sollposition	0-3: abs. Sollposition	-

### Parameterübergabe (Modus = 05h)

Über **Index (Byte 11)** stellen Sie den Parameter ein, dessen Wert Sie über **Byte 12-15** vorgeben können. Der Wert wird an das Modul übertragen, indem Sie den **Modus 05h in Byte 10** vorgeben.

Näheres hierzu auf der Folgeseite.

## FM 253 - Parametrierung

### Übersicht

Die Parameterdaten werden zusammen mit den Fahrdaten in dem 16Byte großen Datenblock an das Modul übergeben. Zur Parametrierung geben Sie im **"Index"-Byte (Byte 11)** über die **Index-Nr.** den Parameter an, dessen Wert Sie ändern möchten.

Den neuen Wert tragen Sie in **Byte 12-15** ein.

Sobald Sie das **"Modus"-Byte (Byte 10)** mit **05h** belegen, wird der Parameter an das Modul übertragen.



Bitte beachten Sie, dass neue Parameter nur übernommen werden, wenn zuvor ein Modus-Wechsel stattgefunden hat. Wechseln Sie hierzu nach jeder Parameterübergabe in den Idle-Modus ("Modus"-Byte 10 = 00h).

### Parameter in Flash ablegen

Die Parameter, die Sie an das Modul übertragen, werden im RAM abgelegt. Solange das Modul mit Spannung versorgt wird, bleiben die Parameter erhalten. Über die Index-Nr. 97h haben Sie aber auch die Möglichkeit, die Parameter im internen Flash abzulegen.

Nach PowerOn stehen die Parameter dann wieder zur Verfügung.

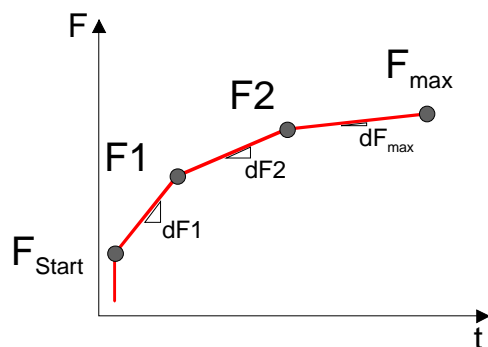
### Parametrierung über FCs

Von VIPA erhalten Sie FCs, die den Einsatz des FM 253 vereinfachen sollen. Beispielsweise können Sie mit den FCs 201 und 202 Ihr Modul parametrieren.

Die Ansteuerung des Antriebs erfolgt mit dem FC 200. Über diesen Baustein stehen Ihnen mit Ausnahme von "Parameter setzen" alle Modi zur Verfügung.

### Zusammenhang der Parameter

In der nachfolgenden Abbildung sind die wichtigsten Zusammenhänge der Parameter aufgeführt. Die Zuordnung der entsprechenden Index-Nr. finden Sie in der Tabelle weiter unten.



**Index bei Parameter setzen** Über die Index-Nr. stellen Sie in Byte 11 den Parameter ein, dessen Wert Sie in Byte 12-15 vorgeben können.

Index	Parameter	Einheit	Wertebereich	Default	Bemerkung
00h	Fstart	Hz	UINT32	200	Startfrequenz
01h	F1	Hz	UINT32	4000	Grenzfrequenz 1
02h	dF1	Hz	UINT32	100	Beschleunigung von Fstart ⇒ F1
03h	F2	Hz	UINT32	10000	Grenzfrequenz 2
04h	dF2	Hz	UINT32	60	Beschleunigung von F1 ⇒ F2
05h	Fmax	Hz	UINT32	30000	Maximale Fahrfrequenz
06h	dFmax	Hz	UINT32	40	Beschleunigung von F2 ⇒ Fmax
07h	Fpos	Hz	UINT32	30000	Frequenz beim Positionieren
08h	Fref	Hz	UINT32	1000	Frequenz für die Referenzfahrt
10h	Fist	Hz	UINT32	-	Aktuelle Motorfrequenz (nur lesen)
11h	Fsoll	Hz	UINT32	-	Aktuelle Sollfrequenz (nur lesen)
13h	FTarget	Hz	UINT32	-	Zielfrequenz (nur lesen)
97h				-	Parameter in Flash ablegen
98h				-	Parameter aus Flash lesen (Zustand wie nach PowerON)
99h				-	Default-Parameter laden



### Hinweis!

Bei der Vergabe der Parameter für den Antrieb sollten folgende Regeln beachtet werden:

- **dF1** sollte immer **kleiner** als **Fstart** sein
- **dF2** sollte die **Hälfte** von **dF1** betragen
- **dFmax** sollte die **Hälfte** von **dF2** betragen

Hieraus ergibt sich folgender Zusammenhang:

$$4 \cdot dF_{\max} = 2 \cdot dF2 = dF1 < F_{\text{Start}}$$

Fehleingaben werden in gewissen Grenzen von der Firmware des Moduls angepasst.

## FM 253 - Betriebsarten

### Übersicht

Durch das Setzen entsprechender Bits im "Modus"-Byte können Sie folgende Betriebsarten einstellen, die weiter unten näher beschrieben sind:

- Idle-Modus
- Positionieren relativ / absolut
- Dauerlauf
- Position setzen
- Referenzfahrt

### Idle-Modus

Vorgaben: Byte 10 = 00h

Im Idle-Modus erfolgt keine Zustandsänderung des Antriebs. Da nur nach einem Modus-Wechsel Daten neu in das Modul übernommen werden, können Sie durch Sprung in den Idle-Modus und wieder zurück einen Modus-Wechsel initiieren.

Über den Idle-Modus können Sie z.B. einen neuen Auftrag starten, da durch Sprung in den Idle-Modus ein Moduswechsel erkannt wird.

Die Betriebsart "Idle" sollte immer aufgerufen werden, solange keine Aktion ausgelöst werden soll. Im Normalfall wird zum Start einer Aktion nur kurzzeitig in einen anderen Modus verzweigt und anschließend wieder in den Idle-Modus zurückgekehrt.

### Positionieren relativ

Vorgaben: Byte 10 = 01h, Byte 0-3 = relative Sollposition

Beim Positionieren relativ wird eine vorgegebene Schrittzahl auf die aktuelle Position addiert und anschließend angefahren.

Hierbei ist der Positions-Offset (Schrittzahl) als relative Sollposition in Byte 0-3 vorzugeben und anschließend der **Modus (Byte10) auf 01h** zu setzen.

Durch Setzen von Byte 10 auf 01h startet die relative Positionierung.

Für Beschleunigung und Frequenz des Antriebs werden die in den Parametern angegebenen Werte verwendet. Liegen keine Parameter vor, kommen die Default-Parameter zum Einsatz.

Solange der Antrieb in Betrieb ist, wird der Ausgang "**Achse in Fahrt**" gesetzt. Nach Erreichen der Position wird dieser Ausgang gelöscht und der Ausgang "**Position erreicht**" gesetzt.

### Positionieren absolut

Vorgaben: Byte 10 = 08h, Byte 0-3 = absolute Sollposition

Beim absoluten Positionieren wird eine absolute Sollposition angefahren. Hierbei ist die Position (Schrittzahl) als absolute Sollposition in Byte 0-3 vorzugeben und anschließend der **Modus (Byte 10) auf 08h** zu setzen.

Durch Setzen von Byte 10 auf 08h startet die absolute Positionierung.

Fortsetzung ...

**Positionieren  
absolut  
Fortsetzung ...**

Für Beschleunigung und Frequenz des Antriebs werden die in den Parametern angegebenen Werte verwendet. Liegen keine Parameter vor, kommen die Default-Parameter zum Einsatz.

Solange der Antrieb in Betrieb ist, wird der Ausgang "**Achse in Fahrt**" gesetzt. Nach Erreichen der Position wird dieser Ausgang gelöscht und der Ausgang "**Position erreicht**" gesetzt.

**Dauerlauf**

Vorgaben: Byte 10 = 03h, Byte 4-7 = Sollfrequenz

Beim Dauerlauf dreht die Achse solange mit der eingestellten Sollfrequenz bis diese geändert wird.

Hierbei ist die Drehzahl als Sollfrequenz in Byte 4-7 vorzugeben und anschließend der **Modus (Byte10)** auf **03h** zu setzen.

Durch Setzen von Byte 10 auf 03h startet der Antrieb und dreht sich mit der vorgegebenen Frequenz, bis ein neuer Frequenzwert vorgegeben wird.

Eine neue Frequenz wird nur bei Modus-Wechsel übernommen. Dies erreichen Sie, indem Sie nach dem Anlauf des Antriebs in den Idle-Modus (Byte 10 = 00h) wechseln. Geben Sie nun die neue Sollfrequenz an und setzen Sie das Byte 10 wieder auf 03h. Sofort stellt der Antrieb sich auf die neue Frequenz ein.

Für Beschleunigung des Antriebs werden die in den Parametern angegebenen Werte verwendet. Liegen keine Parameter vor, kommen die Default-Parameter zum Einsatz.

Solange der Antrieb in Betrieb ist, wird der Ausgang "**Achse in Fahrt**" gesetzt. Durch Vorgaben von 00h als Sollfrequenz (Moduswechsel erforderlich) stoppt der Antrieb und der Ausgang "Achse in Fahrt" wird wieder zurückgesetzt.

**Antrieb stoppen mit  
"Dauerlauf" und  
Sollfrequenz = 00h**

Durch Angabe einer Sollfrequenz von 00h in Byte 4-7 und dem Modus 03h in Byte 10 können Sie jederzeit den Antrieb stoppen.



**Hinweis!**

Bitte beachten Sie, dass eine Frequenzänderung nur durch einen Modus-Wechsel vom Modul erkannt wird. Dies gilt auch zum Stoppen des Antriebs. Verwenden Sie zum Modus-Wechsel den kurzzeitigen Sprung in den Idle-Modus.

**Position setzen**

Vorgaben: Byte 10 = 06h, Byte 0-3: Positionswert

In der Betriebsart "Position setzen" können Sie den aktuellen Ist-Wert mit einem Wert belegen.

Geben Sie hierzu den neuen Wert in Byte 0-3 vor und setzen Sie anschließend das "Modus"-Byte 10 auf 06h.

## Referenzfahrt

Vorgaben: Byte 10 = 02h, Byte 15 = Steuerbits für Referenzfahrt

Die Referenzfahrt dient der Kalibrierung Ihres Antriebsystems. Der Referenzpunkt sollte im Verfahrensweg liegen.

Vor einer Referenzfahrt ist in Byte 15 der Typ der Referenzfahrt und die Fahrtrichtung zu spezifizieren.

Durch Setzen von Byte 10 auf 02h startet der Antrieb mit seiner Referenzfahrt.

Als Frequenz wird die in den Parametern angegebene Referenzfrequenz verwendet. Liegt kein Parameter vor, kommt der Default-Parameter zum Einsatz.

Für die Referenzfahrt gibt es 6 verschiedene Möglichkeiten, die über Byte 15 vorzugeben sind:

- Referenzfahrt auf Referenzschalter und Positionszähler löschen
- Referenzfahrt auf Referenzschalter und Positionszähler beibehalten
- Referenzfahrt auf Endschalter B und Positionszähler löschen
- Referenzfahrt auf Endschalter B und Positionszähler beibehalten
- Referenzfahrt auf Endschalter A und Positionszähler löschen
- Referenzfahrt auf Endschalter A und Positionszähler beibehalten

## Steuerbits für die Referenzfahrt

Die Steuerbits in Byte 15 haben folgende Belegung:

Byte 15	Parameter
Bit 0	1: Richtung vorwärts 0: Richtung rückwärts
Bit 1	1: nach Referenzfahrt Position löschen 0: nach Referenzfahrt Position nicht löschen
Bit 2	Referenzfahrt auf Referenzschalter
Bit 3	Referenzfahrt auf Endschalter B
Bit 4	Referenzfahrt auf Endschalter A



### Hinweis!

Beim Start der Referenzfahrt ist darauf zu achten, dass über Bit 0 immer eine Richtung vorzugeben ist und von den Bits 2 ... 4 immer nur ein Bit gesetzt werden darf!

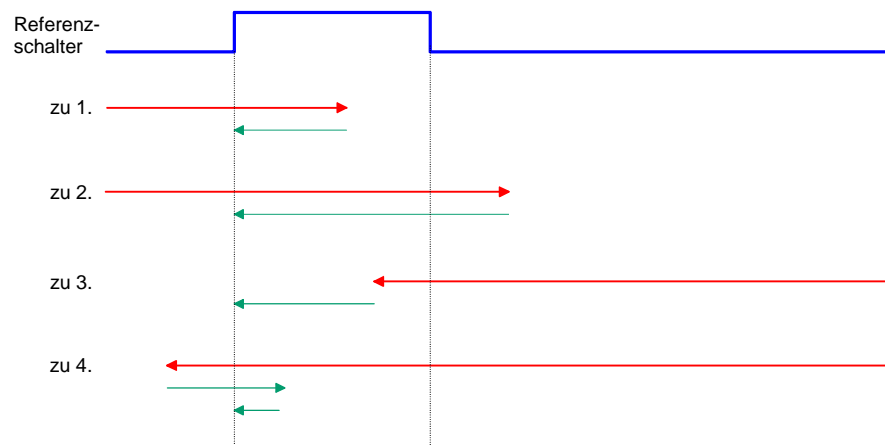
## Referenzfahrt auf Referenzschalter

Bei der Referenzfahrt wird immer zuerst mit der in FREF vorgegebenen Geschwindigkeit gestartet. Die Richtung muss im Variablen Parameter (Byte 15, Bit 0) vorgegeben werden. Sobald die steigende Flanke des Referenzschalters erkannt wird, bremst der Motor auf FSTART ab.

Abhängig von der Referenzgeschwindigkeit kann der Antrieb während des Bremsens über den Referenzschalter hinausfahren oder nicht.

Folgende 4 Anfahrten an den Referenzschalter sind möglich:

1. Motor kommt von links, bremst innerhalb des Referenzschalters ab und fährt rückwärts mit FSTART bis die fallende Flanke des Referenzschalters erkannt wird.
2. Motor kommt von links, fährt beim Bremsen über den Referenzschalter hinaus und fährt rückwärts mit FSTART über die steigende Flanke hinweg, bis die fallende Flanke des Referenzschalters erkannt wird.
3. Motor kommt von rechts, bremst innerhalb des Referenzschalters ab, und fährt mit FSTART bis die fallende Flanke des Referenzschalters erkannt wird.
4. Motor kommt von rechts, fährt beim Bremsen über den Referenzschalter hinaus, er ändert seine Drehrichtung und fährt mit FSTART bis die steigende Flanke des Referenzschalters erkannt wird, dreht nochmals die Richtung und fährt bis die fallende Flanke des Referenzschalters erkannt wird.



## Referenzfahrt auf Endschalter

Sie können über die Endschalter A und B Ihre Strecke begrenzen.

Bei der Referenzfahrt auf Endschalter startet der Antrieb und fährt mit der voreingestellten Geschwindigkeit Fref und der vorgegebenen Drehrichtung bis der entsprechende Endschalter aktiv wird, stoppt abrupt, ändert seine Drehrichtung und fährt mit Fstart bis der Endschalter wieder inaktiv wird.



### Hinweis!

Wird die Referenzfahrt über Endschalter realisiert, so ist zwingend darauf zu achten, dass hinter dem Endschalter genügend Platz zum Abrampen des Motors vorhanden ist!



## FM 253 - Datenübergabe > CPU

### Rückmeldung

Von dem MotionControl Stepper Modul wird zyklisch an die CPU ein Datenblock geschickt, der verschiedene Informationen zum aktuellen Status des Antriebs beinhaltet. Der Datenblock ist 16Byte lang und hat folgenden Aufbau:

Byte-Nr.	Inhalt	Länge
0-3	Ist-Position	4Byte
4-7	Ist-Frequenz	4Byte
8-9	Fehlermeldungen	2Byte
10	aktueller Modus	1Byte
11	Status	1Byte
12-15	Variablen Daten	4Byte

### Ist-Position, Ist-Frequenz

Über diese zwei Parameter wird immer die aktuelle Ist-Position und Ist-Frequenz Ihres Antriebs angezeigt.

### Fehlermeldungen

Die aktuell erkannten Fehler werden in den Fehler-Bits von Byte 8-9 zurückgemeldet. Die Fehler bleiben solange aktiv, bis die entsprechenden Bits zurückgesetzt werden.

Solange ein Fehler ansteht, wird nach dem Rücksetzen das entsprechende Fehler-Bit wieder gesetzt.

Folgende Fehlermeldungen werden verwendet:

#### *Fehlerbyte (Byte 8-9)*

Byte 9	Beschreibung
Bit 0	Fehler in der internen Zustandsverwaltung
Bit 1	System hat neu gebootet (kommt immer nach PowerON)
Bit 2	Fehler beim Überprüfen der Parameter im Flash, Motorparameter nicht mehr gültig
Bit 3	Funktion ist nicht erlaubt während der Motor läuft
Bit 4	Motor ist derzeit gesperrt
Bit 5	Fehler beim Positionieren des Motors
Bit 6	Endschalter ist/war aktiv
Bit 7	Frequenz wurde auf Fmax begrenzt
Byte 8	
0	allgemeiner Fehler beim Motor

**Fehlermeldungen zurücksetzen**

Zum Löschen eines anstehenden Fehlers (Byte 8-9) ist das entsprechende Fehlerbit in den Variablen Parameter (Byte 14-15) auf "1" zu setzen.

Sobald Sie den **Modus (Byte 10)** auf **7** setzen, werden die entsprechenden Fehler im Modul zurückgesetzt. Sie können auch mehrere Fehlermeldungen gleichzeitig zurücksetzen. FFFFh in Byte 14-15 setzt beispielsweise alle Fehler zurück.

**Aktueller Modus**

Hier finden Sie immer den Modus, in dem sich Ihr FM 253 aktuell befindet. Folgende Modi können zurückgemeldet werden:

*Modus (Byte 10)*

Byte	Modus
10	00h: IDEL 01h: Positionieren relativ 02h: Referenzfahrt 03h: Dauerlauf Achse 04h: Eingänge lesen 05h: Motor Parameter bearbeiten 06h: Position setzen 07h: Fehler löschen 08h: Positionieren absolut

**Status**

Das Status-Byte gibt Aufschluss über den Zustand des Antriebs. Folgende Statusmeldungen können ausgegeben werden:

*Status (Byte 11)*

Byte 11	Status
Bit 0	1: Antrieb in Fahrt 0: Antrieb in Stop
Bit 1	1: Fahrtrichtung vorwärts 0: Fahrtrichtung rückwärts
Bit 2	1: Antrieb in Position 0: Antrieb nicht in Position

**Eingänge lesen**

Zum Lesen der Eingänge wird der **Modus (Byte 10)** auf **4** gesetzt und nun gibt das Modul in den Variablen-Daten (Byte 15) den Zustand der Endschalter und des Referenzschalters an.

*Eingänge (Byte 15)*

Byte 15	Eingang
Bit 0	Zustand PA Endschalter (1: betätigt, 0: nicht betätigt)
Bit 1	Zustand PB Endschalter (1: betätigt, 0: nicht betätigt)
Bit 2	Zustand RE Referenzschalter (1: betätigt, 0: nicht betätigt)

## FM 253 - Hantierungsbausteine

### Übersicht

Zusammen mit FM 253 sind Hantierungsbausteine erhältlich, die die Bedienung des Moduls komfortabler gestalten sollen. Folgende Hantierungsbausteine sind zur Zeit für das FM 253 verfügbar:

Baustein	Bedeutung
FC 200	Antrieb ansteuern
FC 201	Einstellung eines Parameters
FC 202	Einstellung aller Fahr-Parameter (Index 0...9)

### FC 200 Antrieb ansteuern

Dieser FC dient zur Ansteuerung Ihres Antriebs indem Sie durch Angabe des entsprechenden Modus die Fahrdaten an das Modul übergeben.

Mit diesem FC können Sie alle Modi bis auf "Parameter setzen" und die zugehörigen Parameter an das Modul übertragen.

### Datenübergabe an FM 253 mit SET\_MODE = 1

- Geben Sie den Modus an.
- Versorgen Sie die entsprechenden Parameter mit Daten.
- Starten Sie die Übertragung indem Sie SET\_MODE auf 1 setzen. Ist der Mode gestartet, setzt im nächsten Zyklus der Baustein SET\_MODE wieder zurück und liefert die "IST"-Daten des FM 253.

### Datenübergabe an CPU mit SET\_MODE = 0

Bei Aufruf des FC 200 mit SET\_MODE = 0, werden über die Bezeichner IST\_POSITION, IST\_FREQUENZ, IST\_MODE, FEHLER, STATUS und VAR\_DATEN die "IST"-Daten des FM 253 ausgegeben.

Es ist sinnvoll die einzelnen Werte in einem Datenbaustein anzulegen. Im nachfolgenden Beispiel wurde hierzu der DB 5 verwendet.

### Parameter

Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar
0.0	in	ADRESSE	INT		Eingestellte Basisadresse
2.0	in	SOLL_POSITION	DINT		Übergabe Positionswerte
6.0	in	SOLL_FREQUENZ	DINT		Übergabe Frequenz bei Dauerlauf
10.0	in	VARIABLEN	DWORD		Übergabe von Variablen bei Referenzfahrt
14.0	in	MODE	INT		Übergabe zu bearbeitender Modus
16.0	out	IST_POSITION	DINT		Rückgabe aktuelle Position
20.0	out	IST_FREQUENZ	DINT		Rückgabe aktuelle Frequenz
24.0	out	FEHLER	INT		Fehlerwort
26.0	out	IST_MODE	INT		Rückgabe aktueller Modus
28.0	out	STATUS	BYTE		Rückgabe Statusbits
30.0	out	VAR_DATEN	DWORD		Rückgabe Variablen
34.0	in_out	SET_MODE	BOOL		Funktion starten

**ADRESSE** Anfangs-Adresse, ab der das FM 253 in der CPU abliegt.

**SOLL\_POSITION** Im Modus 01, 06 und 08 geben Sie hier die Sollposition für den Antrieb an.

**SOLL\_FREQUENZ** Im Modus 03 tragen Sie hier die Solldrehzahl als Soll-Frequenz ein.

**VARIABLEN** Geben Sie hier die Steuerbits für die Referenzfahrt (MODE = 02) und zum Fehler-Rücksetzen (MODE = 07) an.

Die Steuerbits für die Referenzfahrt haben folgende Belegung:

*Steuerbits*

VARIABLEN-Byte	Parameter
Bit 0	1: Richtung vorwärts 0: Richtung rückwärts
Bit 1	1: nach Referenzfahrt Position löschen 0: nach Referenzfahrt Position nicht löschen
Bit 2	Referenzfahrt auf Referenzschalter
Bit 3	Referenzfahrt auf Endschalter B
Bit 4	Referenzfahrt auf Endschalter A

Eine Übersicht der Fehler-Bit-Zuordnung finden Sie auf der Folgeseite.

**MODE** Mit diesem Parameter übergeben Sie den Modus an das FM 253. Folgende Modi sind möglich:

*Mode*

Wert	Wert	Beschreibung	Vorgabe in	Rückgabe in
00		Idle-Mode - es erfolgt keine Zustandsänderung des Antriebs, dient zur Parameteränderung	-	-
01		Positionieren relativ - Anfahren der vorgegebenen Schrittzahl	SOLL_POSITION	-
02		Referenzfahrt - Kalibrierung des Antriebs	VARIABLEN	-
03		Dauerlauf Achse - Antrieb dreht mit Sollfrequenz	SOLL_FREQUENZ	-
04		Eingänge lesen - gibt Zustände der Endschalter zurück	-	VAR_DATEN
06		Position setzen - setzt in Modul die Ist-Position ohne Antrieb zu bewegen	SOLL_POSITION	-
07		Fehler löschen - löscht das mit 1 aktivierte Fehlerbit	VARIABLEN	-
08		Positionieren absolut - Anfahren der Sollposition	SOLL_POSITION	-

**IST\_POSITION,  
IST\_FREQUENZ**

Über diese zwei Parameter wird die aktuelle Ist-Position und Ist-Frequenz Ihres Antriebs angezeigt.

**FEHLER**

Hier finden Sie eventuelle Fehlermeldungen. Die Fehler bleiben solange aktiv, bis die Fehlerursache beseitigt ist und die entsprechenden Bits zurückgesetzt werden. Folgende Fehlermeldungen werden verwendet:

*Fehlermeldungen*

FEHLER-Byte 1	Beschreibung
Bit 0	Fehler in der internen Zustandsverwaltung
Bit 1	System hat neu gebootet (kommt immer nach PowerON)
Bit 2	Fehler beim Überprüfen der Parameter im Flash, Motorparameter nicht mehr gültig
Bit 3	Funktion ist nicht erlaubt während der Motor läuft
Bit 4	Motor ist derzeit gesperrt
Bit 5	Fehler beim Positionieren des Motors
Bit 6	Endschalter ist/war aktiv
Bit 7	Frequenz wurde auf Fmax begrenzt
FEHLER-Byte 0	
0	allgemeiner Fehler beim Motor

Das Rücksetzen von Fehlermeldungen erfolgt mit MODE = 07 und VARIABLE = Fehlerbytes

**IST\_MODE**

Gibt den Modus zurück, in dem sich Ihr Modul aktuell befindet.

**STATUS**

Das Status-Byte gibt Aufschluss über den Zustand des Antriebs. Folgende Statusmeldungen können ausgegeben werden:

*Status*

STATUS-Byte	Status
Bit 0	1: Antrieb in Fahrt 0: Antrieb in Stop
Bit 1	1: Fahrtrichtung vorwärts 0: Fahrtrichtung rückwärts
Bit 2	1: Antrieb in Position 0: Antrieb nicht in Position

**VAR\_DATEN**

In VAR\_DATEN wird der Zustand der Eingänge zurückgegeben, nachdem Sie diesen mit MODE = 04 angefordert haben. Zum Lesen der Eingänge wird der **Modus 4** gesetzt und nun gibt das Modul in den Variablen-Daten (Byte 15) den Zustand der Endschalter und des Referenzschalters an.

*Eingänge*

VAR_DATEN-Byte	Eingang
Bit 0	Zustand PA Endschalter (1: betätigt, 0: nicht betätigt)
Bit 1	Zustand PB Endschalter (1: betätigt, 0: nicht betätigt)
Bit 2	Zustand RE Referenzschalter (1: betätigt, 0: nicht betätigt)

**SET\_MODE**

Nachdem Sie die entsprechenden Parameter versorgt haben, werden mit SET\_MODE = 1 die Daten an Ihr Modul übertragen.

Ist der Mode gestartet, setzt im nächsten Zyklus der Baustein SET\_MODE wieder zurück und liefert die "IST"-Daten des FM 253.

**Beispiel**

```

DB 5
DBD 0 Position          DINT    L#0    Positionswert
DBD 4 Frequenz          DINT    L#0    Frequenz für Dauerlauf
DBW 8 reserve           WORD    W#16#0
DBW 10 MODE             INT     0     Mode
DBW 12 Index           INT     0     Index Vorgabe
DBD 14 Variablen_PARAM DWORD   DW#16#0  Var. für Ref.Fahrt/Param...
DBW 18 Reserve1        WORD    W#16#0
DBD 20 Ist_Position     DINT    L#0    aktuelle Position
DBD 24 Ist_Frequenz     DINT    L#0    aktuelle Frequenz
DBW 28 Fehler           INT     0     Fehleranzeige
DBW 30 Ist_Mode         INT     0     aktueller Mode
DBW 32 Status           BYTE    B#16#0  Status-Rückmeldung
DBD 34 VAR_DATEN       DWORD   DW#16#0  Rückgabe Parameter/Daten

CALL FC 200 //FC für StepperModul
  ADRESSE      :=128 //Baugruppenadresse
  SOLL_POSITION:=DB5.DBD 0 //DBD mit Position für absolut/relativ
  SOLL_FREQUENZ:=DB5.DBD 4 //DBD mit Frequenz für Dauerlauf
  VARIABLEN    :=DB5.DBD14 //Daten für Ref_Fahrt/Fehler löschen
  MODE         :=DB5.DBW10 //Modvorgabe für neuen Auftrag
  SET_MODE     :=M1.0 //Start Auftrag
  IST_POSITION :=DB5.DBD20 //aktuelle Position
  IST_FREQUENZ :=DB5.DBD24 //aktuelle Frequenz
  FEHLER      :=DB5.DBW28 //Anzeige Fehler
  IST_MODE     :=DB5.DBW30 //aktueller Mode
  STATUS      :=DB5.DBW32 //Statusbits vom Modul
  VAR_DATEN   :=DB5.DBD34 //Rückgabe von Werten
                                     z.B. Eingänge lesen

```

**FC 201 - einen Parameter setzen**

Mit dem FC 201 ist es möglich einen Parameter auf dem FM 253 zu setzen.

**Parameter**

Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar
0.0	in	ADRESSE	INT		Eingestellte Basisadresse
2.0	in	INDEX	INT		Übergabe INDEX für Parameter
4.0	in	PARAMETER	DWORD		Übergabe Parameterwert
	out				
8.0	in_out	SET_PARA	BOOL		Start Parameterübergabe

**ADRESSE**

Anfangs-Adresse, ab der das FM 253 in der CPU abliegt.

**INDEX**

Über INDEX stellen Sie den Parameter ein, dessen Wert Sie in PARAMETER vorgeben können.

Index	Parameter	Einheit	Wertebereich	Default	Bemerkung
00h	Fstart	Hz	UINT32	200	Startfrequenz
01h	F1	Hz	UINT32	4000	Grenzfrequenz 1
02h	dF1	Hz	UINT32	100	Beschleunigung von Fstart ⇒ F1
03h	F2	Hz	UINT32	10000	Grenzfrequenz 2
04h	dF2	Hz	UINT32	60	Beschleunigung von F1 ⇒ F2
05h	Fmax	Hz	UINT32	30000	Maximale Fahrfrequenz
06h	dFmax	Hz	UINT32	40	Beschleunigung von F2 ⇒ Fmax
07h	Fpos	Hz	UINT32	30000	Frequenz beim Positionieren
08h	Fref	Hz	UINT32	1000	Frequenz für die Referenzfahrt
10h	Fist	Hz	UINT32	-	Aktuelle Motorfrequenz (nur lesen)
11h	Fsoll	Hz	UINT32	-	Aktuelle Sollfrequenz (nur lesen)
13h	FTarget	Hz	UINT32	-	Zielfrequenz (nur lesen)
97h				-	Parameter in Flash ablegen
98h				-	Parameter aus Flash lesen (Zustand wie nach PowerON)
99h				-	Default-Parameter laden

**PARAMETER**

Tragen Sie hier den Wert für den Parameter ein, den Sie über INDEX spezifiziert haben.

**SET\_PARA**

Nachdem Sie die entsprechenden Parameter versorgt haben, wird mit SET\_PARA = 1 der Parameter an Ihr Modul übertragen. Nach der Übertragung wird SET\_PARA automatisch zurückgesetzt.

**FC 202 - FM 253  
parametrieren**

Über den FC 202 können Sie alle wichtigen Parameter des FM 253 einstellen.

**Parameter**

Adresse	Deklaration	Name	Typ	Anfangswert	Kommentar
0.0	in	DATEN_DB	BLOCK_DB		Datenbaustein mit Parametern
2.0	in	ADRESSE	INT		Baugruppenadresse
	out				
4.0	in_out	START	BOOL		Start Parameterübergabe
4.1	in_out	LAUF	BOOL		Übergabe einzel läuft

**DATEN\_DB**

Geben Sie hier den Datenbaustein an, in dem sich Ihre Parameter befinden.

Der DB hat folgenden Aufbau:

DBD 0	Fstart	DINT	L#0	Startfrequenz
DBD 4	F1	DINT	L#0	Grenzfrequenz 1
DBD 8	F2	DINT	L#0	Grenzfrequenz 2
DBD 12	Fmax	DINT	L#0	Maximale Fahrfrequenz
DBD 16	dF1	DINT	L#0	Beschleunigung Fstart --> F1
DBD 20	dF2	DINT	L#0	Beschleunigung F1 --> F2
DBD 24	dFmax	DINT	L#0	Beschleunigung F2 --> Fmax
DBD 28	Fpos	DINT	L#0	Frequenz beim Positionieren
DBD 32	Fref	DINT	L#0	Frequenz bei Referenzfahrt
DBD 36	StepRepeat	DINT	L#0	Schritt zwischen Frequenzberechnung

**ADRESSE**

Anfangs-Adresse, ab der das FM 253 in der CPU abliegt.

**START**

Nachdem Sie den DB erzeugt haben, können Sie mit START = 1 die Parameter an Ihr Modul übertragen.

Sobald alle Parameter übergeben sind, wird START wieder zurückgesetzt.

**LAUF**

In dieser Variable wird zyklusübergreifend ein Zustand gespeichert und ist für die Einzelparameterübergabe erforderlich.



## FM 254 - MotionControl Servo

### Leistungsmerkmale

- Mikroprozessorgesteuertes Positionier-Modul für Antriebe mit analoger Sollwert-Schnittstelle ( $\pm 10V$  Regleransteuerung)
- 7 Betriebsarten
- Lagegeregeltes Positionieren
- Das Modul arbeitet selbständig und wird über ein Anwenderprogramm im System 200V gesteuert.
- Daten werden in einem Flash-RAM gespeichert. Es ist keine Batterie erforderlich.

### Einsatzgebiete

Sie können das sowohl für einfache Positionierungen als auch für Verfahrprofile einsetzen, die höchste Ansprüche an Dynamik, Genauigkeit und Geschwindigkeit erfordern.

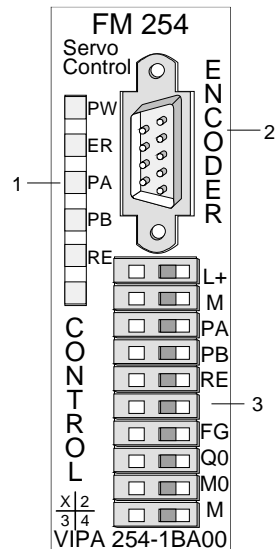
Aufgrund der unterschiedlichen Betriebsmodi lässt sich das Modul auch für Positionieraufgaben in Maschinen mit hohen Taktraten einsetzen.

Typische Einsatzgebiete sind:

- Förder- und Transporteinrichtungen, Transferstraßen und Montagelinien
- Pressen
- Holzbearbeitungsmaschinen
- Handhabungsgeräte
- Beschickungseinrichtungen
- Verpackungsmaschinen
- Hilfsbewegungen bei Dreh- und Fräsmaschinen

## FM 254 - MotionControl Servo - Aufbau

### Frontansicht



- [1] LED Statusanzeigen
- [2] Anschluss für Drehgeber
- [3] Anschluss für Versorgungsspannung, Antrieb, Endschalter und Ausgänge

### Komponenten

#### LEDs

Das FM 254 besitzt 6 LEDs, die der Statusanzeige dienen.

Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
PW	Gelb	DC 24V Versorgungsspannung liegt an
ER	Rot	interner Fehler
PA	Grün	Grenzwert A überfahren, Eingang PA gesetzt
PB	Grün	Grenzwert B überfahren, Eingang PB gesetzt
RE	Grün	Referenzpunkt überfahren
FG	Grün	Freigabe des Antriebs

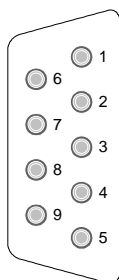


#### Hinweis!

Sollte im Betrieb die PW-LED nicht leuchten, so könnte dies an einem Kurzschluss in der DC 24V-Spannungsversorgung liegen.

Überprüfen Sie hierzu auch die Anschlüsse des Encoder-Steckers.

Bleibt bei abgezogenem Encoder-Stecker die Power-LED aus, so ist das Modul defekt.

**Encoder-  
Schnittstelle***9-poligen SubD Stecker*

Pin	Belegung
1	Geberversorgung +24V
2	Geberversorgung +5V
3	R+ Takteingang Nullimpuls
4	B+ Takteingang
5	A+ Takteingang
6	Masse Geberversorgung
7	R- Takteingang Nullimpuls
8	B- Takteingang
9	A- Takteingang

**Control-  
Schnittstelle**

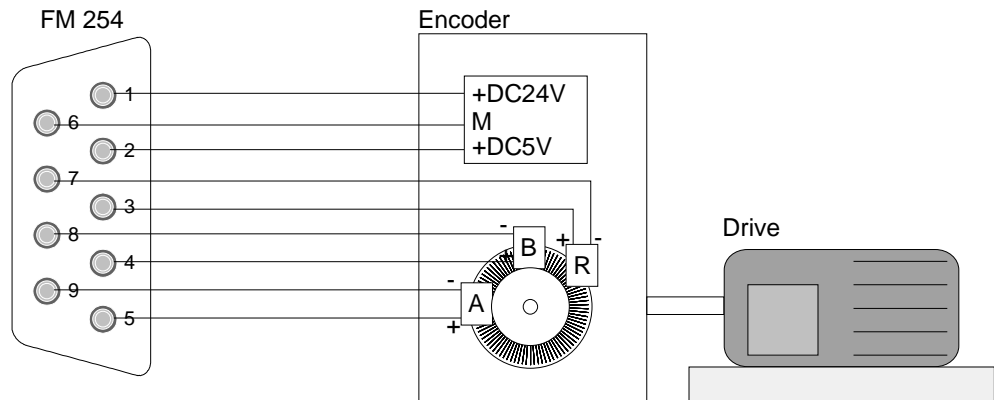
Pin	Belegung
1	Versorgungsspannung DC 24V
2	Masse 24V
3	Eingang Anfangsschalter negierter Eingang
4	Eingang Endschalter negierter Eingang
5	Eingang Referenzschalter negierter Eingang
6	reserviert
7	Ausgang Reglerfreigabe
8	Analogausgang Masse
9	Analogausgang +
10	Schirm

## FM 254 - Anschluss eines Antriebs mit Drehgeber

### Anschluss eines Drehgebers (Encoder)

Der Anschluss eines Drehgebers erfolgt über den 9-poligen SubD-Stecker. Das Modul liefert die DC 24V und DC 5V Spannungsversorgung für den Drehgeber.

Die nachfolgende Abbildung zeigt den Anschluss eines Drehgebers:



### Anschluss von Versorgungs- spannung, Antrieb, Endschalter und Ausgänge

#### Spannungsversorgung

Über Pin 1 und 2 ist das Modul mit DC 24V zu versorgen.

#### Endschalter

An das Modul können Sie bis zu 3 Endschalter (Öffner) anschließen.

An Klemme 3 und 4 (PA und PB) schließen Sie die Schalter an, mit denen Sie die Strecke begrenzen. Sobald einer dieser Schalter betätigt wird, wird der Antrieb sofort angehalten und kann dann nur noch in die Gegenrichtung gefahren werden.

An Klemme 5 schließen Sie den Referenzschalter an, der für die Abstimmung Ihres Antriebs auf das MotionControl Servo Modul erforderlich ist.

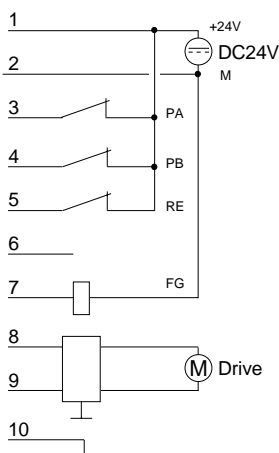
Der Endschalter, der im Modus Hardware-Fahrt den Antrieb zum Stillstand bringt, ist auch an Klemme 5 anzuschließen.

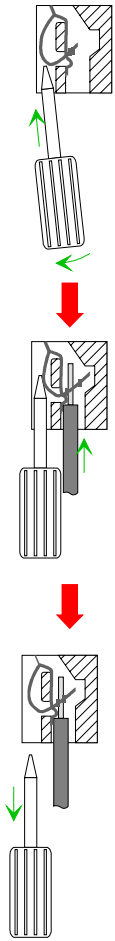
#### Ausgänge

Das Modul besitzt 2 Ausgänge, die das Modul direkt ansteuert. Zur Zeit wird aber nur der Ausgang "Regler Freigabe" (Pin 7) angesteuert. Der andere Ausgang ist für zukünftige Erweiterungen. Die Freigabe aktivieren Sie durch Setzen von Bit 0 in den Fahr-Daten.

#### Antrieb

An Pin 8 und 9 wird ein Analogsignal geliefert, das zur  $\pm 10V$  Regler-ansteuerung eines Antriebs verwendet werden kann.



**Verdrahtung**

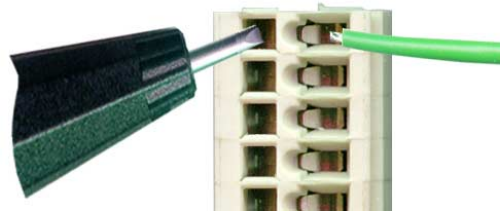
Der Antrieb und die Endschalter sind an "Control" anzuschließen. Hierbei wird ein 10-poliger Stecker mit Federklemmtechnik der Firma WAGO eingesetzt. Die Verdrahtung mit Federklemmtechnik ermöglicht eine schnelle Verdrahtung. Im Gegensatz zu Schraubverbindungen ist diese Technik erschütterungssicher.

Sie können Drähte mit dem Querschnitt von  $0,08\text{mm}^2$  bis  $1,5\text{mm}^2$  anschließen.

Die Verdrahtung geschieht auf die gleiche Weise wie bei den großen Federklemmen des System 200V.

Drücken Sie mit einem passenden Schraubendreher die Federklemme in der quadratische Öffnung nach hinten und führen Sie durch die rechteckige Öffnung den Draht ein.

Durch Herausziehen des Schraubendrehers wird der Draht sicher gehalten.



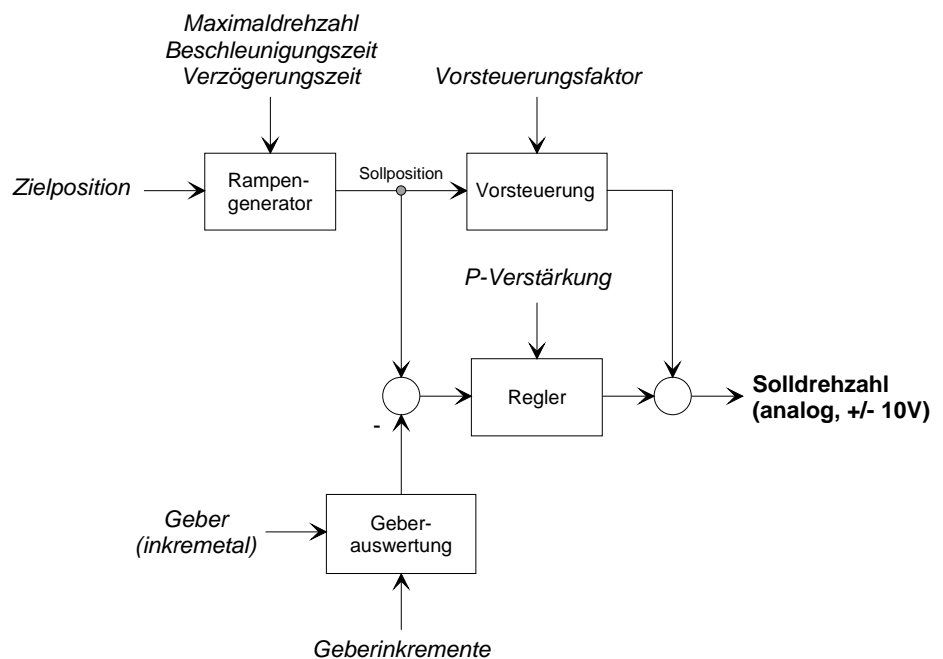
## FM 254 - Übersicht der Parameter und Übergabewerte

In der nachfolgenden Tabelle sind alle Parameter und Übergabewerte aufgeführt. Das Zusammenspiel der Parameter ist im Blockdiagramm dargestellt.

### Übersicht

Wert	Größe	Einheit	physikalischer Wertebereich
Zielposition Sollposition Istposition	32 Bit	$\frac{1 \text{ Umdrehung}}{2^{16}}$	0,0 ... 65535,9999 Umdr.
Maximaldrehzahl	16 Bit	1/min	10 ... 1000 1/min
Beschleunigungszeit Verzögerungszeit	16 Bit	10 ms	10 ms ... 100 s
P-Verstärkung	16 Bit	0,1	0,0 ... 1000,0
Vorsteuerungsfaktor	16 Bit	0,1	0,0 ... 1,0
Geber-Strichzahl	16 Bit	1	10 ... 10000
Betriebszustand	16 Bit	bitcodiert	

### Blockdiagramm



## FM 254 - Parametrierung

Bei der Inbetriebnahme sind zur Parametrierung des MotionControl Servo Moduls 16 Byte Parameterdaten zu übergeben. Die Parametrierdaten haben folgenden Aufbau:

### Parameter-Daten (nur schreiben)

Byte-Nr.	Name	Länge	Wertebereich	Einheit
1, 0	Maximaldrehzahl	2Byte	10.....1000	1/min
3, 2	reserviert	2Byte	-	-
5, 4	reserviert	2Byte	-	-
7, 6	P-Verstärkung	2Byte	0,0 ... 1000,0	0,1
9, 8	Vorsteuerungsfaktor	2Byte	0,0 ... 1,0	0,1
11, 10	Geberstrichzahl	2Byte	10 ... 10000	1
13, 12	Referenzdrehzahl	2Byte	10 ... 1000	1/min
14	Pos-erreicht-Fenster	1Byte	0.....255	1INK
15	Schleppabstand	1Byte	4.....1020	4INK

### Parameter- beschreibung

#### *Maximaldrehzahl*

Stellen Sie hier die Drehzahl ein, die Ihr Antrieb maximal annehmen soll.

#### *P-Verstärkung, Vorsteuerungsfaktor*

Über diese zwei Werte können Sie das Regelverhalten beeinflussen.

#### *Geberstrichzahl*

Mit diesem Parameter passen Sie Ihr MotionControl Servo Modul an den angeschlossenen Drehgeber an.

#### *Referenzdrehzahl*

Diese Drehzahl wird für die Referenzfahrt verwendet, bei der sich das MotionControl Servo Modul neu auf die Regelstrecke einstellt.

#### *Pos-erreicht-Fenster*

Nach Erreichen der Zielposition wird der Antrieb durch fortlaufende Regelung in der Zielposition gehalten. Der Antrieb steht nicht still.

Mit dem *Pos-erreicht-Fenster* können Sie ein Bereichsfenster in Inkrementen angeben um den der Ist-Wert bei erreichter Position abweichen darf ohne dass eine Regelung stattfindet, der Antrieb also still steht.

#### *Schleppabstand*

Über diesen Parameter bestimmen Sie, ab welchem Schleppfehler, d.h. Differenz zwischen Ist- und Soll-Wert, der Antrieb gestoppt wird.

## FM 254 - Datenübergabe >> FM 254

### Fahr-Daten

Zur Ansteuerung des MotionControl Servo Moduls können folgende Werte von der CPU auf das Modul FM 254 geschrieben werden.

Byte-Nr.	Name	Länge	Wertebereich	Einheit
3, 2, 1, 0	Zielposition	4Byte	32 Bit Integer	Geberinkremente
5, 4	Steuerungsbytes	2Byte		
7, 6	Drehzahl	2Byte	10.....1000	1/min
9, 8	Beschleunigungszeit	2Byte	1...10000	10ms
13, 12, 11, 10	Parameterfeld	4Byte		
15, 14	Feldkennung	2Byte		

### Steuerungsbytes (Byte 4 und Byte 5)

Byte	Bit 7 ... Bit 0
4	Bit 0: Freigabe Bit 1: Betriebsart Referenzfahrt positiv Bit 2: Betriebsart Referenzfahrt negativ Bit 3: Betriebsart Hardware-Fahrt positiv Bit 4: Betriebsart Hardware-Fahrt negativ Bit 5: Betriebsart Kettenmaß Bit 6: Betriebsart Kettenmaß endlos Bit 7: Übernahme Zielposition
5	Bit 0: reserviert Bit 1: Tippbetrieb Drehrichtung positiv Bit 2: Tippbetrieb Drehrichtung negativ Bit 7 ... Bit 3: reserviert

Die Betriebsarten sind auf den Folgeseiten beschrieben.

### Parameterfeld und Feldkennungen (Byte 10 ... Byte 14)

Über die Angabe einer Feldkennung können Sie weitere Parameter zusammen mit den Fahrdaten an das FM 254 übergeben. Die Parameter für die entsprechende Feldkennung sind im Parameterfeld einzutragen (Byte 10...13).

Sofern keine Feldkennungen übertragen werden, arbeitet das FM 254 mit den angegebenen Default-Einstellungen.

Feldkennng.	Bedeutung	Wertebereich	Einheit	Default-Einstellung
FF01h	Software-Endschalter (+)	32 Bit Integer	Geberinkremente	7FFF.FFFF
FF02h	Software-Anfangssch.(-)	32 Bit Integer	Geberinkremente	8000.0001
FF03h	Drehzahl Tippbetrieb	10.....1000	1/min	Referenzdrehzahl
FF04h	Verzögerungszeit	1...10000	10ms	Beschleunigungszeit



## FM 254 - Betriebsarten

### Übersicht

Durch das Setzen entsprechender Bits in den Steuerungsbytes können Sie folgende Betriebsarten einstellen:

- Positionierbetrieb (Anfahrt einer absoluten Zielposition)
- Referenzfahrt (Kalibrierung des Antriebsystems)
- Hardwarefahrt (Fahrt bis Referenzschalter)
- Kettenmaß (Fahrt bei relativer Zielangabe durch Addition)
- Kettenmaß endlos (relative Fahrt ohne Zählerüberlauf)
- Tippbetrieb

### Positionierbetrieb

#### Funktion

Im Positionierbetrieb wird die absolute Zielposition erst dann an das FM 254 übergeben, wenn das Bit "Übernahme Zielposition" sitzt.

Wird mit gesetztem Freigabebit eine neue Position vorgegeben, fährt der Antrieb mit den zuvor eingestellten Drehzahl- und Beschleunigungswerten selbständig an die gewünschte Position  $\pm$  Pos-erreicht-Fenster und setzt das "Position erreicht"-Bit. Nach der Übergabe der Fahrparameter starten Sie den Antrieb durch Setzen des Freigabe-Bits. Während der Fahrt zeigt das Modul durch Setzen von Bit 1 bzw. 2 die Drehrichtung an.

Sobald die Abweichung zwischen Soll- und Ist-Position größer ist als das angegebene Schleppfehler-Fenster, wird die Positionierung abgebrochen und der Motor angehalten. Dies wird dem Programm durch Setzen des Schleppfehler-Bits Bit 0 im Byte 5 angezeigt. Durch Rücksetzen des Freigabe-Bits können Sie das Schleppfehler-Bit wieder löschen. Hierbei wird die Soll-Position gleich der Ist-Position gesetzt.

Der Antrieb stoppt auch, sobald Soft- bzw. Hardware-Schalter überfahren werden, die die Strecke begrenzen.

Durch Setzen des Freigabe-Bits können Sie die Fahrt jederzeit fortsetzen.

Die Beschleunigungs-/Verzögerungszeiten können vor Absetzen eines neuen Auftrags geändert werden.

Die Vorgabe einer neuen Drehzahl durch Änderung der Fahr-Daten ist immer möglich. Wird die Drehzahl während der Fahrt geändert, so wird anhand der aktuellen Beschleunigung/Verzögerung auf die neue Drehzahl geregelt.

#### Steuerungsbytes

Die Steuerungsbytes, mit denen Sie diese Betriebsart einstellen, sind Bestandteil der Fahrdaten.

Eine allgemeine Beschreibung der Fahrdaten finden Sie weiter oben.

Byte	Bit 7 ... Bit 0
4	Bit 0: Freigabe (Antrieb startet) Bit 6 ... Bit 1: 0 Bit 7: irrelevant
5	irrelevant

## Referenzfahrt

**Funktion** Die Referenzfahrt dient der Kalibrierung Ihres Antriebssystems. Der Referenzpunkt sollte im Verfahrensweg liegen.

Starten der Referenzfahrt:

- Setzen Sie das Freigabe-Bit.
- Geben Sie mit dem Bit "Referenzfahrt positiv" bzw. "Referenzfahrt negativ" die Referenzfahrt frei.
  - Der Antrieb fährt nun mit der im Parametersatz eingestellten Referenzdrehzahl an den Referenzpunkt.
  - Sobald der Referenzpunkt überfahren wird, wird der Referenz-Endschalter betätigt (LED RE geht aus).
  - Die Position des Referenzpunkts wird gespeichert.
  - Der Antrieb dreht nun zurück bis zum nächsten Gebernulimpuls.

Hiermit ist die Referenzfahrt beendet und es wird das Bit "Referenz erkannt" gesetzt.



### Hinweis!

Bitte beachten Sie, dass in der Betriebsart "Referenzfahrt" die Angabe einer Sollposition nicht erforderlich ist. Die Sollposition wird ignoriert.

**Steuerungsbytes** Die Steuerungsbytes, mit denen Sie diese Betriebsart einstellen, sind Bestandteil der Fahrdaten.

Eine allgemeine Beschreibung der Fahrdaten finden Sie auf Seite 11-30.

Byte	Bit 7 ... Bit 0
4	Bit 0: Freigabe (Antrieb startet) Bit 2 ... Bit 1: 01: Referenzfahrt positiv 10: Referenzfahrt negativ Bit 6 ... Bit 3: 0 Bit 7: irrelevant
5	irrelevant

## Betriebsart Hardware-Fahrt

### Funktion

Dieser Modus dient der reinen Zielfahrt, bis ein überfahrener Endschalter den Antrieb stoppt. Der Endschalter ist am Referenzschalter-Eingang anzuschließen.

Für die Fahrt werden die zuvor eingegebenen Drehzahl- und Beschleunigungs- bzw. Verzögerungswerte verwendet. Nach dem Erreichen des Endschalters wird die Position intern zwischengespeichert und der Antrieb mit der eingestellten Verzögerungszeit angehalten.

Nach dem Stillstand des Antriebs fährt dieser rückwärts den Endschalter an und bleibt da stehen. Daraufhin wird das Bit 3 gesetzt für "Position erreicht". Für die Rückwärtsfahrt verwendet das MotionControl Servo Modul die über die Parametrierung eingestellte Referenzdrehzahl.

Durch Rücksetzen und erneutem Setzen der Bits "Freigabe" und "HW-Ref positiv" wird eine neue Fahrt angestoßen.

Die Beschleunigungs-/Verzögerungszeit kann vor Absetzen eines neuen Auftrags geändert werden.

Wird die Drehzahl während der Fahrt geändert, so wird anhand der aktuellen Beschleunigung/Verzögerung auf die neue Drehzahl geregelt.



### Hinweis!

Bitte beachten Sie, dass in der Betriebsart "Hardware-Fahrt" die Angabe einer Sollposition nicht erforderlich ist. Die Sollposition wird ignoriert.

### Steuerungsbytes

Die Steuerungsbytes, mit denen Sie diese Betriebsart einstellen, sind Bestandteil der Fahrdaten.

Eine allgemeine Beschreibung der Fahrdaten finden Sie auf Seite 11-30.

Byte	Bit 7 ... Bit 0
4	Bit 0: Freigabe (Antrieb startet) Bit 2 ... Bit 1: 0 Bit 4 ... Bit 3: 01: Hardware-Fahrt positiv 10: Hardware-Fahrt negativ Bit 6 ... 5: 0 Bit 7: irrelevant
5	irrelevant

**Betriebsart  
Kettenmaß**

**Funktion**

Im Modus Kettenmaß werden relative Positionen verarbeitet, d.h. der Wert, der als Sollposition übergeben wird, wird auf die aktuelle Position addiert.

Mit Setzen des Freigabe-Bits bewegt sich der Antrieb um den eingestellten relativen Wert in positive bzw. negative Richtung. Hierbei verwendet der Antrieb die zuvor eingestellten Drehzahl- und Beschleunigungswerte. Wird eine negative Position eingegeben, fährt der Antrieb rückwärts.

Die Beschleunigungs-/Verzögerungszeit können Sie vor Absetzen eines neuen Auftrags ändern.

Wird die Drehzahl während der Fahrt geändert, so wird anhand der aktuellen Beschleunigung/Verzögerung auf die neue Drehzahl geregelt.

**Steuerungsbytes**

Die Steuerungsbytes, mit denen Sie diese Betriebsart einstellen, sind Bestandteil der Fahrdaten.

Eine allgemeine Beschreibung der Fahrdaten finden Sie auf Seite 11-30.

Byte	Bit 7 ... Bit 0
4	Bit 0: Freigabe (Antrieb startet) Bit 4 ... Bit 1: 0 Bit 5: 1 (Kettenmaß) Bit 6: 0 Bit 7: irrelevant
5	irrelevant

---

**Betriebsart**  
**Kettenmaß**  
**endlos**
**Funktion**

Hier wird nach der Freigabe der als Position übergebene Wert relativ angefahren, und nach Erreichen des Werts die Soll- und Istposition auf Null gesetzt. In diesem Modus können Sie den Antrieb ohne einen Zählerüberlauf in eine Richtung fahren.

Die Beschleunigungs-/Verzögerungszeit können Sie vor Absetzen eines neuen Auftrags ändern.

Die Vorgabe einer neuen Drehzahl ist immer möglich. Wird die Drehzahl während der Fahrt geändert, so wird anhand der aktuellen Beschleunigung/Verzögerung auf die neue Drehzahl geregelt.

**Steuerungsbytes**

Die Steuerungsbytes, mit denen Sie diese Betriebsart einstellen, sind Bestandteil der Fahrdaten.

Eine allgemeine Beschreibung der Fahrdaten finden Sie weiter oben.

Byte	Bit 7 ... Bit 0
4	Bit 0: Freigabe (Antrieb startet) Bit 5 ... Bit 1: 0 Bit 6: 1 (Kettenmaß endlos) Bit 7: irrelevant
5	irrelevant

## Betriebsart Tipp-Betrieb

**Funktion** Der Antrieb wird durch Setzen des Bit 0 in Byte 4 und zuvor eingestellter Drehzahl und Beschleunigung freigegeben. Durch Setzen von Bit 1 bzw. Bit 2 in Byte 5 wird eine Drehrichtung vorgegeben und der Antrieb startet. Der Antrieb stoppt sobald Bit 1 bzw. Bit 2 in Byte 5 zurückgesetzt wird.

**Steuerungsbytes** Die Steuerungsbytes, mit denen Sie diese Betriebsart einstellen, sind Bestandteil der Fahrdaten.  
Eine allgemeine Beschreibung der Fahrdaten finden Sie weiter oben.

Byte	Bit 7 ... Bit 0
4	Bit 0: Freigabe (Antrieb startet)
5	Bit 0: V1.08 - Reset Counter bei Tipp-Betrieb (Flanke 0 nach 1 setzt die Istposition auf null zurück) *1) Bit 1: 1 Drehrichtung positiv Bit 2: 1 Drehrichtung negativ



### Hinweis!

\*1) Das Rücksetzen des Zählers darf nur im Tipp-Betrieb erfolgen. Im Positionierbetrieb würde der Regler aufgrund des gesprungen Istwertes Schleppfehler melden.

## FM 254 - Datenübergabe >> CPU

Von dem MotionControl Servo Modul werden folgende Werte zyklisch an die CPU übergeben und dort abgelegt.

Byte-Nr.	Name	Länge	Wertebereich	Einheit
3, 2, 1, 0	Sollposition	4Byte	32 Bit Integer	Geberinkremente
7, 6, 5, 4	Istposition	4Byte	32 Bit Integer	Geberinkremente
9, 8	Solldrehzahl	2Byte	10.....1000	1/min
11, 10	Betriebszustand	2Byte	bitcodiert	
13, 12	reserviert	2Byte	-	-
15, 14	Antwort Feldkennung	2Byte		hex

### *Betriebszustand*

Byte	Bit 7 ... Bit 0
10	Bit 0: Freigabe erfolgt Bit 1: Rechtsdrehung Bit 2: Linksdrehung Bit 3: Position erreicht Bit 4: HW-Anfangsschalter betätigt Bit 5: HW-Endschalter betätigt Bit 6: HW-Referenzschalter betätigt Bit 7: Referenz erkannt
11	Bit 0: Schleppfehler erkannt Bit 4: SW-Endschalter linksdrehend Bit 5: SW-Endschalter rechtsdrehend Bit 7 ... 1: Irrelevant

### **Beispiel**

Ist z.B. das MotionControl Servo Modul in Ihrer CPU ab der Peripherie-Adresse PY128 abgelegt, so finden Sie die "Sollposition" ab PY128 bis PY131.

Die anderen Werte liegen gemäß der oberen Liste dahinter im Peripheriebereich.

Die 2Byte für den "Betriebszustand" beispielsweise befinden sich dann unter PY138...PY139.

## Technische Daten

### MotionControl Stepper FM 253

Elektrische Daten	VIPA 253-1BA00
Anzahl der Achsen	1
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme über Rückwandbus	typ. 320mA, max. 500mA
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse / Schnittstellen	
"Antriebs"-Schnittstelle	Steuerausgänge für Takt, Richtung und Freigabe mit Differenzsignal nach RS422
Max. Impulsfrequenz	200kHz
Digitale Eingänge	
Anzahl	3
Funktion	2 Endschalter, Referenzschalter
Signalspannung "0"	0 ... 5V
Signalspannung "1"	15 ... 28,8V
Digitale Ausgänge	
Anzahl	2
Funktion	"In Fahrt", "Position erreicht"
Ausgangsstrom je Ausgang	1A dauerkurzschlussfest
Potenzialtrennung	ja
Programmierdaten	
Eingabedaten	16Byte
Ausgabedaten	16Byte
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g



**MotionControl**  
**Servo Modul**  
**FM 254**

Elektrische Daten	VIPA 254-1BA00
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme	200mA
Stromaufnahme Rückwandbus	100mA
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse / Schnittstellen	
Encoder	Inkrementalencoder
Signalspannungen	5V nach RS 422
Versorgungsspannung	5,2V / 300mA 24V / 300mA
Eingangsfrequenz und Leitungslänge	max. 1MHz bei 10m Leitungslänge geschirmt max. 500kHz bei 35m Leitungslänge geschirmt
Control	
Sollwertausgang	-10 ... +10V
Digitale Eingänge	
Anzahl	3
Versorgungsspannung	DC 24V
Digitale Ausgänge	
Anzahl	1
Potentialtrennung	nein
Ausgangsstrom	0,5A
Lampenlast	5W
Programmierdaten	
Eingabedaten	16Byte
Ausgabedaten	16Byte
Parameterdaten	16Byte
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g



## Teil 12 Spannungsversorgungen

### Überblick

In diesem Kapitel sind die Spannungsversorgungen für das System 200V näher beschrieben.

Nachfolgend sind beschrieben:

- Spannungsversorgung 2A
- Montage und Verdrahtung
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 12 Spannungsversorgungen</b> .....	<b>12-1</b>
Sicherheitshinweise .....	12-2
Systemübersicht .....	12-3
PS 207/2 - Spannungsversorgung - Aufbau.....	12-4
PS 207/2CM - Spannungsversorgung mit Klemmen - Aufbau .....	12-6
Montage.....	12-8
Verdrahtung.....	12-9
Technische Daten.....	12-10

## Sicherheitshinweise

### Bestimmungsgemäße Verwendung

Die Spannungsversorgungen sind konstruiert und gefertigt:

- zur DC24V-Versorgung von System 200V Komponenten
- für den Einbau zusammen mit System 200V Komponenten auf einer Trageschiene
- für den Betrieb als DC24V stand-alone Spannungsversorgung
- für den Einbau in einen Schaltschrank mit ausreichender Lüftung
- für den industriellen Einsatz

**Nachfolgend finden Sie die Vorsichtsmaßnahmen, die beim Einsatz der System 200V Spannungsversorgungen einzuhalten sind.**



### Gefahr!

- Die Spannungsversorgungen dürfen ausschließlich in Bereiche eingebaut werden, die nur dem Instandhalter zugänglich sind!
- Die Spannungsversorgungen sind nicht zugelassen für den Einsatz in explosionsgefährdeten Umgebungen (EX-Zone)!
- Die Spannungsversorgungen sind vor dem Beginn von Installations- und Instandhaltungsarbeiten unbedingt freizuschalten, d.h. vor Arbeiten an einer Spannungsversorgung oder an der Zuleitung, ist die Spannungszuführung stromlos zu schalten (Stecker ziehen, bei Festanschluss ist die zugehörige Sicherung abzuschalten)!
- Anschluss und Änderungen dürfen nur durch ausgebildetes Elektro-Fachpersonal ausgeführt werden!
- Bedingt durch die kompakte Bauweise kann zur Gewährleistung einer ausreichenden Kühlung der Berühr- und Brandschutz nicht eingehalten werden. Aus diesem Grund ist der Brandschutz durch die Konstruktion der Umgebung des eingebauten Netzteils sicherzustellen (z.B. Einbau in einen Schaltschrank, der die Brandschutzordnungen erfüllt)!
- Bitte beachten Sie die nationalen Vorschriften und Richtlinien im jeweiligen Verwenderland (Installation, Schutzmaßnahmen, EMV ...).

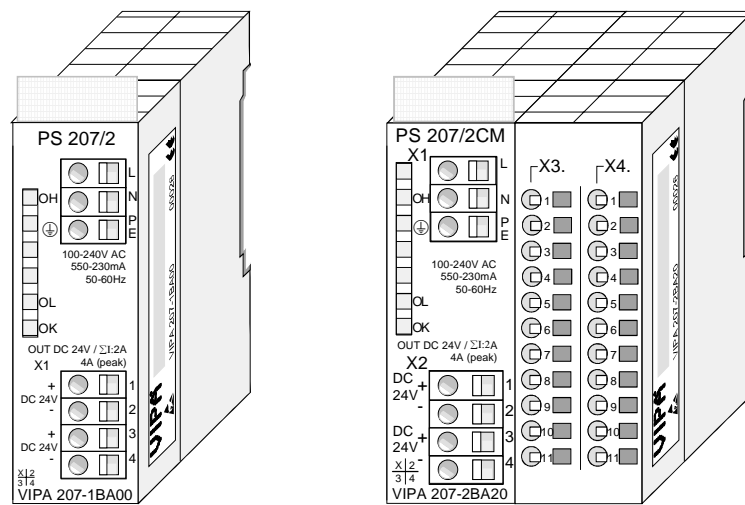
## Systemübersicht

### Allgemeines

Die hier vorgestellten System 200V Spannungsversorgungen besitzen einen Wide-Range-Input von AC 100 ... 240V. Sie haben eine Ausgangsspannung von DC 24V bei 2A/48W.

Da sich alle Ein- und Ausgänge auf der Frontseite befinden und das Gehäuse zum Rückwandbus isoliert ist, können Sie die Spannungsversorgung zusammen mit Ihrem System 200V auf die Hutschiene montieren oder als kompakte externe Spannungsversorgung verwenden.

Folgende Spannungsversorgungen sind zur Zeit erhältlich:



### Bestelldaten

Bestellnummer	Beschreibung
VIPA 207-1BA00	Netzteil primär AC 100...240V, sekundär DC24V, 2A, 48W
VIPA 207-2BA20	Netzteil primär AC 100...240V, sekundär DC24V, 2A, 48W mit Klemmenmodul 2x11 Klemmen

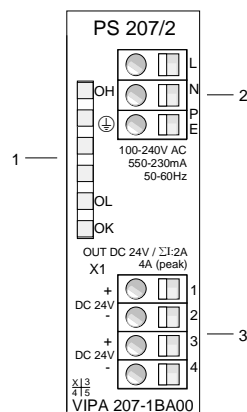
## PS 207/2 - Spannungsversorgung - Aufbau

### Eigenschaften

Die Spannungsversorgung zeichnet sich durch folgende Eigenschaften aus:

- Wide-Range-Input AC 100...240V ohne manuelle Umschaltung
- Ausgangsspannung DC 24V, 2A, 48W
- Einsetzbar zusammen mit System 200V auf Tragschiene oder als "stand-alone" Gerät
- Schutz gegen Kurzschluss, Überlast und Leerlauf
- Wirkungsgrad typ 90% bei  $I_{nenn}$

### Aufbau



- [1] LED Statusanzeige
- [2] AC IN 100 ... 240V
- [3] DC OUT 24V, 2A, 48W

### LEDs

Auf der Front der Spannungsversorgung befinden sich 3 LEDs, die der Fehlerdiagnose dienen.

Die Verwendung und die jeweiligen Farben finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
OH	rot	Overheat: Leuchtet bei Überhitzung
OL	gelb	Overload: Leuchtet wenn der Summenstrom den maximal zulässigen Summenstrom von ca. 3A übersteigt.
OK	grün	Leuchtet wenn keine Störung vorliegt und die DC24V-Spannungsversorgung sichergestellt ist.



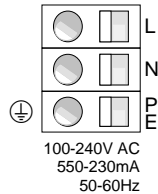
### Hinweis!

Während des Betriebs leuchtet immer nur 1 LED.

Sobald während des Betriebs alle LEDs erlöschen, liegt entweder ein Kurzschluss vor oder das Netzteil ist defekt.

## Anschlussbelegungen

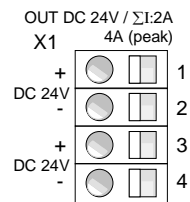
### Eingangsspannung INPUT AC 100...240V



Über die Eingangsbuchse ist die Spannungsversorgung mit Wechselspannung zu versorgen.

Eine Schmelzsicherung schützt den Eingang gegen Überlast.

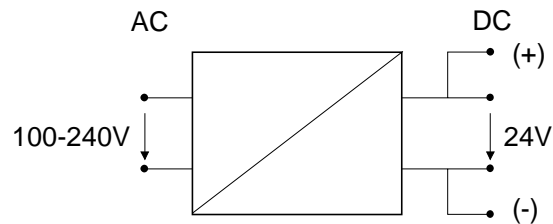
### Ausgangsspannung OUTPUT DC 24V, 2A



Hier können Sie an zwei Anschlüssen System 200V Module anschließen, die extern mit DC 24V zu versorgen sind.

Beide Ausgänge sind kurzschlussicher und haben je eine Ausgangsspannung von DC24V bei einem Summenstrom von max. 2A.

### Prinzipschaltbild



### Gefahr!

- Die Spannungsversorgungen sind vor dem Beginn von Installations- und Instandhaltungsarbeiten unbedingt freizuschalten, d.h. vor Arbeiten an einer Spannungsversorgung oder an der Zuleitung, ist die Spannungszuführung stromlos zu schalten (Stecker ziehen, bei Festanschluss ist die zugehörige Sicherung abzuschalten)!
- Anschluss und Änderungen dürfen nur durch ausgebildetes Elektro-Fachpersonal ausgeführt werden!

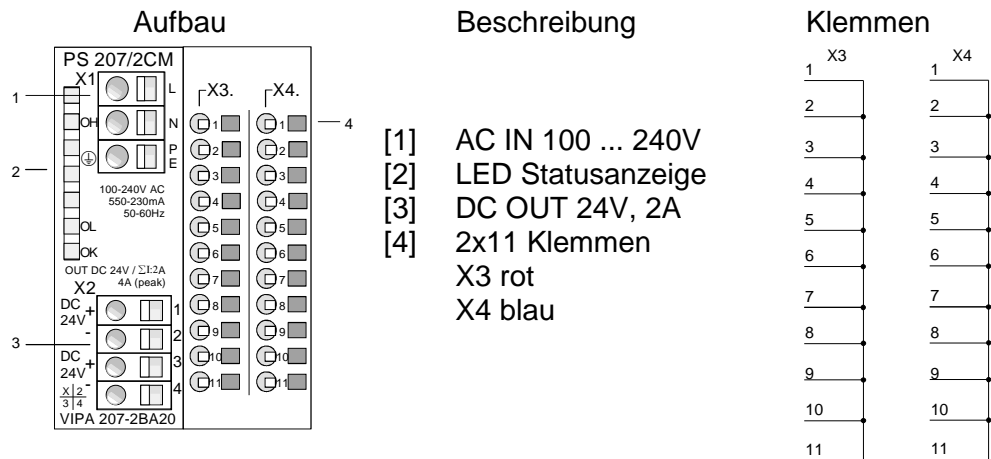
# PS 207/2CM - Spannungsversorgung mit Klemmen - Aufbau

## Eigenschaften

Die Spannungsversorgung zeichnet sich durch folgende Eigenschaften aus:

- Wide-Range-Input AC 100...240V ohne manuelle Umschaltung
- Ausgangsspannung DC 24V, 2A, 48W
- Einsetzbar zusammen mit System 200V auf Tragschiene oder als "stand-alone" Gerät
- Schutz gegen Kurzschluss, Überlast und Leerlauf
- Wirkungsgrad typ 90% bei  $I_{nenn}$
- Klemmenmodul mit 2x11 Klemmen

## Aufbau



## LEDs

Auf der Front der Spannungsversorgung befinden sich 3 LEDs, die der Fehlerdiagnose dienen.

Die Verwendung und die jeweiligen Farben finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
OH	rot	Overheat: Leuchtet bei Überhitzung
OL	gelb	Overload: Leuchtet wenn der Summenstrom den maximal zulässigen Summenstrom von ca. 3A übersteigt.
OK	grün	Leuchtet wenn keine Störung vorliegt und die DC24V-Spannungsversorgung sichergestellt ist.



### Hinweis!

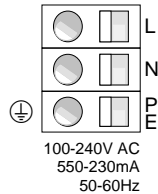
Während des Betriebs leuchtet immer nur 1 LED.

Sobald während des Betriebs alle LEDs erlöschen, liegt entweder ein Kurzschluss vor oder das Netzteil ist defekt.



## Anschlussbelegungen

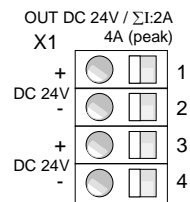
### Eingangsspannung INPUT AC 100...240V



Über die Eingangsbuchse ist die Spannungsversorgung mit Wechselspannung zu versorgen.

Eine Schmelzsicherung schützt den Eingang gegen Überlast.

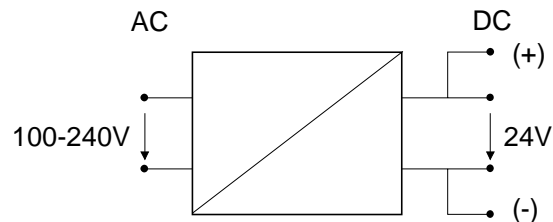
### Ausgangsspannung OUTPUT DC 24V, 2A



Hier können Sie an zwei Anschlüssen System 200V Module anschließen, die extern mit DC 24V zu versorgen sind.

Beide Ausgänge sind kurzschlussicher und haben je eine Ausgangsspannung von DC24V bei einem Summenstrom von max. 2A.

### Prinzipschaltbild



### Gefahr!

- Die Spannungsversorgungen sind vor dem Beginn von Installations- und Instandhaltungsarbeiten unbedingt freizuschalten, d.h. vor Arbeiten an einer Spannungsversorgung oder an der Zuleitung, ist die Spannungszuführung stromlos zu schalten (Stecker ziehen, bei Festanschluss ist die zugehörige Sicherung abzuschalten)!
- Anschluss und Änderungen dürfen nur durch ausgebildetes Elektro-Fachpersonal ausgeführt werden!

## Montage

Für die Spannungsversorgungen gibt es 2 Montagemöglichkeiten:

- Zusammen mit Ihren System 200V Modulen gemeinsam auf eine Tragschiene montiert. Hierbei darf die Spannungsversorgung immer nur am äußeren Rand Ihres System 200V montiert werden, da ansonsten der Rückwandbus unterbrochen ist.  
Die Spannungsversorgungen haben keine Verbindung zum Rückwandbus.
- Als "stand-alone" Spannungsversorgung auf einer Tragschiene montiert.

Bitte beachten sie bei der Auswahl des Montageorts, dass die Spannungsversorgung im Betrieb ausreichend gekühlt wird.



### Gefahr!

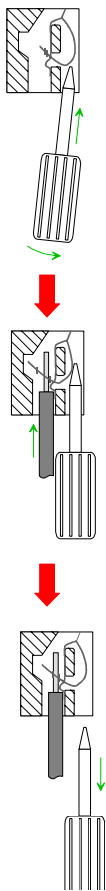
- Die Spannungsversorgungen dürfen ausschließlich in Bereiche eingebaut werden, die nur dem Instandhalter zugänglich sind!
- Die Spannungsversorgungen sind vor dem Beginn von Installations- und Instandhaltungsarbeiten unbedingt freizuschalten, d.h. vor Arbeiten an einer Spannungsversorgung oder an der Zuleitung, ist die Spannungszuführung stromlos zu schalten (Stecker ziehen, bei Festanschluss ist die zugehörige Sicherung abzuschalten)!
- Anschluss und Änderungen dürfen nur durch ausgebildetes Elektro-Fachpersonal ausgeführt werden!
- Bedingt durch die kompakte Bauweise kann zur Gewährleistung einer ausreichenden Kühlung der Berühr- und Brandschutz nicht eingehalten werden. Aus diesem Grund ist der Brandschutz durch die Konstruktion der Umgebung des eingebauten Netzteils sicherzustellen (z.B. Einbau in einen Schaltschrank, der die Brandschutzordnungen erfüllt)!
- Bitte beachten Sie die nationalen Vorschriften und Richtlinien im jeweiligen Verwenderland (Installation, Schutzmaßnahmen, EMV ...).

## Verdrahtung

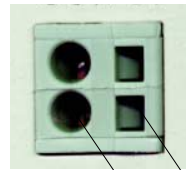
Für die Verdrahtung werden Anschlussklemmen mit Federklemmtechnik eingesetzt.

Sie können Drähte mit einem Querschnitt von  $0,08\text{mm}^2$  bis  $2,5\text{mm}^2$  anschließen. Es können sowohl flexible Litzen ohne Aderendhülse, als auch starre Leiter verwendet werden.

### Anschluss mit Federklemmtechnik



Die Leitungen befestigen Sie wie folgt an den Federklemmkontakten:



Rechteckige Öffnung für Schraubendreher

Runde Öffnung für Drähte

Die nebenstehende 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  $2,5\text{mm}^2$  anschließen.
- Durch Entfernen des Schraubendrehers wird der Draht über einen Federkontakt sicher mit dem Steckverbinder verbunden.



### Gefahr!

- Die Spannungsversorgungen sind vor dem Beginn von Installations- und Instandhaltungsarbeiten unbedingt freizuschalten, d.h. vor Arbeiten an einer Spannungsversorgung oder an der Zuleitung, ist die Spannungszuführung stromlos zu schalten (Stecker ziehen, bei Festanschluss ist die zugehörige Sicherung abzuschalten)!
- Anschluss und Änderungen dürfen nur durch ausgebildetes Elektro-Fachpersonal ausgeführt werden!

## Technische Daten

### Spannungs- versorgung PS 207/2, 2A, 48W

Elektrische Daten	VIPA 207-1BA00
Eingangsnennspannung	AC 100...240V
Frequenz	50 Hz / 60 Hz
Eingangsnennstrom	0,24A / AC 230V
- Einschaltstrom	max.30A
Pufferzeit (bei Netzspannung AC $\geq 150V$ )	min.10 ms
Ausgangsnennspannung	DC 24V $\pm$ 5 %
- Restwelligkeit	< 100 mV <sub>ss</sub>
- Leerlaufspannungsfestigkeit	ja
Ausgangsnennstrom	2A (48W); 4A (peak)
Wirkungsgrad	typ. 90% bei I <sub>nenn</sub>
Verlustleistung	5W bei Nennlast
Leerlaufleistung	1,5W
Statusanzeigen (LED)	über LEDs auf der Frontseite
Betriebsbedingungen	
Temperaturbereich	Betrieb 0°C...55°C (lineares Derating von 40°C bis 55°C mit 1,3W/°C)
Lagerung	- 25°C...+ 85°C
EMV	DIN EN 61000 / Teil4-8
Zulassung/CE	ja
Schutz generell	Kurzschluss; Überlast; Übertemperatur
Montage	IP 20
Anschluss	DIN-Tragschiene Federzugklemme Eingang L, N, PE Ausgang 2xDC24V parallel
Mechanische Daten	
Abmessungen (BxHxT)	25,4x76x76mm
Gewicht	250 g
Bestelldaten	
AC 100V-240V DC 24V / 2A	VIPA 207-1BA00

**Spannungs-  
versorgung  
PS 207/2CM,  
2A, 48W**

Elektrische Daten	VIPA 207-2BA20
Eingangsnennspannung	AC 100...240V
Frequenz	50Hz / 60Hz
Eingangsnennstrom	0,24A / AC 230V
- Einschaltstrom	max.30A
Pufferzeit (bei Netzspannung AC $\geq 150V$ )	min.10ms
Ausgangsnennspannung	DC 24V $\pm$ 5 %
- Restwelligkeit	< 100 mV <sub>ss</sub>
- Leerlaufspannungsfestigkeit	ja
Ausgangsnennstrom	2A (48W); 4A (peak)
Wirkungsgrad	typ. 90% bei I <sub>nenn</sub>
Verlustleistung	5W bei Nennlast
Leerlaufleistung	1,5W
Statusanzeigen (LED)	über LEDs auf der Frontseite
Betriebsbedingungen	
Temperaturbereich	Betrieb 0°C...55°C (lineares Derating von 40°C bis 55°C mit 1,2W/°C)
Lagerung	- 25°C...+ 85°C
EMV	DIN EN 61000 / Teil4-8
Zulassung/CE	ja
Schutz generell	Kurzschluss; Überlast; Übertemperatur
Montage	IP 20
Anschluss	DIN-Tragschiene Federzugklemme Eingang L, N, PE Ausgang 2xDC24V parallel
Klemmenmodul	
Anzahl der Reihen	2
Anzahl der Klemmen je Reihe	11
Maximaler Klemmenstrom	10A
Mechanische Daten	
Abmessungen (BxHxT)	50,8x76x76mm
Gewicht	300 g
Bestelldaten	
AC 100V-240V	VIPA 207-2BA20
DC 24V / 2A	



## Teil 13 Digitale Eingabe-Module

### Überblick

Inhalt dieses Kapitels ist der Aufbau und die Funktionsweise der Digitalen Eingabe-Module von VIPA.

Nachfolgend sind beschrieben:

- Systemübersicht der Digitalen Eingabe-Module
- Eigenschaften
- Aufbau
- Anschluss- und Prinzipschaltbild
- Technische Daten

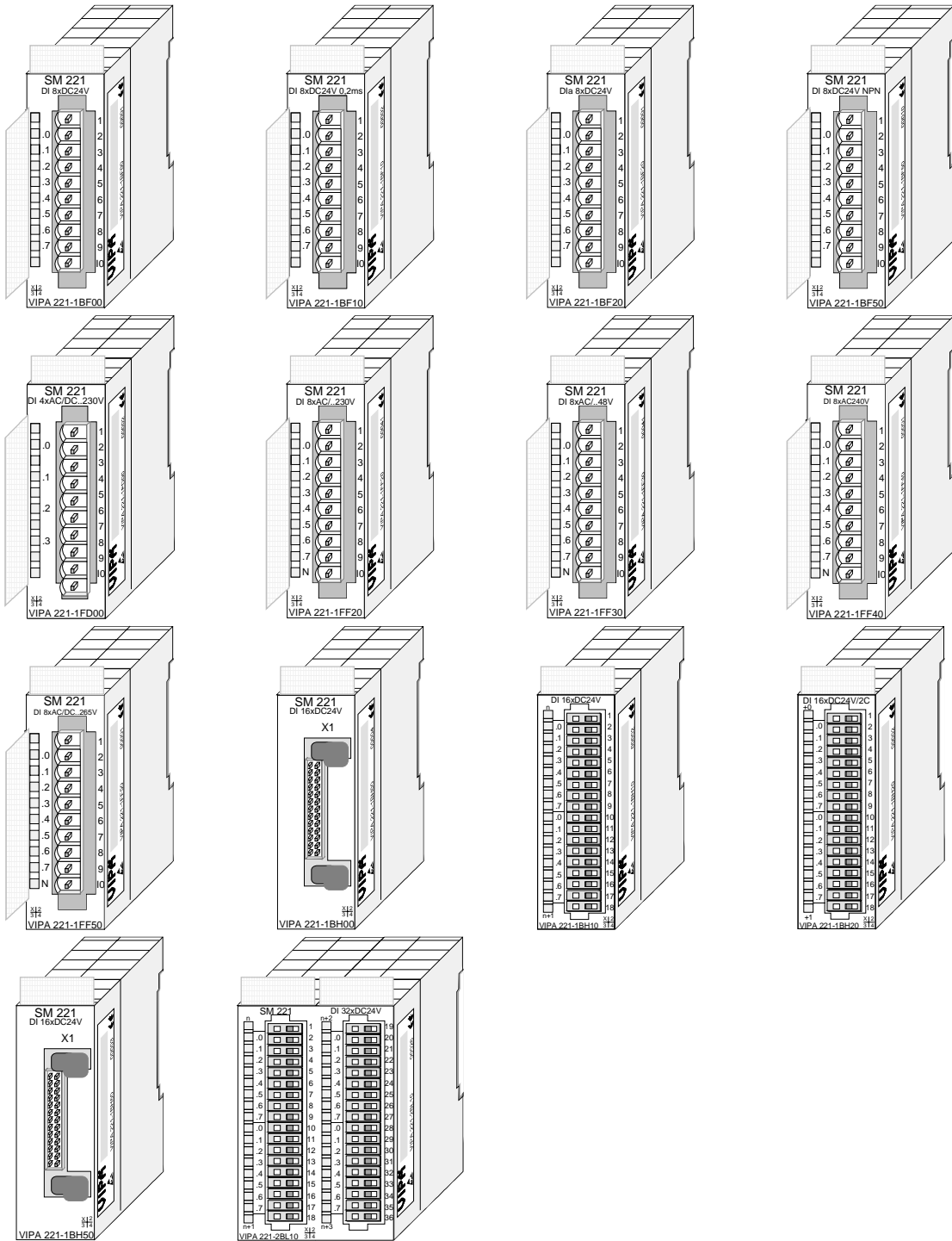
### Inhalt

Thema	Seite
<b>Teil 13 Digitale Eingabe-Module .....</b>	<b>13-1</b>
Systemübersicht .....	13-2
DI 8xDC 24V .....	13-4
DI 8xDC 24V 0,2ms .....	13-6
DIa 8xDC 24V .....	13-8
DI 8xDC 24V NPN .....	13-10
DI 4xAC/DC 90...230V .....	13-12
DI 8xAC/DC 60...230V .....	13-14
DI 8xAC/DC 24...48V .....	13-16
DI 8xAC 240V .....	13-18
DI 8xAC/DC 180...265V .....	13-20
DI 16xDC 24V mit UB4x .....	13-22
DI 16xDC 24V .....	13-24
DI 16xDC 24V/1C .....	13-26
DI 16xDC 24V NPN .....	13-36
DI 32xDC 24V .....	13-38

# Systemübersicht

## Eingabe-Module SM 221

Nachfolgend finden Sie eine Übersicht der bei VIPA zur Zeit erhältlichen Digitalen Eingabe-Module:





**Bestelldaten  
Eingabe-Module**

Typ	Bestellnummer	Seite
DI 8xDC 24V	VIPA 221-1BF00	13-4
DI 8xDC 24V 0,2ms	VIPA 221-1BF10	13-6
DIa 8xDC 24V	VIPA 221-1BF20	13-8
DI 8xDC 24V NPN	VIPA 221-1BF50	13-10
DI 4xAC/DC 90...230V	VIPA 221-1FD00	13-12
DI 8xAC/DC 60...230V	VIPA 221-1FF20	13-14
DI 8xAC/DC 24...48V	VIPA 221-1FF30	13-16
DI 8xAC 240V	VIPA 221-1FF40	13-18
DI 8xAC/DC 180...265V	VIPA 221-1FF50	13-20
DI 16xDC 24V mit UB4x	VIPA 221-1BH00	13-22
DI 16xDC 24V	VIPA 221-1BH10	13-24
DI 16xDC 24V/1C	VIPA 221-1BH20	13-26
DI 16xDC 24V NPN	VIPA 221-1BH50	13-36
DI 32xDC 24V	VIPA 221-2BL10	13-38

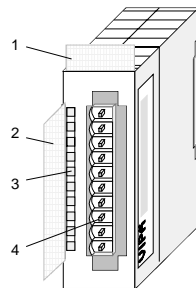
## DI 8xDC 24V

**Bestelldaten** DI 8xDC 24V VIPA 221-1BF00

**Beschreibung** Das digitale Eingabe-Modul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem. Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 8 Eingänge, potenzialgetrennt zum Rückwandbus
  - Eingangs-Nennspannung DC 24V
  - Geeignet für Schalter und Näherungsschalter
  - Statusanzeige der Kanäle durch LED

**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

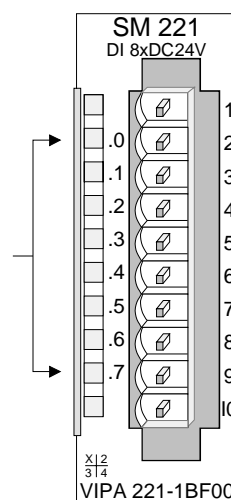
**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

.0... .7 LEDs (grün)  
E.0 bis E.7  
ab ca. 15V wird das Signal "1" erkannt und die entsprechende LED angesteuert

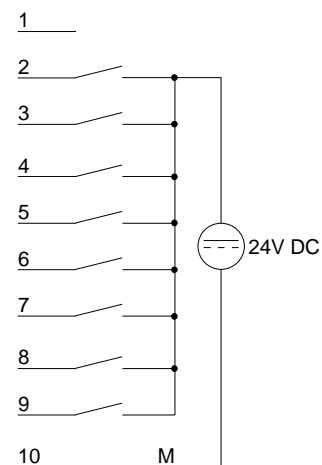
**Pin Belegung**

- 1 nicht belegt
- 2 Eingang E.0
- 3 Eingang E.1
- 4 Eingang E.2
- 5 Eingang E.3
- 6 Eingang E.4
- 7 Eingang E.5
- 8 Eingang E.6
- 9 Eingang E.7
- 10 Masse

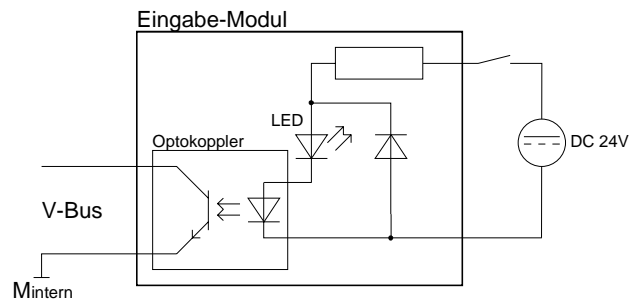


Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 221-1BF00
Anzahl der Eingänge	8
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	0 ... 5V
Signalspannung "1"	15 ... 28,8V
Eingangsfiler Zeitverzögerung	3ms
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	20mA
Potenzialtrennung	500V <sub>eff</sub> (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	1Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

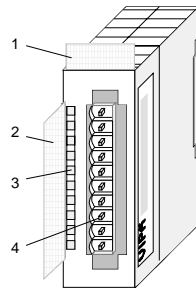
# DI 8xDC 24V 0,2ms

**Bestelldaten** DI 8xDC 24V 0,2ms VIPA 221-1BF10

**Beschreibung** Das digitale Eingabe-Modul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem. Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 8 Eingänge, potenzialgetrennt zum Rückwandbus
  - Verzögerungszeit 0,2ms
  - Eingangs-Nennspannung DC 24V
  - Geeignet für Schalter und Näherungsschalter
  - Statusanzeige der Kanäle durch LED

**Aufbau**

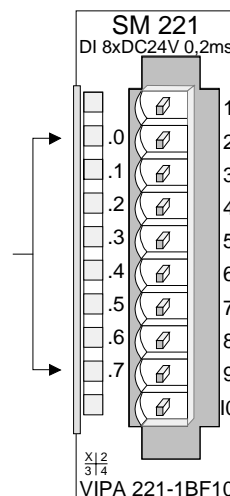


- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

<b>LED</b>	<b>Beschreibung</b>	<b>Pin</b>	<b>Belegung</b>
------------	---------------------	------------	-----------------

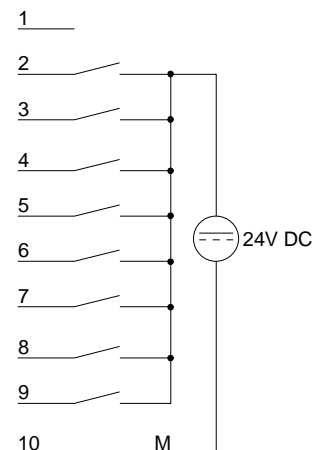
.0... .7	LEDs (grün) E.0 bis E.7 ab ca. 15V wird das Signal "1" erkannt und die entsprechende LED angesteuert		
----------	--	--	--



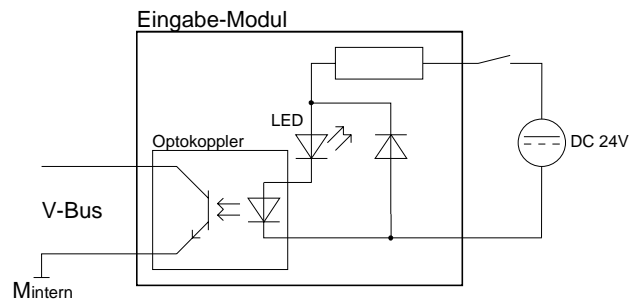
1	nicht belegt	1	Eingang E.0
2	Eingang E.1	2	Eingang E.1
3	Eingang E.2	3	Eingang E.2
4	Eingang E.3	4	Eingang E.3
5	Eingang E.4	5	Eingang E.4
6	Eingang E.5	6	Eingang E.5
7	Eingang E.6	7	Eingang E.6
8	Eingang E.7	8	Eingang E.7
9	Masse	9	
10		10	Masse

Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 221-1BF10
Anzahl der Eingänge	8
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	0 ... 5V
Signalspannung "1"	15 ... 28,8V
Eingangsfiler Zeitverzögerung	0,2ms
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	20mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	1Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

## Dla 8xDC 24V

**Bestelldaten** Dla 8xDC 24V VIPA 221-1BF20

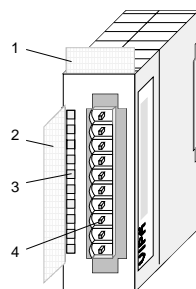
**Beschreibung** Das digitale Alarmeingabe-Modul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem.

Alle Eingänge sind alarmfähig. Mit steigender Flanke des Eingangs wird der Alarm aktiviert. Durch den Alarm wird der OB40 in der CPU aufgerufen. Ist dieser OB nicht vorhanden, wird der OB85 aufgerufen. Ist dieser OB ebenfalls nicht programmiert, so geht die CPU in STOP.

Das Modul hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 8 Alarmeingänge, potenzialgetrennt zum Rückwandbus
  - Eingangs-Nennspannung DC 24V
  - Geeignet für Signale, die schnell ausgewertet werden müssen (Schalter und Näherungsschalter)
  - Statusanzeige der Kanäle durch LED

**Aufbau**

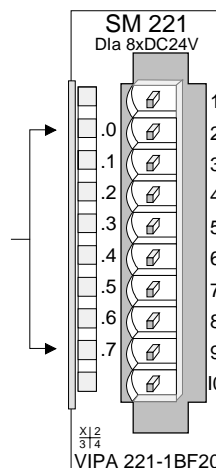


- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

.0... .7 LEDs (grün)  
E.0 bis E.7  
ab ca. 15V wird das Signal "1" erkannt und die entsprechende LED angesteuert

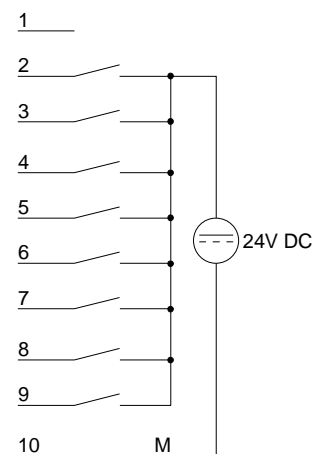


**Pin Belegung**

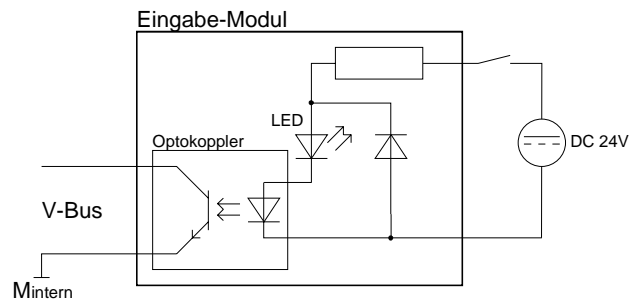
- 1 nicht belegt
- 2 Eingang E.0
- 3 Eingang E.1
- 4 Eingang E.2
- 5 Eingang E.3
- 6 Eingang E.4
- 7 Eingang E.5
- 8 Eingang E.6
- 9 Eingang E.7
- 10 Masse

Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild

**Hinweis!**

Das Modul können Sie ab folgenden CPU-Firmware-Versionen im System 200V einsetzen:

CPU 21x: Version 2.2.1

CPU 24x: Version 3.0.6

Der Einsatz unter älteren Firmware-Versionen führt zu Fehlermeldungen und die CPU geht in STOP!

## Technische Daten

Elektrische Daten	VIPA 221-1BF20
Anzahl der Alarmeingänge	8
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	0 ... 5V
Signalspannung "1"	15 ... 28,8V
Eingangsfiler Zeitverzögerung	3ms
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	140mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	1Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

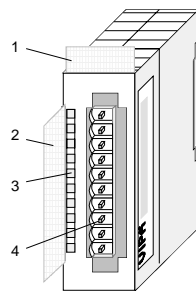
## DI 8xDC 24V NPN

**Bestelldaten** DI 8xDC 24V NPN VIPA 221-1BF50

**Beschreibung** Das digitale Eingabe-Modul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem. Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen. Ein Eingang wird aktiviert sobald er auf Masse geschaltet wird.

- Eigenschaften**
- 8 Eingänge, potenzialgetrennt zum Rückwandbus
  - m-lesend (Signal "1" bei auf Eingang geschalteter Masse)
  - Eingangs-Nennspannung DC 24V
  - Geeignet für Schalter und Näherungsschalter
  - Statusanzeige der Kanäle durch LED

**Aufbau**



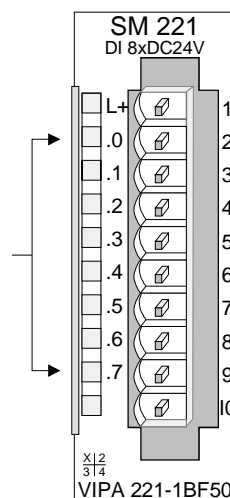
- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

.0... .7 LEDs (grün)  
E.0 bis E.7  
sobald ein Eingang auf Masse liegt wird das Signal "1" erkannt und die entsprechende LED angesteuert

**Pin Belegung**

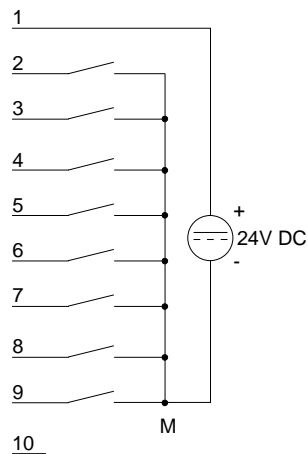


- 1 +DC 24V
- 2 Eingang E.0
- 3 Eingang E.1
- 4 Eingang E.2
- 5 Eingang E.3
- 6 Eingang E.4
- 7 Eingang E.5
- 8 Eingang E.6
- 9 Eingang E.7 / Masse
- 10 reserviert

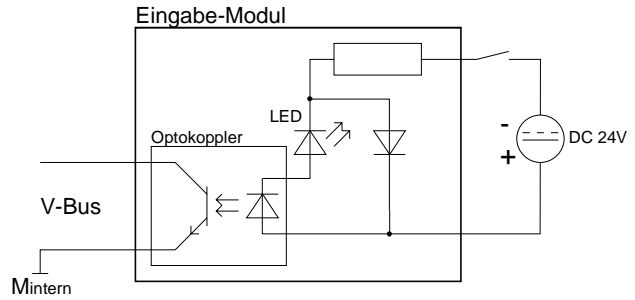


**Anschlussbild und  
Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Technische Daten**

Elektrische Daten	VIPA 221-1BF50
Anzahl der Eingänge	8
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	15 ... 28,8V
Signalspannung "1"	0 ... 5V
Eingangsfilter Zeitverzögerung	3ms
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	20mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	1Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

## DI 4xAC/DC 90...230V

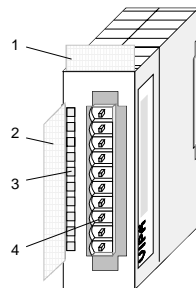
**Bestelldaten** DI 4xAC/DC 90...230V VIPA 221-1FD00

**Beschreibung** Das digitale Eingabe-Modul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem.

Es hat 4 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 4 Eingänge, zum Rückwandbus und untereinander potenzialgetrennt
  - Statusanzeige der Kanäle durch LED
  - Eingangs-Nennspannung AC/DC 90 ... 230V

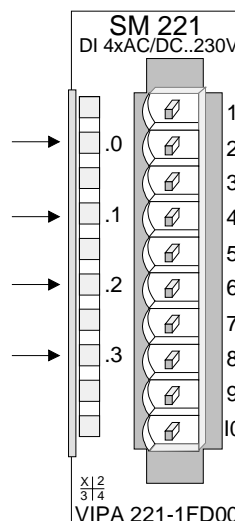
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

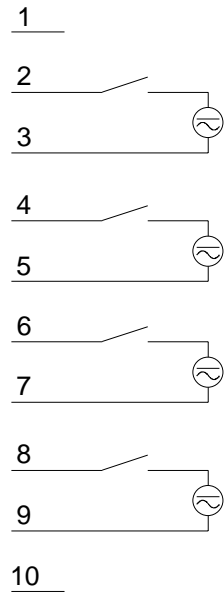
**Statusanzeige  
Steckerbelegung**

LED	Beschreibung	Pin	Belegung
.0	LEDs (grün)	1	nicht belegt
.1	E.0 bis E.3	2	E.0
.2	ab ca. DC 80V bzw. AC 65V (50Hz) wird das Signal "1" erkannt und die entsprechende LED angesteuert	3	Neutralleiter E.0
.3		4	E.1
		5	Neutralleiter E.1
		6	E.2
		7	Neutralleiter E.2
		8	E.3
		9	Neutralleiter E.3
		10	nicht belegt

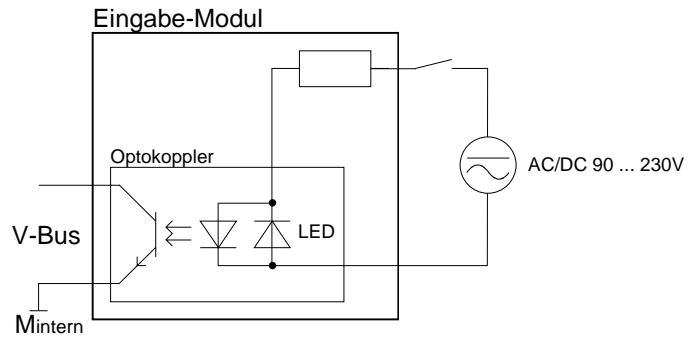


**Anschlussbild und  
Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Technische Daten**

Elektrische Daten	VIPA 221-1FD00
Anzahl der Eingänge	4
Nenneingangsspannung	AC/DC 90 ... 230V
Signalspannung "0"	AC/DC 0 ... 35V
Signalspannung "1"	AC/DC 90 ... 230V
Eingangsfiler Zeitverzögerung	25ms
Frequenz der Eingangsspannung	50 ... 60Hz
Eingangswiderstand	136kΩ
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	80mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	1Byte (Bit 0 ... Bit 3)
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

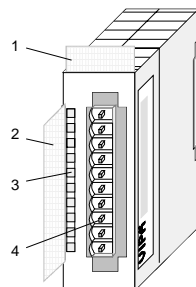
## DI 8xAC/DC 60...230V

**Bestelldaten** DI 8xAC/DC 60...230V VIPA 221-1FF20

**Beschreibung** Das digitale Eingabe-Modul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem.  
Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 8 Eingänge, zum Rückwandbus potenzialgetrennt
  - Eingangsnennspannung AC/DC 60 ... 230V
  - Statusanzeige der Kanäle durch LED

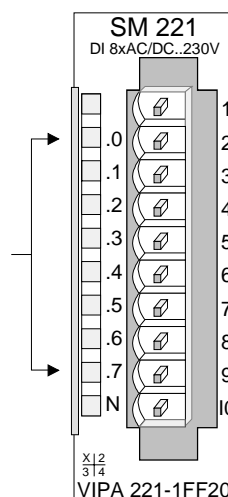
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

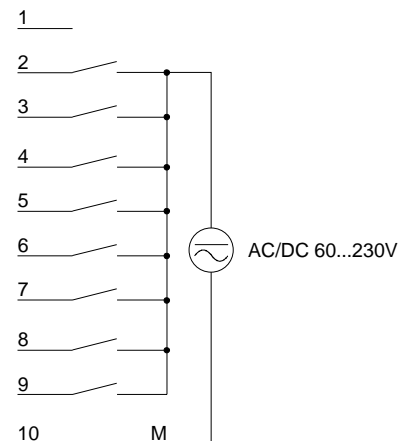
**Statusanzeige  
Steckerbelegung**

LED	Beschreibung	Pin	Belegung
.0... .7	LEDs (grün) E.0 bis E.7 ab ca. DC 55 V bzw. AC 45 V (50Hz) wird das Signal "1" erkannt und die entsprechende LED angesteuert	1	nicht belegt
		2	Eingang E.0
		3	Eingang E.1
		4	Eingang E.2
		5	Eingang E.3
		6	Eingang E.4
		7	Eingang E.5
		8	Eingang E.6
		9	Eingang E.7
		10	Neutralleiter

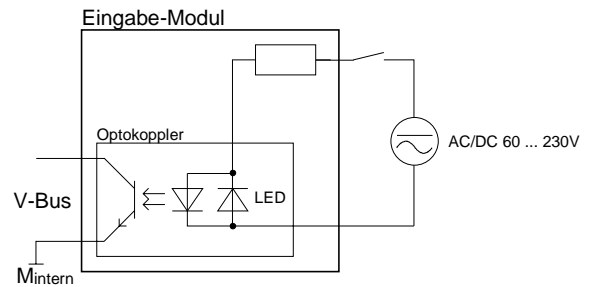


Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 221-1FF20
Anzahl der Eingänge	8
Nenneingangsspannung	AC/DC 60 ... 230V
Signalspannung "0"	AC/DC 0 ... 35V
Signalspannung "1"	AC/DC 60 ... 230V
Eingangsfiler Zeitverzögerung	25ms
Frequenz der Eingangsspannung	50 ... 60Hz
Eingangswiderstand	136kΩ
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	80mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	1Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

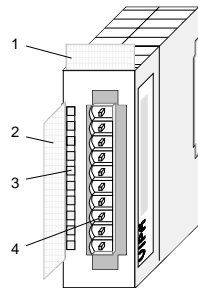
## DI 8xAC/DC 24...48V

**Bestelldaten** DI 8xAC/DC 24...48V VIPA 221-1FF30

**Beschreibung** Das digitale Eingabe-Modul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem.  
Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 8 Eingänge, potenzialgetrennt zum Rückwandbus
  - Eingangsnennspannung AC/DC 24 ... 48V
  - Statusanzeige der Kanäle durch LED

**Aufbau**

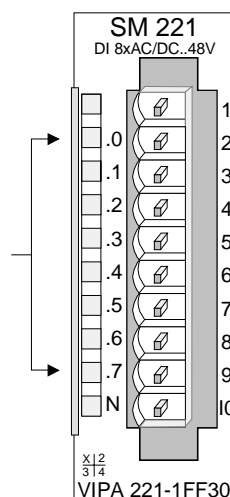


- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

<b>LED</b>	<b>Beschreibung</b>	<b>Pin</b>	<b>Belegung</b>
------------	---------------------	------------	-----------------

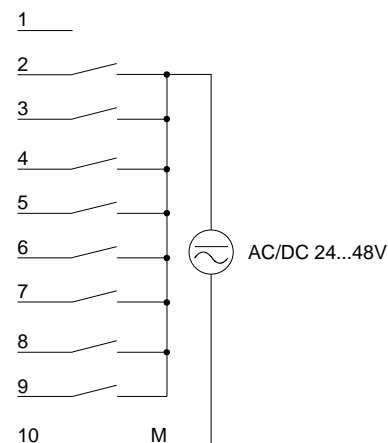
.0... .7	LEDs (grün) E.0 bis E.7 ab ca. DC 14V bzw. AC 12V (50Hz) wird das Signal "1" erkannt und die entsprechende LED angesteuert		
----------	--	--	--



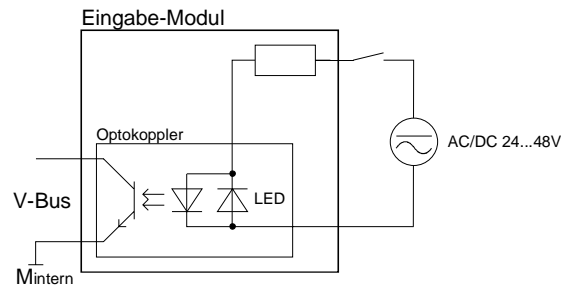
1	nicht belegt	
2	Eingang E.0	
3	Eingang E.1	
4	Eingang E.2	
5	Eingang E.3	
6	Eingang E.4	
7	Eingang E.5	
8	Eingang E.6	
9	Eingang E.7	
10	Neutralleiter	

Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 221-1FF30
Anzahl der Eingänge	8
Nenneingangsspannung	AC/DC 24 ... 48V
Signalspannung "0"	AC/DC 0 ... 8V
Signalspannung "1"	AC/DC 18 ... 48V
Eingangsfiler Zeitverzögerung	25ms
Frequenz der Eingangsspannung	50 ... 60Hz
Eingangswiderstand	16,4kΩ
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	80mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	1Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

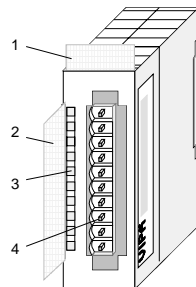
## DI 8xAC 240V

**Bestelldaten** DI 8xAC 240V VIPA 221-1FF40

**Beschreibung** Das digitale Eingabe-Modul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem.  
 Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.  
 In einem definierten Spannungsbereich ändert sich der Signalzustand des jeweiligen Eingangs nicht (Hysterese).

- Eigenschaften**
- 8 Eingänge, zum Rückwandbus potenzialgetrennt
  - Eingangs-Nennspannung AC 240V
  - Statusanzeige der Kanäle durch LED
  - Hysterese
  - Stromaufnahme 20mA je Eingang

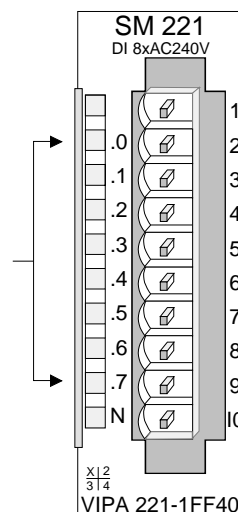
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

LED	Beschreibung
.0... .7	LEDs (grün) E.0 bis E.7 ab ca. AC 190V (50Hz) wird das Signal "1" erkannt und die entsprechende LED angesteuert

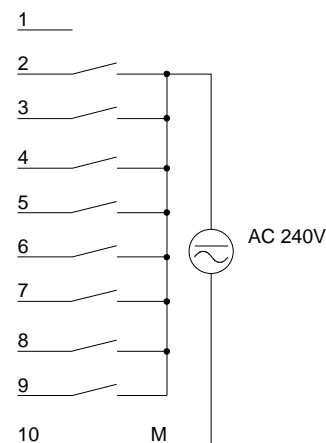


Pin	Belegung
1	nicht belegt
2	Eingang E.0
3	Eingang E.1
4	Eingang E.2
5	Eingang E.3
6	Eingang E.4
7	Eingang E.5
8	Eingang E.6
9	Eingang E.7
10	Neutralleiter

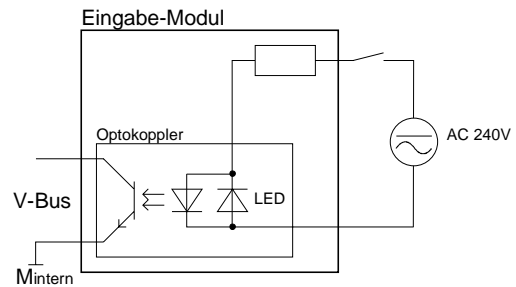


Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild

**Hinweis!**

Diese Baugruppe ist für Spannungen bis max. AC 260V spezifiziert.

Sofern im Netz induktive Lasten auftreten, muss diese Last entweder an der Baugruppe direkt oder an dem entsprechenden Gerät, z.B. durch Löschglieder, ausgefiltert werden.

## Technische Daten

Elektrische Daten	VIPA 221-1FF40
Anzahl der Eingänge	8
Nenneingangsspannung	AC 240V
Stromaufnahme je Eingang	20mA
Signalspannung "0"	AC 0 ... 70V
Hysterese	AC 90 ... 160V
Signalspannung "1"	AC 190 ... 260V
Eingangsfiler Zeitverzögerung	25ms
Frequenz der Eingangsspannung	50 Hz
Eingangswiderstand	136kΩ
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	80mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	1Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	ca. 50g

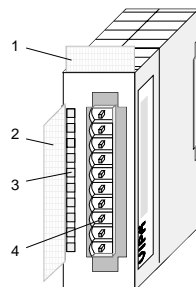
## DI 8xAC/DC 180...265V

**Bestelldaten** DI 8xAC/DC 180...265V VIPA 221-1FF50

**Beschreibung** Das digitale Eingabe-Modul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem.  
Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 8 Eingänge, zum Rückwandbus potenzialgetrennt
  - Eingangs-Nennspannung AC/DC 180...265V
  - Statusanzeige der Kanäle durch LED

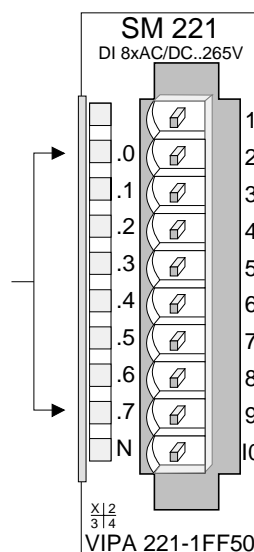
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-Adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

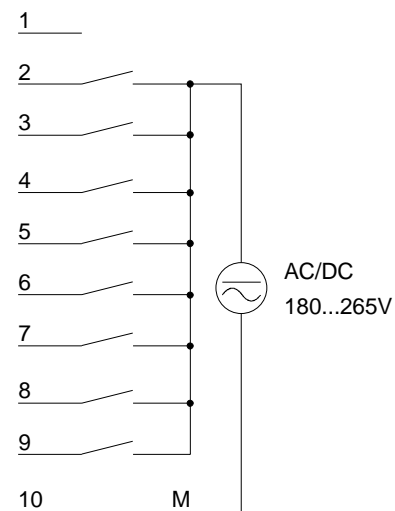
**Statusanzeige  
Steckerbelegung**

LED	Beschreibung	Pin	Belegung
.0... .7	LEDs (grün) E.0 bis E.7 ab ca. DC 150V bzw. AC 170V (50Hz) wird das Signal "1" erkannt und die entsprechende LED angesteuert	1	nicht belegt
		2	Eingang E.0
		3	Eingang E.1
		4	Eingang E.2
		5	Eingang E.3
		6	Eingang E.4
		7	Eingang E.5
		8	Eingang E.6
		9	Eingang E.7
		10	Neutralleiter

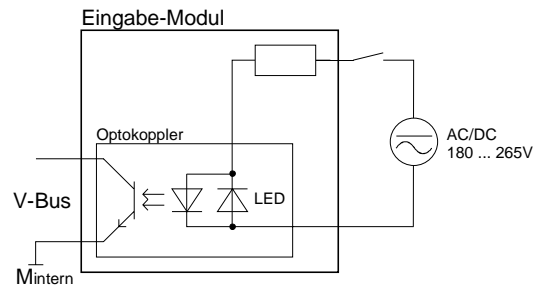


Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 221-1FF50
Anzahl der Eingänge	8
Nenneingangsspannung	AC/DC 180...265V
Signalspannung "0"	AC/DC 0 ...150V
Signalspannung "1"	AC/DC 180 ... 265V
Eingangsfiler Zeitverzögerung	25ms
Frequenz der Eingangsspannung	50 ... 60Hz
Eingangswiderstand	136kΩ
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	30mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	1Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

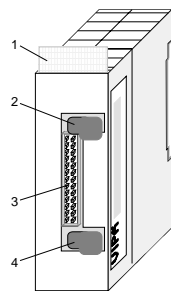
## DI 16xDC 24V mit UB4x

**Bestelldaten** DI 16xDC 24V VIPA 221-1BH00

**Beschreibung** Es erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem. Für den Einsatz des Moduls ist eine UB4x-Umsetzbaugruppe erforderlich. Es hat 16 Kanäle, die ihren Zustand auf der UB4x durch Leuchtdioden anzeigen. Das Modul ist über ein Flachrundkabel (DEA-KB91C) mit einer Umsetzbaugruppe (DEA-UB4x) zu verbinden.

- Eigenschaften**
- 16 Eingänge, potenzialgetrennt zum Rückwandbus
  - Eingangs-Nennspannung DC 24V
  - Geeignet für Schalter und Näherungsschalter
  - Statusanzeige der Kanäle durch LED an Umsetzbaugruppe UB4x

**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Verriegelung
- [3] Wannenstecker zum Anschluss einer Umsetzbaugruppe UB4x über Flachrundkabel
- [4] Verriegelung

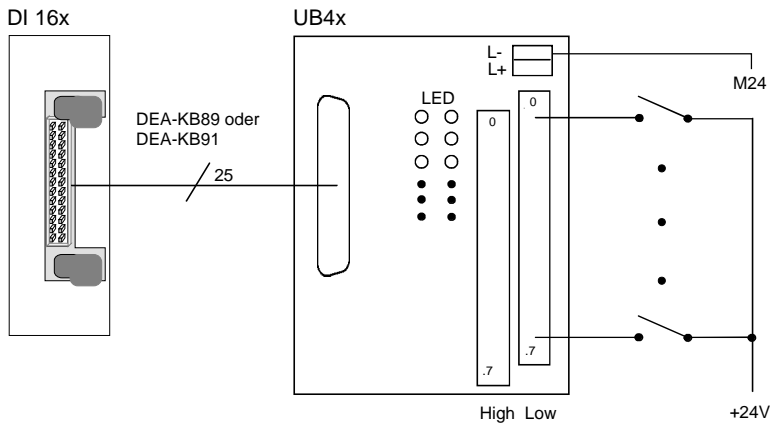
**Statusanzeige auf UB4x**

LED	Beschreibung
0... .15	LEDs (gelb) E.0 bis E.7 High E.0 bis E.7 Low ab ca. 15V wird das Signal "1" erkannt und die entsprechende LED angesteuert
L+ L-	LED (grün) Versorgungsspannung liegt an

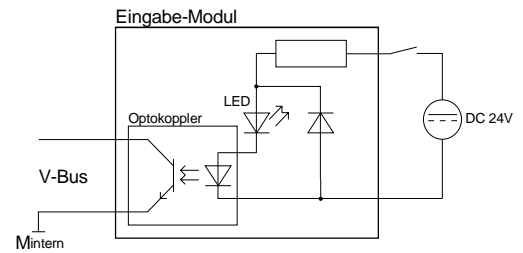
**Steckerbelegung Modul**

Stecker	Pin	Belegung
	23...26	Versorgungsspannung +DC 24V
	22	Eingang E.0
	21	Eingang E.1
	.	.
	.	.
	.	.
	8	Eingang E.14
	7	Eingang E.15
	1...6	Versorgungsspannung Masse

## Anschluss UB4x



## Prinzipschaltbild Modul



## Technische Daten

Elektrische Daten	VIPA 221-1BH00
Anzahl der Eingänge	16
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	0 ... 5V
Signalspannung "1"	15 ... 28,8V
Eingangsfiter Zeitverzögerung	3ms
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	20mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der UB4x
Programmierdaten	
Eingabedaten	2Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

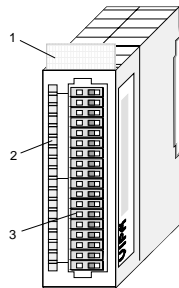
# DI 16xDC 24V

**Bestelldaten** DI 16xDC 24V VIPA 221-1BH10

**Beschreibung** Das digitale Eingabemodul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem. Es hat 16 Kanäle, die ihren Zustand durch LEDs anzeigen.

- Eigenschaften**
- 16 Eingänge, potenzialgetrennt zum Rückwandbus
  - Eingangs-Nennspannung DC 24V
  - Geeignet für Schalter und Näherungsschalter
  - Statusanzeige der Kanäle durch LED

**Aufbau**

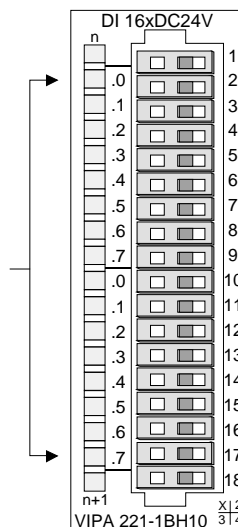


- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] LED-Statusanzeige
- [3] Steckerleiste

**Statusanzeige  
Steckerbelegung**

LED	Beschreibung	Pin	Belegung
-----	--------------	-----	----------

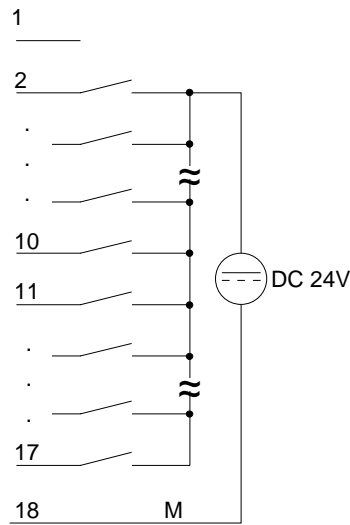
.0 ... .7	LEDs (grün) E.0 bis E.7 (je Byte) ab ca. 15V wird das Signal "1" erkannt und die entsprechende LED angesteuert		
-----------	--	--	--



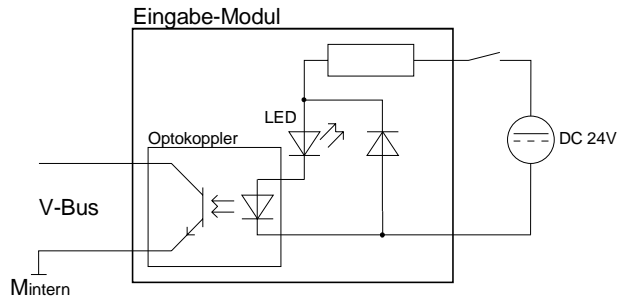
1	nicht belegt
2	Eingang E.0
3	Eingang E.1
4	Eingang E.2
.	.
.	.
.	.
15	Eingang E.13
16	Eingang E.14
17	Eingang E.15
18	Masse

**Anschluss- und  
Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Technische Daten**

Elektrische Daten	VIPA 221-1BH10
Anzahl der Eingänge	16
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	0 ... 5V
Signalspannung "1"	15 ... 28,8V
Eingangsfilter Zeitverzögerung	3ms
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	30mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	2Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

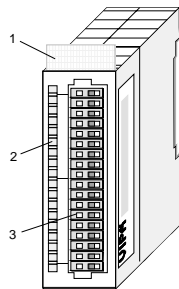
# DI 16xDC 24V/1C

**Bestelldaten** DI 16xDC 24V/1C VIPA 221-1BH20

**Beschreibung** Das digitale Eingabemodul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem. Es hat 16 Kanäle, die ihren Zustand durch LEDs anzeigen. Zusätzlich können die ersten beiden Eingänge einen Zähler ansteuern.

- Eigenschaften**
- 16 Eingänge, potenzialgetrennt zum Rückwandbus
  - Projektierbare Zählerfunktion (Impuls, Frequenz) für die ersten 2 Eingänge
  - Geeignet für Schalter und Näherungsschalter
  - Statusanzeige der Kanäle durch LED

**Aufbau**

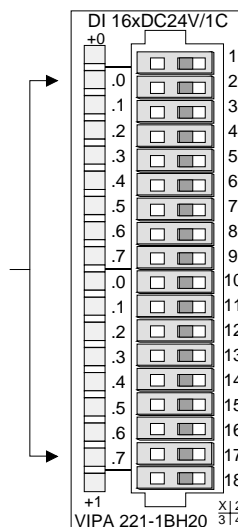


- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] LED-Statusanzeige
- [3] Steckerleiste

**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

.0 ... .7 LEDs (grün)  
E.0 bis E.7 (je Byte)  
ab ca. 15V wird das Signal "1" erkannt und die entsprechende LED angesteuert



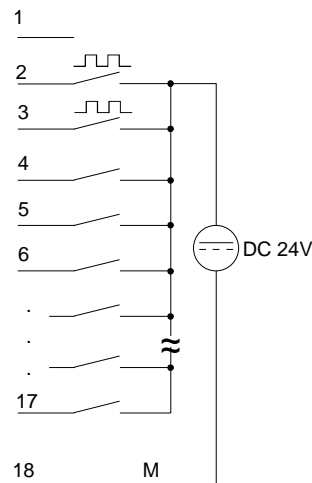
**Pin Belegung**

- 1 nicht belegt
- 2 Eingang E.0/Zähler (A)
- 3 Eingang E.1/Zähler (B)
- 4 Eingang E.2
- . . .
- . . .
- . . .
- 15 Eingang E.13
- 16 Eingang E.14
- 17 Eingang E.15
- 18 Masse

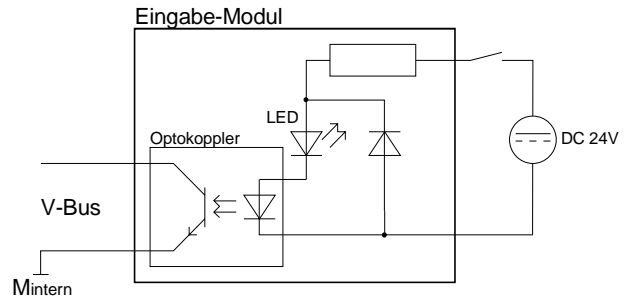


**Anschluss- und Prinzipschaltbild**

**Anschlussbild**



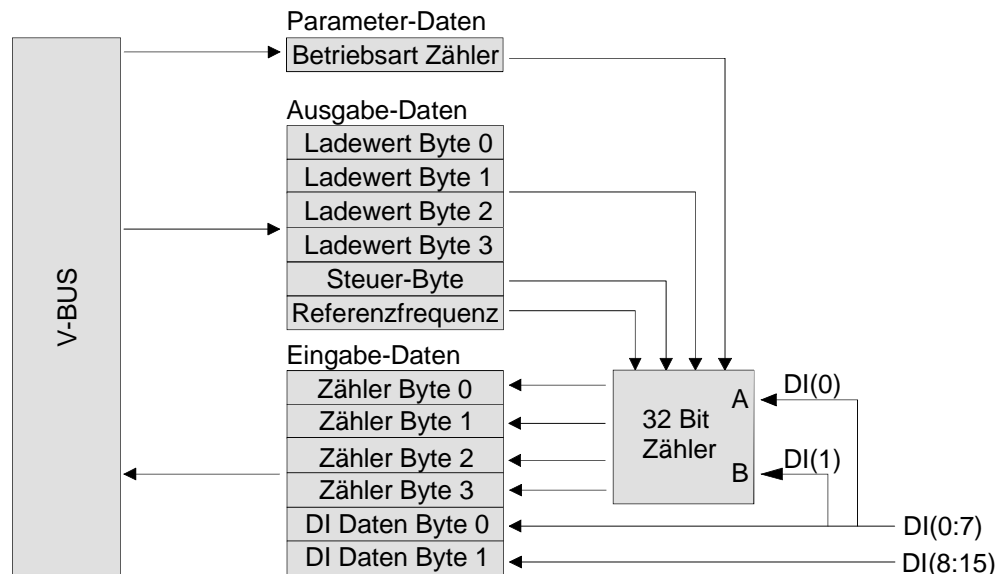
**Prinzipschaltbild**



**Übersicht Modul-funktionen**

Das Modul ist ein digitales 16Bit Eingabemodul für System 200V kombiniert mit einem 1-Kanal 32Bit Zähler.

Die Eingänge DI [0] und DI [1] werden gleichzeitig als "normale" Prozesseingänge und als Zählereingänge genutzt (Signal A und Signal B).



Über das Schreiben der Ausgabedaten DO können Sie den Zählerstand mit einem Wert vorbesetzen und eine Referenzfrequenz vorgeben. Die Aktivierung dieser Werte erfolgt mit dem Steuer-Byte.

Über 1Byte Parameterdaten können Sie die Zähler-Betriebsart vorgeben. Es werden 5 Zählermodi unterstützt. Durch Lesezugriff auf die entsprechenden Bytes der Eingabedaten DI können Sie den Zählerstand ausgeben.

Den Zählvorgang starten bzw. stoppen Sie über das Steuer-Byte (SW-Gate).

**Zählbereich / Grenzwerte**

Das Zähler-Modul kann auf- und abwärts zählen. Der Zählerwert ist 32Bit breit und vom Typ Integer ohne Vorzeichen. Daher sind die Zählergrenzen folgende:

Unterer Grenzwert Zähler	Oberer Grenzwert Zähler
0	+ 4.294.967.295 ( $2^{32} - 1$ )

**Ladewert**

Dem Zähler können Sie einen Ladewert vorgeben. Wenn ein Ladewert zugewiesen wurde, startet der Zähler mit dem Auf- bzw. Abwärtszählen von diesem Wert ab bis zum oberen bzw. unteren Grenzwert und springt nach Erhalt eines neuen Zählimpulses zum unteren (Aufwärtszählen) bzw. oberen Grenzwert (Abwärtszählen), um erneut mit dem Zählen zu beginnen.

Im der Betriebsart "Frequenzmessung" wird der Ladewert zur Vorgabe des Zeitfensters genutzt.

Die Aktivierung eines Ladewerts im Zähler erfolgt durch Setzen von LOAD im Steuer-Byte.

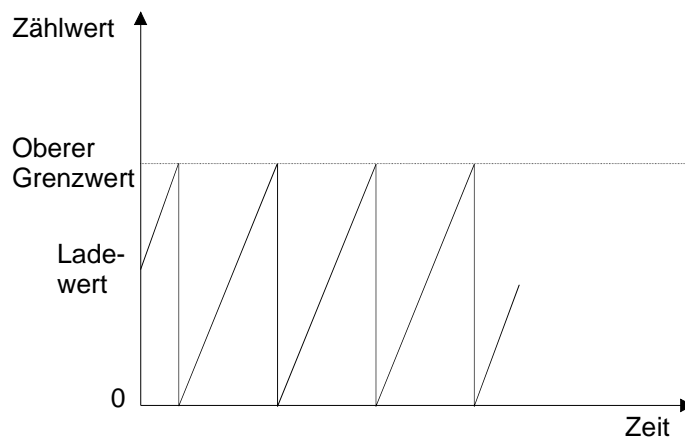
**Fortlaufende Zählfunktion**

In allen Zählermodi wird eine fortlaufende Zählfunktion genutzt, wie sie im Folgenden beschrieben wird.

Wenn der Zähler beim Aufwärtszählen den oberen Grenzwert erreicht und dann einen neuen Zählimpuls erhält, springt der Zähler zum unteren Grenzwert und beginnt erneut zu zählen.

Wenn der Zähler beim Abwärtszählen den unteren Grenzwert erreicht und dann einen neuen Zählimpuls erhält, springt der Zähler zum oberen Grenzwert zurück und beginnt erneut abwärts zu zählen.

Der Zählbereich ist in allen Modi von 0 bis + 4.294.967.295 und kann nicht geändert werden. Der Zähler beginnt mit dem Zählen bei 0 nach einem kompletten Neustart oder durch das Setzen des CLEAR-Bit im Steuer-Byte.



**Zähleraktivierung  
über Software-  
Gate**

Viele Applikationen basieren darauf, dass der Zähler zu einer bestimmten Zeit abhängig von anderen Ereignissen gestartet oder gestoppt werden kann. Diese Start- und Stop-Funktion des Zählprozesses ist über ein Software-Gate geregelt. Wenn das Gate geöffnet ist, kann der Zähler von Zählimpulsen erreicht werden und beginnt mit dem Zählen, ist das Gate geschlossen, können Zählimpulse nicht an den Zähler weitergegeben werden und der Zähler stoppt.

Die Ansteuerung des Software-Gate erfolgt über die Bits START und STOP des Steuer-Byte. Das Setzen von START öffnet das Software-Gate, das Setzen von STOP schließt das Gate.

**Modul-Zugriff**

Das Modul belegt für Ein- und Ausgabedaten je 6Byte im Adressbereich. Zur Vorgabe der Zählerbetriebsart stehen 1Byte Parametrierdaten zur Verfügung.

Das Laden des Zählers bzw. die Vorgabe einer Referenzfrequenz erfolgt über ein Steuer-Byte, indem Sie im Ausgangs-Adressbereich den gewünschten Wert eintragen und durch Setzen von Bit 2 des Steuer-Bytes den Wert im Zähler aktivieren.

Den Zählerstand und die Zustände der Eingänge finden Sie im Eingangs-Adressbereich. Auch während des Zählbetriebs können alle Eingabekanäle abgefragt werden.

**Eingangsdaten  
DI-Datenbytes**

Das Modul belegt im Adressbereich 6Byte Eingangsdaten, die über direkten Lesezugriff erreichbar sind. Die Eingangsbytes 0 bis 3 sind dem 32Bit breiten Zählerwert zugewiesen, die Bytes 4 und 5 den 16Bit der digitalen Eingänge.

Byte	Bit 7 ... 0
0	Zählerwert Byte 0
1	Zählerwert Byte 1
2	Zählerwert Byte 2
3	Zählerwert Byte 3
4	Daten DI Byte 0 (E.7 ... E.0)
5	Daten DI Byte 1 (E.15 ... E.8)

**Ausgangsdaten  
DO-Datenbytes**

Das Modul hat 6Byte Ausgabedaten.

Byte 0 bis 3 sind entsprechend dem gewählten Zählermodus einem Ladewert zugeordnet. Byte 4 dient als Steuer-Byte für den Zähler.

Über Byte 5 geben Sie eine Referenzfrequenz für die Zählermodi "Frequenzmessung" und "Periodenmessung" vor.

Byte	Bit 7 ... 0
0	Ladewert Byte 0
1	Ladewert Byte 1
2	Ladewert Byte 2
3	Ladewert Byte 3
4	Steuer-Byte
5	Referenzfrequenz

*Steuer-Byte:*

Bit	Funktion
0	1 = START Zähler (Software-Gate ist offen)
1	1 = STOP Zähler (Software-Gate ist geschlossen)
2	1 = LOAD Zähler
3	1 = CLEAR Zähler
4	
...	reserviert
7	

*Anwahl Referenzfrequenz:*

Wert	Referenzfrequenz
00h	16 MHz
01h	8 MHz
02h	4 MHz
03h	1 MHz
04h	100 kHz
05h	10 kHz
06h	1 kHz
07h	100 Hz
andere Werte	nicht erlaubt

**Parameterdaten** Das Modul hat 1Byte Parameterdaten zur Auswahl der Betriebsart des Zählers.

Byte	Bit 7 ... 0
0	Betriebsart Zähler

*Anwahl Betriebsart Zähler:*

Wert	Betriebsart Zähler
00h	Vierfach-Impulsauswertung
01h	Impuls- und Richtungsauswertung
02h	Clock Up / Clock Down Auswertung
03h	Frequenzmessung
04h	Periodenmessung
andere	nicht erlaubt

## Zählermodi

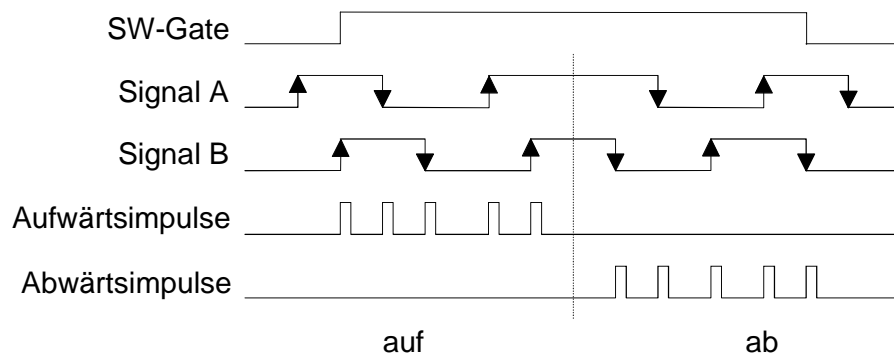
### Vierfach-Impulsauswertung (Modus 00h)

Vierfach-Impulsauswertung bedeutet, dass die steigenden und fallenden Flanken von A und B ausgewertet werden. Je nachdem welcher Kanal vor-eilt, wird der Zähler nach oben bzw. nach unten gezählt.

In dieser Betriebsart haben E.0 und E.1 folgende Belegung und Funktion:

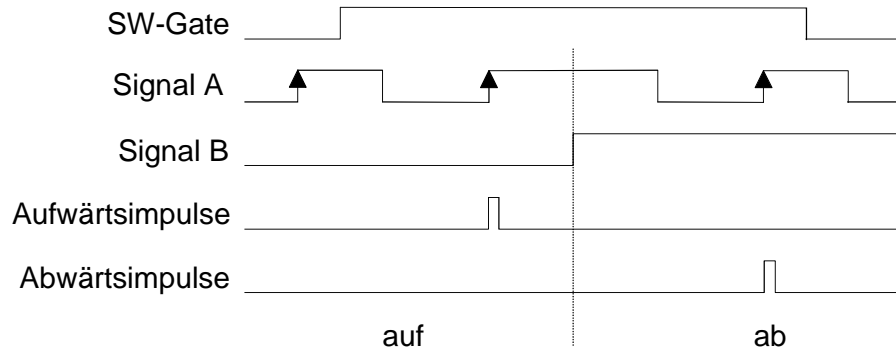
E.0 als Kanal A: Eilt Kanal A vor, wird der Zähler nach oben gezählt.

E.1 als Kanal B: Eilt Kanal B vor, wird der Zähler nach unten gezählt.



**Impuls- und Richtungs- auswertung (Modus 01h)**

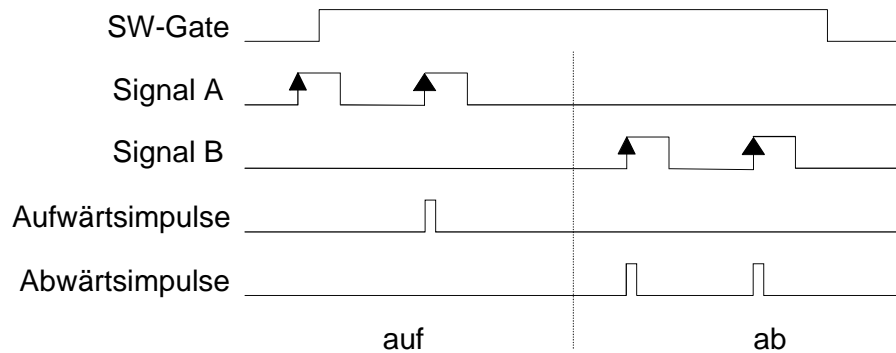
Bei der Impuls- und Richtungs- auswertung wird jede steigende Flanke von Kanal A ausgewertet. Die Zählrichtung bestimmen Sie über Kanal B. In dieser Betriebsart haben E.0 und E.1 folgende Belegung und Funktion:  
 E.0 als Kanal A: Clock-Impuls für der Zähler bei steigender Flanke  
 E.1 als Kanal B: Gibt die Zählrichtung vor (0 = aufwärts, 1 = abwärts)



**Clock Up / Clock Down Auswertung (Modus 02h)**

In diesem Modus werden die steigenden Flanken von Kanal A und B ausgewertet. Der Zähler zählt mit jeder steigenden Flanke A nach oben und mit jeder steigenden Flanke B nach unten.

E.0 als Kanal A: Clock-up-Impuls für den Zähler bei steigender Flanke  
 E.1 als Kanal B: Clock-down-Impuls für den Zähler bei steigender Flanke



**Frequenzmessung (Modus 03h)**

Im Modus zur Frequenzmessung zählt der Zähler die Anzahl der steigenden Flanken von Kanal A innerhalb eines spezifizierten Zeitfensters. Kanal B wird nicht benutzt.

Das Zeitfenster  $T_w$  bestimmen Sie indirekt, indem Sie eine Referenzfrequenz in DO Byte 5 und einen Ladewert in DO Byte 0 ... 3 vorgeben. Es gilt folgende Formel:

$$T_w = \frac{1}{\text{Referenzfrequenz}} * \text{Ladewert}$$

Durch Setzen von Bit 2 des "Steuer"-Bytes wird das Zeitfenster übertragen. Sobald Sie das Software-Gate aktivieren, startet der Referenzzähler mit der ersten steigenden Flanke von Kanal A und wird mit jeder steigenden Flanke der Referenzzuhr erhöht.

Wenn der Referenzzähler den Ladewert erreicht (Zeit  $T_w$  ist abgelaufen), wird der aktuelle Zählerwert nach DI Byte 0 ... 3 kopiert und kann entsprechend ausgelesen werden.

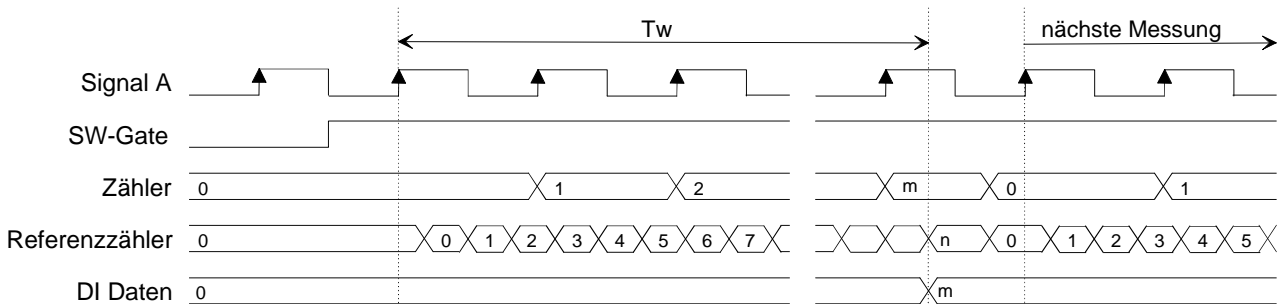
Anschließend werden Zähler und Referenzzähler automatisch zurückgesetzt und die nächste Frequenzmessung startet mit der nächsten steigenden Flanke von Kanal A. Treten nicht mindestens zwei steigende Flanken von Kanal A innerhalb des Zeitfensters  $T_w$  auf, wird der Zählerwert für diese Messung als 0 interpretiert.

Die Frequenzmessung wird über das Software Gate gestartet und gestoppt. Solange das Software Gate geöffnet ist, wird die Frequenz von Kanal A gemessen.

Der Zähler kann jederzeit über CLEAR=1 im "Steuer"-Byte gelöscht werden, während der Ladewert gültig bleibt, bis ein neuer Wert geladen wird oder ein Reset erfolgt.

Die aktuelle Frequenz können Sie über die folgende Formel ermitteln:

$$\text{Frequenz} = \text{Referenzfrequenz} * \frac{\text{Zählerwert}}{\text{Ladewert}}$$



**Beispiel:** Referenzfrequenz : 1 MHz  
 Ladewert (n) : 1.000.000  
 Zählwert (m) : 10.000

$$\text{Frequenz} = 1 \text{ MHz} * \frac{10.000}{1.000.000} = 10 \text{ kHz}$$

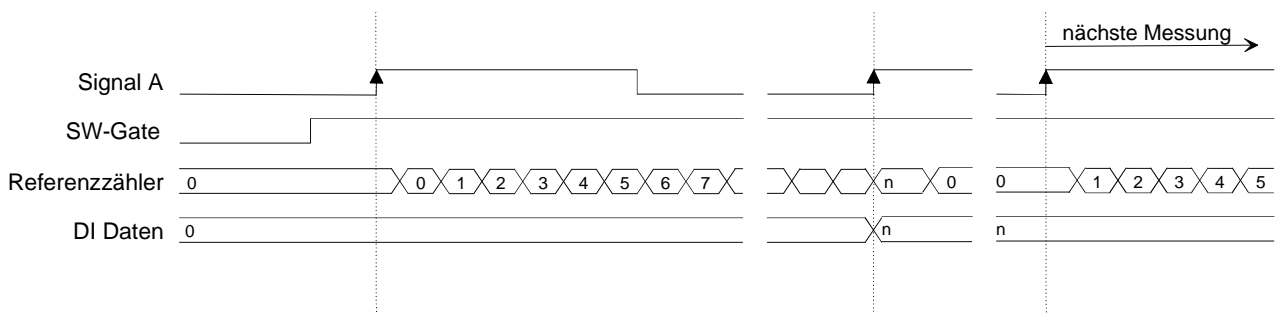
**Periodenmessung (Modus 04h)**

Bei sehr niedrigen Frequenzen ist es sinnvoll anstelle der Frequenz die Periode zu messen. In der Betriebsart "Periodenmessung" wird die Zeit zwischen zwei steigenden Flanken von Kanal A gemessen, indem die Anzahl der steigenden Flanken der gewählten Referenzuhr zwischen den zwei steigenden Flanken von Kanal A gemessen wird. Auch hier wird Kanal B nicht genutzt.

Die Periodenmessung wird mit Hilfe des Software-Gate gestartet und gestoppt. Solange das Software Gate geöffnet ist, wird die Periode von Kanal A kontinuierlich gemessen. Der Zähler kann jederzeit über CLEAR=1 im "Steuer"-Byte gelöscht werden. Anschließend wird die Periodenmessung mit der nächsten steigenden Flanke von Kanal A erneut gestartet.

Die aktuelle Signalperiode können Sie über folgende Formel ermitteln:

$$Periode = \frac{1}{\text{Referenzfrequenz}} * \text{Zählerwert}$$



**Beispiel:** Referenzfrequenz : 1 MHz  
 Zählwert (n) : 10.000

$$Periode = \frac{1}{1\text{ MHz}} * 10.000 = 10\text{ ms}$$



**Hinweis!**

Der Zählerwert bleibt solange gültig, bis die nächste Messung abgeschlossen ist bzw. der Zähler zurückgesetzt wird.

Sollte die nachfolgende Messung nicht abgeschlossen werden z.B. weil keine zweite steigende Flanke von Signal A auftritt, erhalten Sie den alten Zählerwert und nicht den Inhalt des Referenzzählers.



**Technische Daten**

Elektrische Daten	VIPA 221-1BH20
Anzahl der Eingänge	16
Zähler	1 (2 Eingänge A, B)
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	0 ... 5V
Signalspannung "1"	15 ... 28,8V
Eingangsfiler Zeitverzögerung	3ms
Eingangsfiler Impulseingang	100µs
Maximalfrequenz	100kHz
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	100mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	6 Byte
Ausgabedaten	6 Byte
Parameterdaten	1 Byte
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

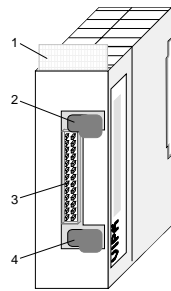
## DI 16xDC 24V NPN

**Bestelldaten** DI 16xDC 24V NPN VIPA 221-1BH50

**Beschreibung** Das Modul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem. Ein Eingang wird aktiviert sobald er auf Masse geschaltet wird.

- Eigenschaften**
- 16 Eingänge, potenzialgetrennt zum Rückwandbus
  - m-lesend (Signal "1" bei auf Eingang geschalteter Masse)
  - Eingangs-Nennspannung DC 24V
  - Geeignet für Schalter und Näherungsschalter

**Aufbau**



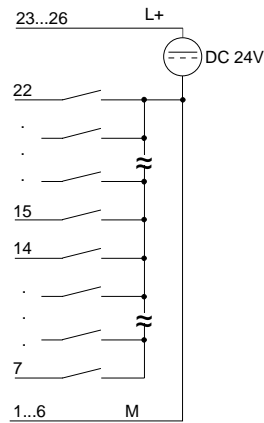
- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Verriegelung
- [3] Wannenstecker zum Anschluss der Eingabe
- [4] Verriegelung

**Steckerbelegung Wannenstecker**

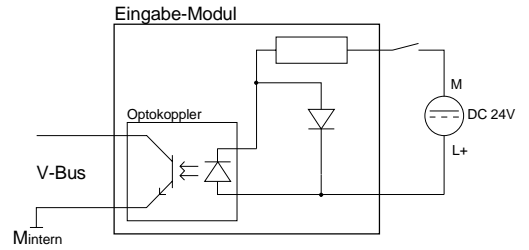
Stecker	Pin	Belegung
26	23...26	Versorgungsspannung +DC 24V
	22	Eingang E.0
	21	Eingang E.1
	.	.
	.	.
	8	Eingang E.14
	7	Eingang E.15
4	1...6	Versorgungsspannung Masse
2		

**Anschluss- und  
Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Technische Daten**

Elektrische Daten	VIPA 221-1BH50
Anzahl der Eingänge	16
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	15 ... 28,8V
Signalspannung "1"	0 ... 5V
Eingangsfilter Zeitverzögerung	3ms
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	20mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	-
Programmierdaten	
Eingabedaten	2Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g

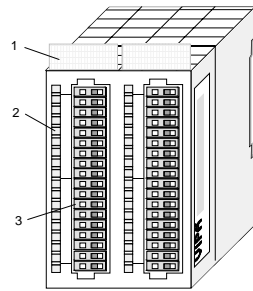
# DI 32xDC 24V

**Bestelldaten** DI 32xDC 24V VIPA 221-2BL10

**Beschreibung** Das digitale Eingangsmodul erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Bussystem. Es hat 32 Kanäle, die ihren Zustand durch LEDs anzeigen.

- Eigenschaften**
- 32 Eingänge, potenzialgetrennt zum Rückwandbus
  - Eingangs-Nennspannung DC 24V
  - Geeignet für Schalter und Näherungsschalter
  - Statusanzeige der Kanäle durch LED

**Aufbau**

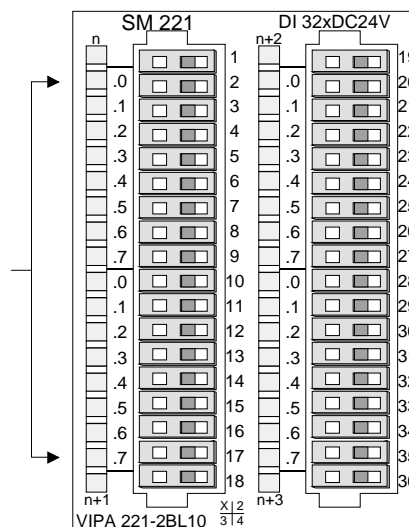


- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] LED-Statusanzeige
- [3] Steckerleiste

**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

.0 ... .7 LEDs (grün)  
E.0 bis E.7 (je Byte)  
ab ca. 15V wird das  
Signal "1" erkannt  
und die  
entsprechende LED  
angesteuert

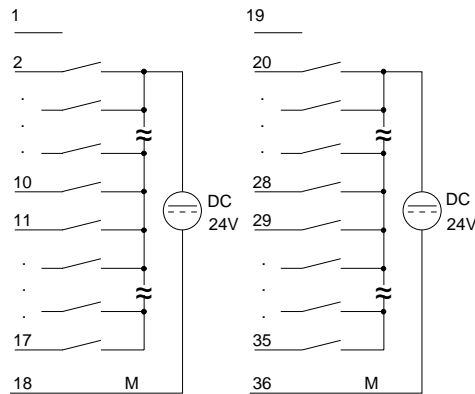


**Pin Belegung**

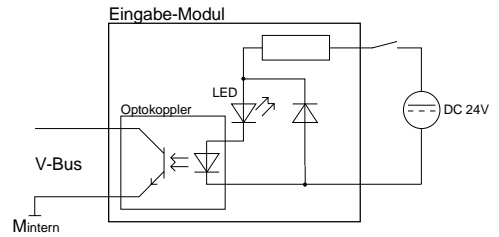
1	nicht belegt
2...17	Eingang E.0...E.15
.	.
.	.
.	.
18	Masse
19	nicht belegt
.	.
.	.
.	.
20 ... 35	Eingang E.16...E.31
36	Masse

**Anschluss- und  
Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Technische Daten**

Elektrische Daten	VIPA 221-2BL10
Anzahl der Eingänge	32
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	0 ... 5V
Signalspannung "1"	15 ... 28,8V
Eingangsfiler Zeitverzögerung	3ms
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	50mA
Potenzialtrennung	in 2 Gruppen zu je 16 Eingängen 500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	4Byte
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8 x 76 x 76
Gewicht	50g



## Teil 14 Digitale Ausgabe-Module

### Überblick

Inhalt dieses Kapitels ist der Aufbau und die Funktionsweise der Digitalen Ausgabe-Module von VIPA.

Nachfolgend sind beschrieben:

- Systemübersicht der Digitalen Ausgabe-Module
- Eigenschaften
- Aufbau
- Anschluss- und Prinzipschaltbilder
- Technische Daten

### Inhalt

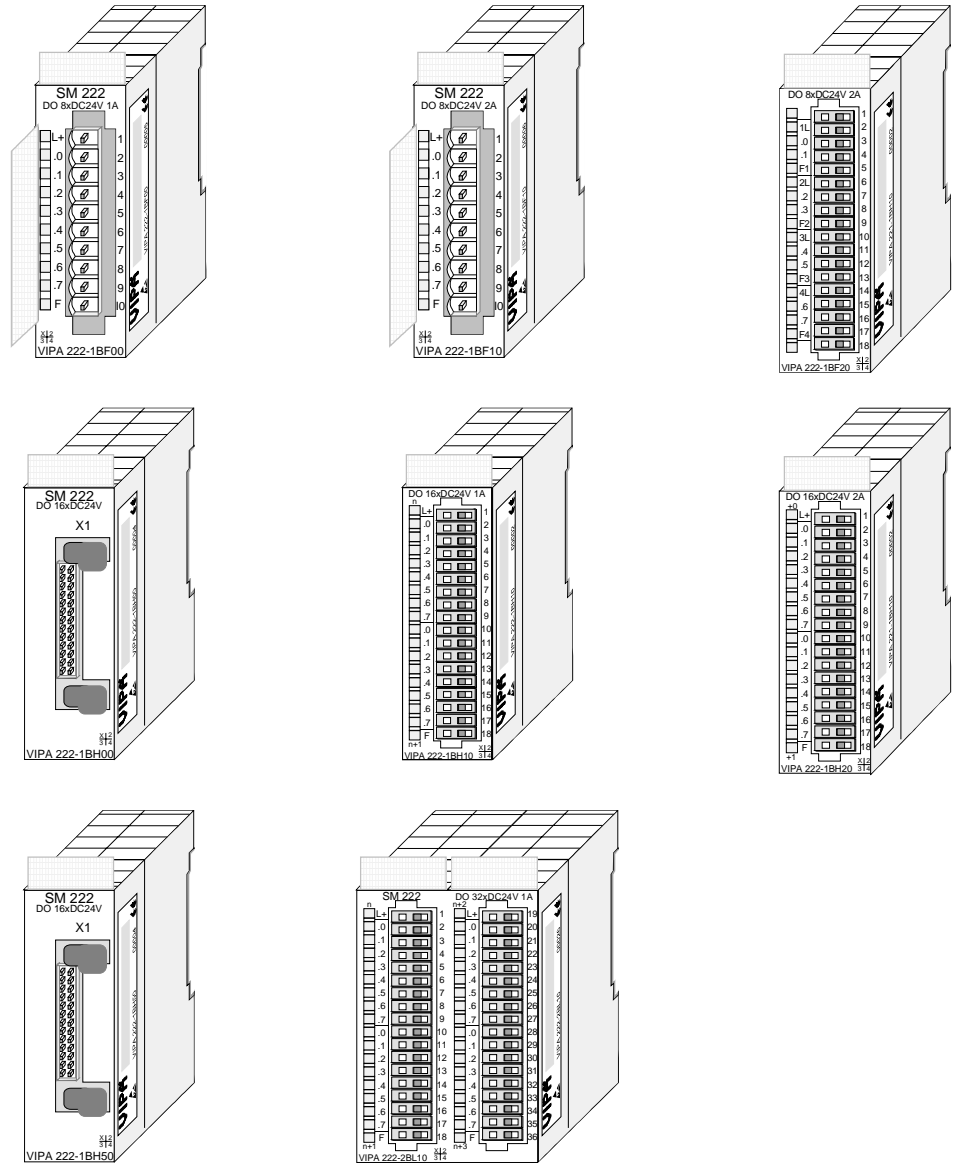
Thema	Seite
<b>Teil 14 Digitale Ausgabe-Module .....</b>	<b>14-1</b>
Systemübersicht .....	14-2
DO 8xDC 24V 1A.....	14-4
DO 8xDC 24V 2A.....	14-6
DO 8xDC 24V 2A potenzialgetrennt 4 á 2.....	14-8
DO 16xDC 24V 0,5A mit UB4x .....	14-10
DO 16xDC 24V 1A.....	14-12
DO 16xDC 24V 2A.....	14-14
DO 16xDC 24V 0,5A NPN .....	14-16
DO 32xDC 24V 1A.....	14-18
DO 8xRelais COM .....	14-20
DO 4xRelais.....	14-22
DO 4xRelais bistabil.....	14-24
DO 8xSolid State COM.....	14-26
DO 4xSolid State .....	14-28

# Systemübersicht

## Ausgabe-Module SM 222

Nachfolgend finden Sie eine Übersicht der bei VIPA zur Zeit erhältlichen Digitalen Ausgabe-Module:

### DC 24V- Ausgabe-Module

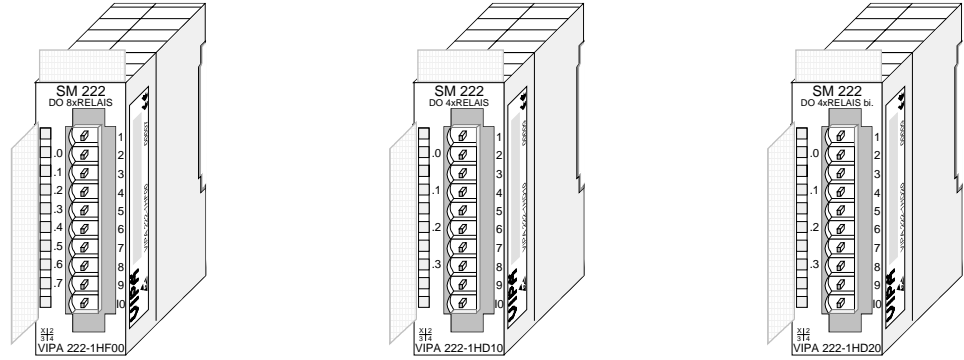


### Bestelldaten DC 24V- Ausgabe-Module

Typ	Bestellnummer	Seite
DO 8xDC 24V 1A	VIPA 222-1BF00	14-4
DO 8xDC 24V 2A	VIPA 222-1BF10	14-6
DO 8xDC 24V 2A potenzialgetrennt 4 á 2	VIPA 222-1BF20	14-8
DO 16xDC 24V 0,5A mit UB4x	VIPA 222-1BH00	14-10
DO 16xDC 24V 1A	VIPA 222-1BH10	14-12
DO 16xDC 24V 2A	VIPA 222-1BH20	14-14
DO 16xDC 24V 0,5A NPN	VIPA 222-1BH50	14-16
DO 32xDC 24V 1A	VIPA 222-2BL10	14-18



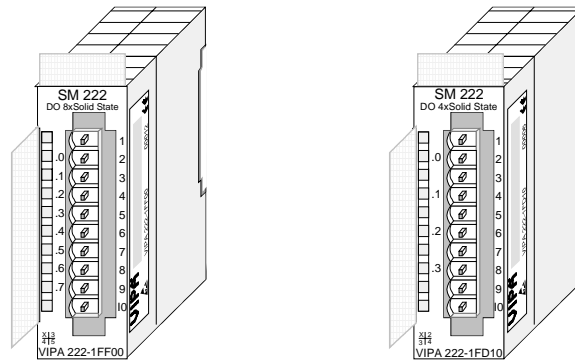
**Relais-Ausgabe-Module**



**Bestelldaten Relais-Ausgabe-Module**

Typ	Bestellnummer	Seite
DO 8xRelais COM	VIPA 222-1HF00	14-20
DO 4xRelais	VIPA 222-1HD10	14-22
DO 4xRelais bistabil	VIPA 222-1HD20	14-24

**Solid-State-Ausgabe-Module**



**Bestelldaten Solid-State-Ausgabe-Module**

Typ	Bestellnummer	Seite
DO 8xSolid State COM	VIPA 222-1FF00	14-26
DO 4xSolid State	VIPA 222-1FD10	14-28

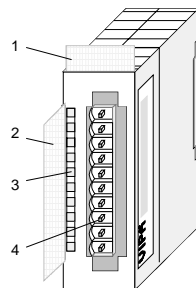
## DO 8xDC 24V 1A

**Bestelldaten** DO 8xDC 24V 1A VIPA 222-1BF00

**Beschreibung** Das digitale Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über die Ausgänge an die Prozessebene. Das Modul ist mit DC 24V über den Frontstecker zu versorgen. Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 8 Ausgänge, potenzialgetrennt zum Rückwandbus
  - Versorgungsspannung DC 24V
  - Ausgangsstrom 1A
  - Geeignet für Magnetventile und Gleichstromschütze
  - LEDs für Versorgungsspannung und Fehlermeldung
  - Aktivanzeige der Kanäle durch LED

**Aufbau**



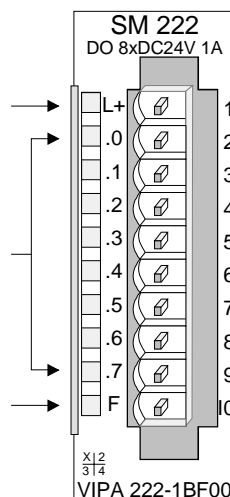
- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bitadresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

- L+ LED (gelb)  
Versorgungsspannung liegt an
- .0... .7 LEDs (grün)  
A.0 bis A.7  
sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert
- F LED (rot)  
Fehler bei Überlast, Überhitzung oder Kurzschluss

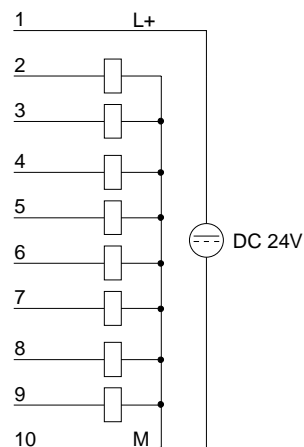
**Pin Belegung**



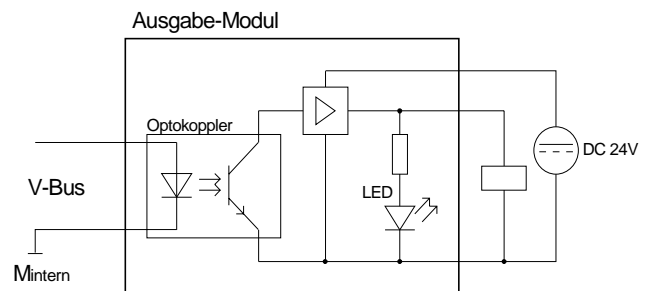
- 1 Versorgungsspg. DC 24V
- 2 Ausgang A.0
- 3 Ausgang A.1
- 4 Ausgang A.2
- 5 Ausgang A.3
- 6 Ausgang A.4
- 7 Ausgang A.5
- 8 Ausgang A.6
- 9 Ausgang A.7
- 10 Versorgungsspg. Masse

Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 222-1BF00
Anzahl der Ausgänge	8
Nennlastspannung	DC 24V (18...35V) über ext. Netzteil
Stromaufnahme an L+ ohne Last (alle A.x=aus)	10mA
Ausgangsstrom je Kanal	1A dauerkurzschlussfest
Summenstrom aller 8 Kanäle	8A
Stromaufnahme über Rückwandbus	50mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	1Byte
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	50g

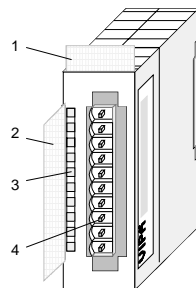
## DO 8xDC 24V 2A

**Bestelldaten** DO 8xDC 24V 2A VIPA 222-1BF10

**Beschreibung** Das digitale Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über die Ausgänge an die Prozessebene. Das Modul ist mit DC 24V über den Frontstecker zu versorgen. Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen. Jeder Ausgang kann mit bis zu 2A belastet werden.

- Eigenschaften**
- 8 Ausgänge, potenzialgetrennt zum Rückwandbus
  - Versorgungsspannung DC 24V
  - Ausgangsstrom 2A
  - Geeignet für Magnetventile und Gleichstromschütze
  - LEDs für Versorgungsspannung und Fehlermeldung
  - Aktivanzeige der Kanäle durch LED

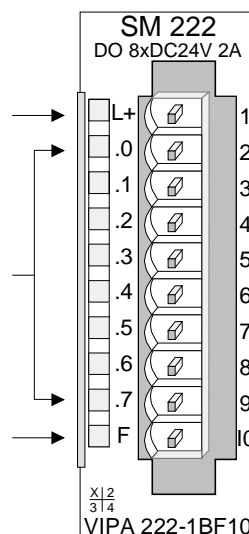
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bitadresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

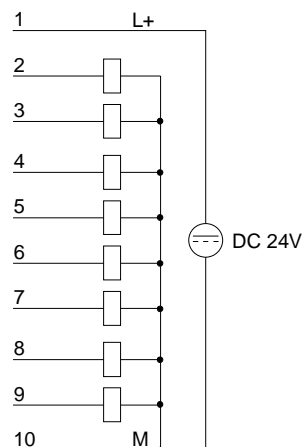
LED	Beschreibung
L+	LED (gelb) Versorgungsspannung liegt an
.0... .7	LED (grün) A.0 bis A.7 sobald ein Ausgang aktiv ist wird die entsprechende LED angesteuert
F	LED (rot) Fehler bei Überlast, Überhitzung, Kurzschluss



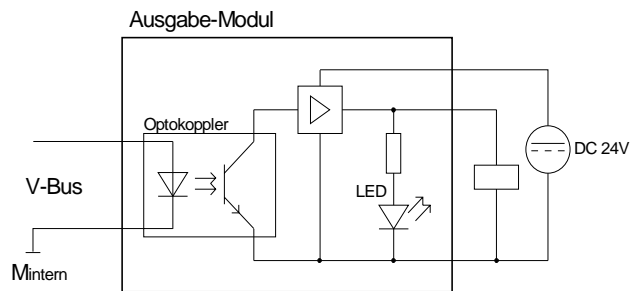
Pin	Belegung
1	Versorgungsspg. DC 24V
2	Ausgang A.0
3	Ausgang A.1
4	Ausgang A.2
5	Ausgang A.3
6	Ausgang A.4
7	Ausgang A.5
8	Ausgang A.6
9	Ausgang A.7
10	Versorgungsspg. Masse

Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 222-1BF10
Anzahl der Ausgänge	8
Nennlastspannung	DC 24V (18...35V) über ext. Netzteil
Stromaufnahme an L+ ohne Last (alle A.x=aus)	10mA
Ausgangsstrom je Kanal	2A dauerkurzschlussfest
Gleichzeitigkeitsfaktor	$I_D=50\%$ (8A)
Stromaufnahme über Rückwandbus	50mA
Summenstrom aller 8 Kanäle	12A
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	1Byte
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	50g

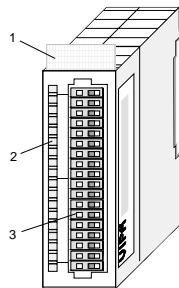
## DO 8xDC 24V 2A potenzialgetrennt 4 á 2

**Bestelldaten** DO 8xDC 24V 2A VIPA 222-1BF20

**Beschreibung** Das digitale Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über die Ausgänge an die Prozessebene. Das Modul ist mit DC 24V über den Frontstecker zu versorgen. Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen. Jeder Ausgang kann mit bis zu 2A belastet werden.

- Eigenschaften**
- 8 Ausgänge, potenzialgetrennt zum Rückwandbus
  - Potenzialtrennung in 4 Gruppen à 2 Ausgänge
  - Versorgungsspannung DC 24V
  - Ausgangsstrom 2A
  - Geeignet für Magnetventile und Gleichstromschütze
  - LEDs für Versorgungsspannung und Fehlermeldung
  - Aktivanzeige der Kanäle durch LED

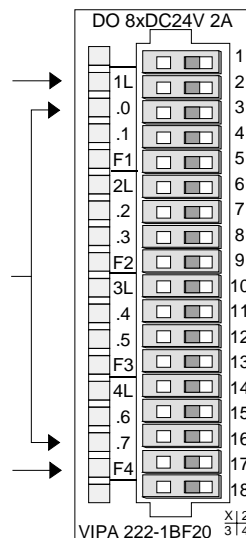
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] LED-Statusanzeige
- [3] Steckerleiste

**Statusanzeige  
Steckerbelegung**

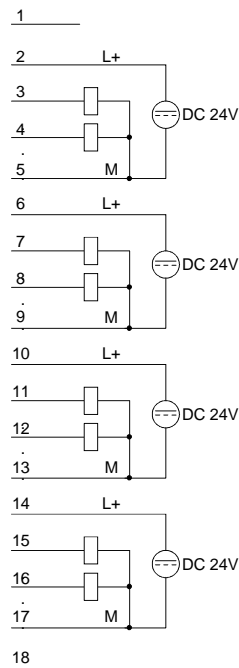
LED	Beschreibung
1L...4L	LED (gelb) Versorgungsspannung liegt an
.0... .7	LED (grün) A.0 bis A.7 sobald ein Ausgang aktiv ist wird die entsprechende LED angesteuert
F1...F4	LED (rot) Fehler bei Überlast, Überhitzung, Kurzschluss



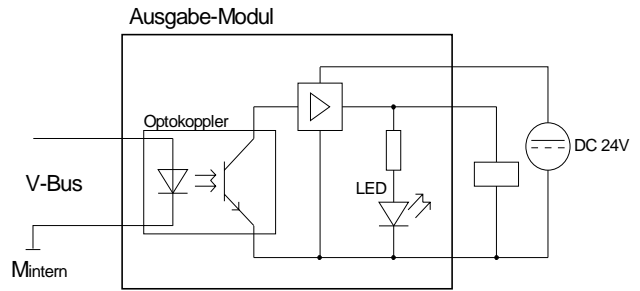
Pin	Belegung
1	nicht belegt
2	Versorgungsspg. 1L+
3	Ausgang A.0
4	Ausgang A.1
5	Masse 1M
6	Versorgungsspg. 2L+
7	Ausgang A.2
8	Ausgang A.3
9	Masse 2M
...	...
14	Versorgungsspg. 4L+
15	Ausgang A.6
16	Ausgang A.7
17	Masse 4M
18	nicht belegt

**Anschlussbild und  
Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Technische Daten**

Elektrische Daten	VIPA 222-1BF20
Anzahl der Ausgänge	8
Nennlastspannung	DC 24V (18...35V) über ext. Netzteil
Stromaufnahme an L+ ohne Last (alle A.x=aus)	10mA
Ausgangsstrom je Kanal	2A dauerkurzschlussfest
Stromaufnahme über Rückwandbus	50mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	1Byte
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	50g

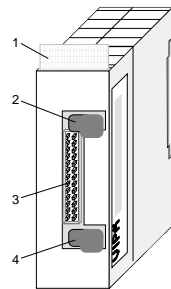
## DO 16xDC 24V 0,5A mit UB4x

**Bestelldaten** DO 16xDC 24V 0,5A VIPA 222-1BH00

**Beschreibung** Das digitale Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über die Ausgänge an die Prozessebene. Das Modul ist mit 24V über den Frontstecker zu versorgen. Es hat 16 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen. Für den Einsatz des Moduls ist eine Umsetzbaugruppe (DEA-UB4x) erforderlich. Das Modul ist über ein Flachrundkabel (DEA-KB91C) mit der Umsetzbaugruppe zu verbinden.

- Eigenschaften**
- 16 Ausgänge, potenzialgetrennt zum Rückwandbus
  - Versorgungsspannung DC 24V
  - Ausgangsstrom 0,5A
  - Geeignet für Magnetventile und Gleichstromschütze
  - LEDs für Versorgungsspannung und Fehlermeldung
  - Aktivanzeige der Kanäle durch LED auf der Umsetzbaugruppe UB4x

**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Verriegelung
- [3] Wannenstecker zum Anschluss der Umsetzbaugruppe UB4x über Flachrundkabel
- [4] Verriegelung

**Statusanzeige auf UB4x**

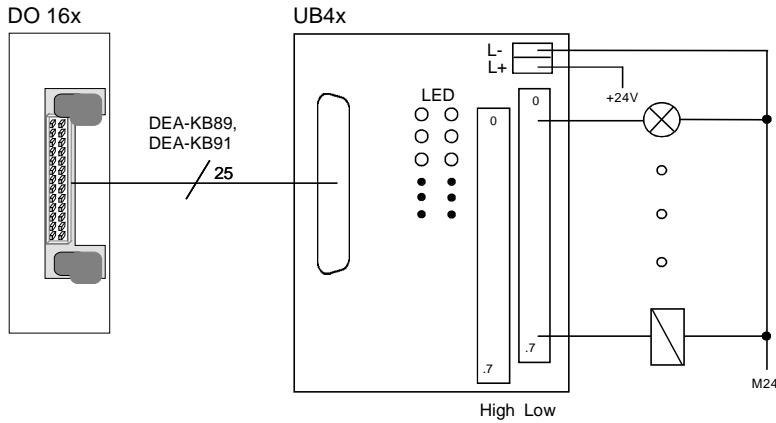
LED	Beschreibung
0... .15	LEDs (gelb) A.0 bis A.7 High A.0 bis A.7 Low sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert
L+ L-	LED (grün) Versorgungsspannung liegt an

**Steckerbelegung Modul**

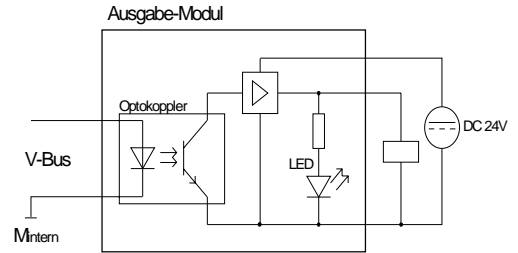
Stecker	Pin	Belegung
	23...26	Versorgungssp. DC 24V
	22	Ausgang A.0
	21	Ausgang A.1
	.	.
	.	.
	.	.
	8	Ausgang A.14
	7	Ausgang A.15
	1...6	Versorgungssp. Masse



**Anschluss UB4x**



**Prinzipschaltbild Modul**



**Technische Daten**

Elektrische Daten	VIPA 222-1BH00
Anzahl der Ausgänge	16
Nennlastspannung	DC 24V (18 ... 35V) über ext. Netzteil
Stromaufnahme an L+ ohne Last (alle A.x=aus)	10mA
Ausgangsstrom je Kanal	0,5A kurzschlussfest
Stromaufnahme über Rückwandbus	100mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf UB4x
<b>Programmierdaten</b>	
Eingabedaten	-
Ausgabedaten	2Byte
Parameterdaten	-
Diagnosedaten	-
<b>Maße und Gewicht</b>	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	50g

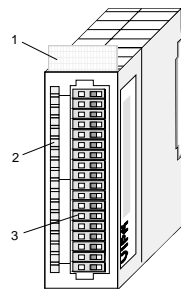
# DO 16xDC 24V 1A

**Bestelldaten** DO 16xDC 24V 1A VIPA 222-1BH10

**Beschreibung** Das digitale Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über die Ausgänge an die Prozessebene. Das Modul ist mit 24V über den Frontstecker zu versorgen. Es hat 16 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 16 Ausgänge, potenzialgetrennt zum Rückwandbus
  - Versorgungsspannung DC 24V
  - Ausgangsstrom 1A
  - Geeignet für Magnetventile und Gleichstromschütze
  - LEDs für Versorgungsspannung und Fehlermeldung
  - Aktivanzeige der Kanäle durch LED

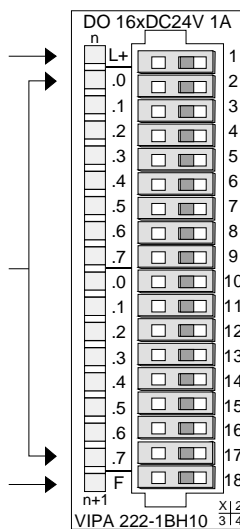
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] LED-Statusanzeige
- [3] Steckerleiste

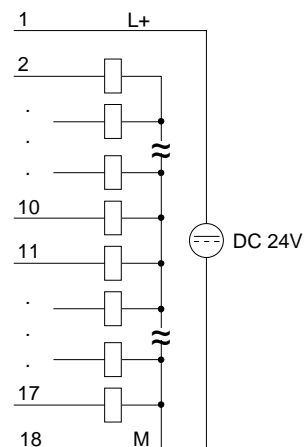
**Statusanzeige  
Steckerbelegung**

LED	Beschreibung	Pin	Belegung
L+	LED (gelb) Versorgungsspannung liegt an	1	Versorgungsspg. DC 24V
A.0 ... A.7	LEDs (grün) A.0 bis A.7 (je Byte) sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert	2	Ausgang A.0
		3	Ausgang A.1
		.	.
		.	.
		.	.
		16	Ausgang A.14
		17	Ausgang A.15
F	LED (rot) Fehler bei Überlast, Überhitzung oder Kurzschluss	18	Versorgungsspg. Masse

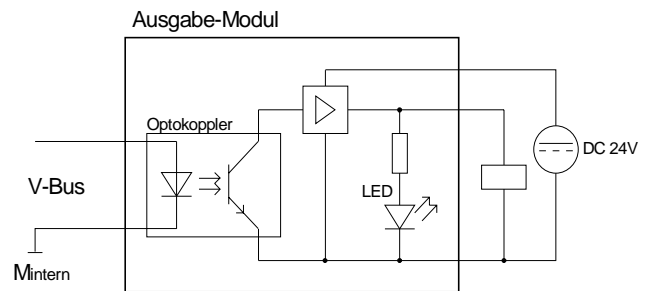


Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 222-1BH10
Anzahl der Ausgänge	16
Nennlastspannung	DC 24V (18 ... 35V) über ext. Netzteil
Stromaufnahme an L+ ohne Last (alle A.x=aus)	10mA
Ausgangsstrom je Kanal	1A dauerkurzschlussfest
max. Summenausgangsstrom	10A
Stromaufnahme über Rückwandbus	80mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	2Byte
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	50g

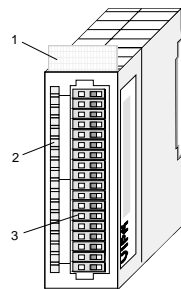
# DO 16xDC 24V 2A

**Bestelldaten** DO 16xDC 24V 2A VIPA 222-1BH20

**Beschreibung** Das digitale Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über die Ausgänge an die Prozessebene. Das Modul ist mit 24V über den Frontstecker zu versorgen. Es hat 16 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 16 Ausgänge, potenzialgetrennt zum Rückwandbus
  - Versorgungsspannung DC 24V
  - Ausgangsstrom 2A
  - Geeignet Magnetventile und Gleichstromschütze
  - LEDs für Versorgungsspannung und Fehlermeldung
  - Aktivanzeige der Kanäle durch LED

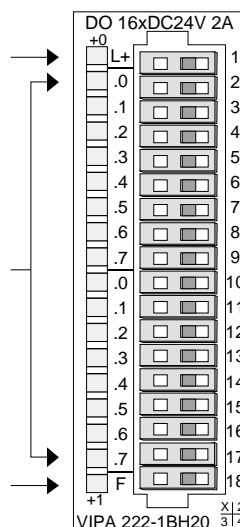
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] LED-Statusanzeige
- [3] Steckerleiste

**Statusanzeige Steckerbelegung**

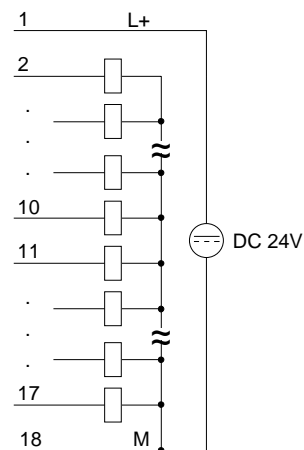
LED	Beschreibung
L+	LED (gelb) Versorgungsspannung liegt an
A.0 ... A.7	LEDs (grün) A.0 bis A.7 (je Byte) sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert
F	LED (rot) Fehler bei Überlast, Überhitzung oder Kurzschluss



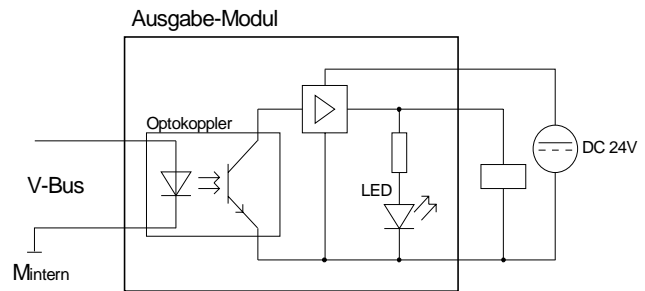
Pin	Belegung
1	Versorgungsspg. DC 24V
2	Ausgang A.0
3	Ausgang A.1
.	.
.	.
.	.
16	Ausgang A.14
17	Ausgang A.15
18	Versorgungsspg. Masse

Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 222-1BH20
Anzahl der Ausgänge	16
Nennlastspannung	DC 24V (18 ... 35V) über ext. Netzteil
Stromaufnahme an L+ ohne Last (alle A.x=aus)	10mA
Ausgangsstrom je Kanal	2A dauerkurzschlussfest
max. Summenausgangsstrom	10A
Stromaufnahme über Rückwandbus	100mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	2Byte
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	50g

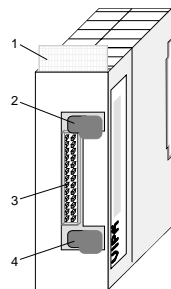
## DO 16xDC 24V 0,5A NPN

**Bestelldaten** DO 16xDC 24V 0,5A NPN VIPA 222-1BH50

**Beschreibung** Das Modul erfasst die binären Steuersignale aus dem Bussystem und aktiviert über Mosfet-Ausgänge Lasten der Prozessebene. Es hat 16, über die Lastspannung verbundene Kanäle, die als Low-Side-Schalter arbeiten. Low-Side-Schalter eignen sich zum Schalten von Massen. Bei einem Kurzschluss zwischen Schaltleitung und Masse wird lediglich die Last aktiviert, bis der Kurzschluss wieder behoben ist. Es entsteht kein Kurzschluss, der die Versorgungsspannung belastet.

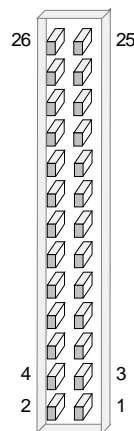
- Eigenschaften**
- 16 Low-Side-Ausgänge
  - Maximale externe Lastspannung DC 32V
  - Ausgangstrom pro Kanal 0,5A
  - Geeignet für Kleinmotoren, Lampen, Magnetventile und Schütze

**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Verriegelung
- [3] Wannenstecker zum Anschluss einer Ausgabebeschaltung
- [4] Verriegelung

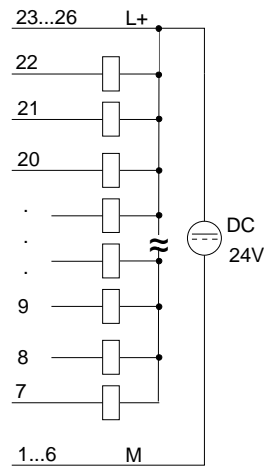
**Steckerbelegung Wannenstecker**



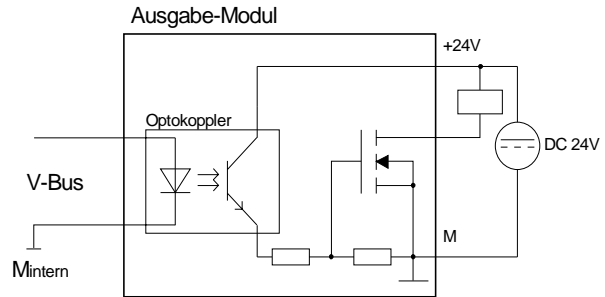
Pin	Belegung
23...26	Versorgungssp. DC 24V
22	Ausgang A.0
21	Ausgang A.1
.	.
.	.
.	.
8	Ausgang A.14
7	Ausgang A.15
1...6	Versorgungssp. Masse

**Anschlussbild und Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Achtung!**

Das Modul ist ohne technischen Eingriff mit einer UB4x von VIPA nicht einsetzbar. Setzen Sie sich hierzu mit der VIPA-Hotline in Verbindung.

**Technische Daten**

Elektrische Daten	VIPA 222-1BH50
Anzahl der Ausgänge	16 über Low-Side
Nennlastspannung	max. DC 24V
max. Ausgangsstrom je Kanal	0,5A
Stromaufnahme über Rückwandbus	100mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Schaltfrequenz	max. 20kHz
Statusanzeige	-
Programmierdaten	
Eingabedaten	-
Ausgabedaten	2Byte
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g

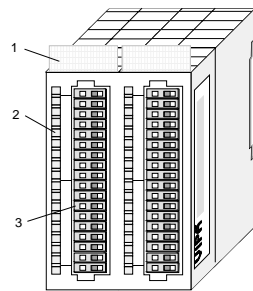
# DO 32xDC 24V 1A

**Bestelldaten** DO 32xDC 24V 1A VIPA 222-2BL10

**Beschreibung** Das digitale Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über die Ausgänge an die Prozessebene. Das Modul ist mit 24V über den Frontstecker zu versorgen. Es hat 32 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 32 Ausgänge, potenzialgetrennt zum Rückwandbus
  - Versorgungsspannung DC 24V
  - Ausgangsstrom je Kanal 1A
  - Geeignet für Magnetventile und Gleichstromschütze
  - LEDs für Versorgungsspannung und Fehlermeldung
  - Aktivanzeige der Kanäle durch LED

**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] LED-Statusanzeige
- [3] Steckerleiste

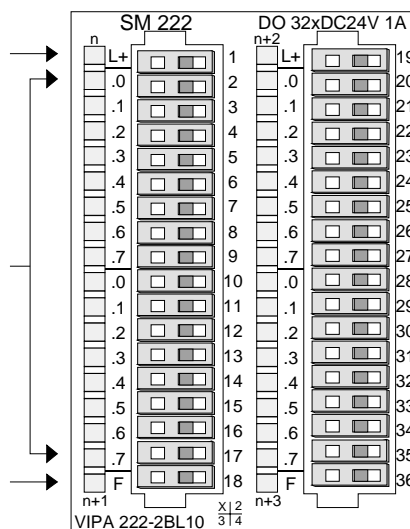
**Statusanzeige Steckerbelegung**

**LED Beschreibung**

- L+ LED (gelb)  
Versorgungsspannung liegt an
- .0 ... .7 LED (grün)  
A.0 bis A.7 (je Byte) sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert
- F LED (rot)  
Fehler bei Überlast, Überhitzung oder Kurzschluss

**Pin Belegung**

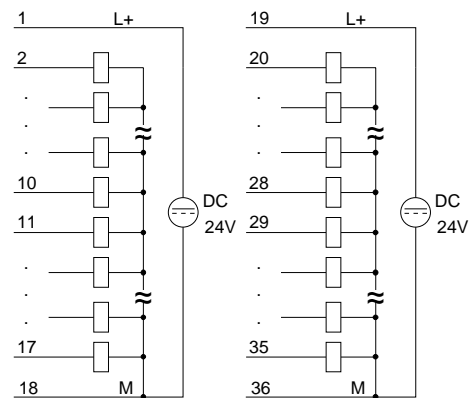
- 1 Versorgungsspg. DC 24V
- 2 Ausgang A.0
- 3 Ausgang A.1
- ... ..
- 17 Ausgang A.15
- 18 Versorgungsspg. Masse
- 19 Versorgungsspg. DC 24V
- 20 Ausgang A.16
- ... ..
- 34 Ausgang A.30
- 35 Ausgang A.31
- 36 Versorgungsspg. Masse



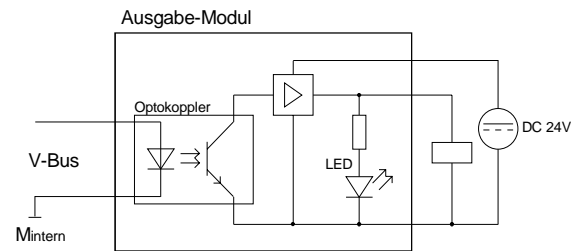


Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 222-2BL10
Anzahl der Ausgänge	32
Nennlastspannung	DC 24V (18 ... 35V) über ext. Netzteil
Stromaufnahme an L+ ohne Last (alle A.x=aus)	15mA
max. Ausgangsstrom je Kanal	1A dauerkurzschlussfest
max. Summenstrom je Reihe	10A
Stromaufnahme über Rückwandbus	140mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	in 2 Gruppen zu je 16 Ausgängen 500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	4Byte
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x76
Gewicht	50g

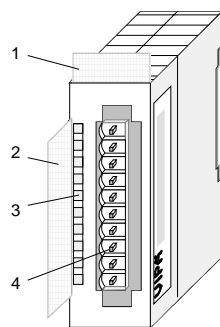
## DO 8xRelais COM

**Bestelldaten** DO 8xRelais COM VIPA 222-1HF00

**Beschreibung** Das digitale Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über Relais-Ausgänge an die Prozessebene. Das Modul wird über den Rückwandbus versorgt. An Klemme 1 ist die Lastspannung anzulegen. Übersteigt der Summenstrom 8A, verteilen Sie den Laststrom auf die Klemmen 1 und 10. Es hat 8 Kanäle, die ihren Zustand durch Leuchtdioden anzeigen.

- Eigenschaften**
- 8 Relais-Ausgänge
  - Versorgung über Rückwandbus
  - Externe Lastspannung AC 230V / DC 30V
  - Ausgangsstrom pro Kanal 5A (AC 230V / DC 30V)
  - Geeignet für Motoren, Lampen, Magnetventile und Gleichstromschütze
  - Aktivanzeige der Kanäle durch LED

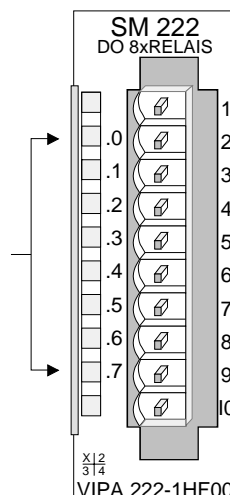
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bitadresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

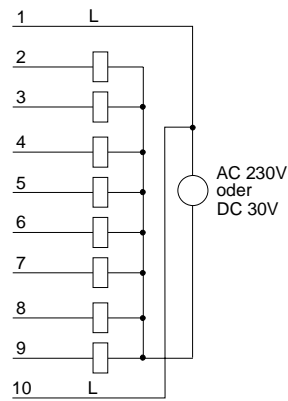
LED	Beschreibung
.0... .7	LED (grün) A.0 bis A.7 sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert



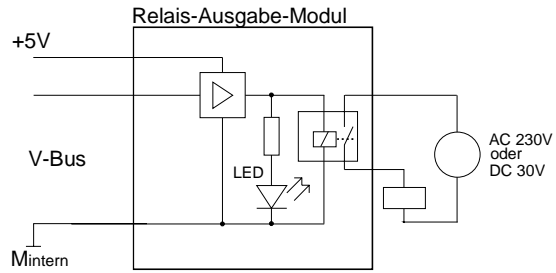
Pin	Belegung
1	Versorgungssp. L
2	Relais-Ausg. A.0
3	Relais-Ausg. A.1
4	Relais-Ausg. A.2
5	Relais-Ausg. A.3
6	Relais-Ausg. A.4
7	Relais-Ausg. A.5
8	Relais-Ausg. A.6
9	Relais-Ausg. A.7
10	Versorgungssp. L

**Anschlussbild und  
Prinzipschaltbild**

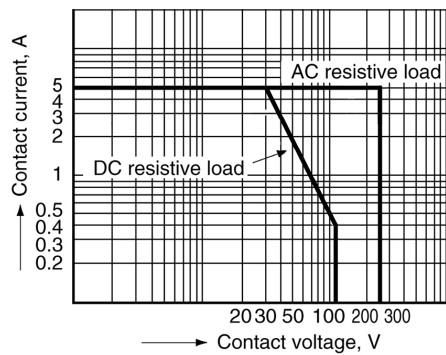
**Anschlussbild**



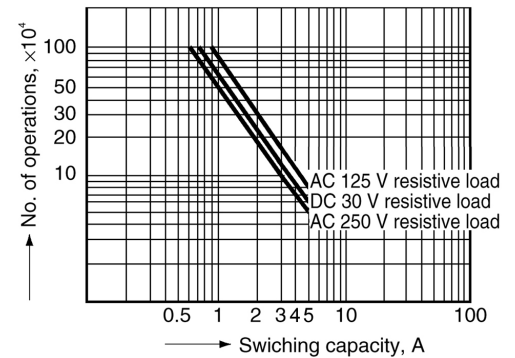
**Prinzipschaltbild**



**Maximale Schaltleistung**



**Lebensdauer**



**Technische Daten**

Elektrische Daten	VIPA 222-1HF00
Anzahl der Ausgänge	8 über Relais
Nennlastspannung	AC 230V oder max. DC 30V
Stromaufnahme an L+ ohne Last (alle A.x=aus)	-
Summenstrom	bei 1 L: max. 8A bei 2 L: max. 16A
max. Ausgangsstrom je Kanal	AC 230V: 5A / DC 30V: 5A
Stromaufnahme über Rückwandbus	270mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung- Rückwandbus)
Schaltfrequenz	max. 100Hz
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	1Byte
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g

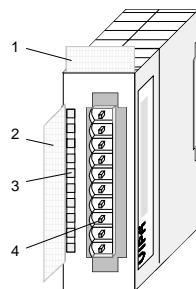
## DO 4xRelais

**Bestelldaten** DO 4xRelais VIPA 222-1HD10

**Beschreibung** Das Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über Relais-Ausgänge an die Prozessebene. Es wird über den Rückwandbus versorgt. Es hat 4 getrennte Kanäle, die als Schalter arbeiten und ihren Zustand durch Leuchtdioden anzeigen. Bei aktiven Verbrauchern sind diese extern zu versorgen.

- Eigenschaften**
- 4 galvanisch getrennte Relais-Ausgänge
  - Versorgung über Rückwandbus
  - Externe Lastspannung AC 230V / DC 30V (mischbar)
  - Max. Ausgangsstrom pro Kanal 5A (AC 230V / DC 30V)
  - Geeignet für Motoren, Lampen, Magnetventile und Gleichstromschütze
  - Aktivanzeige der Kanäle durch LED

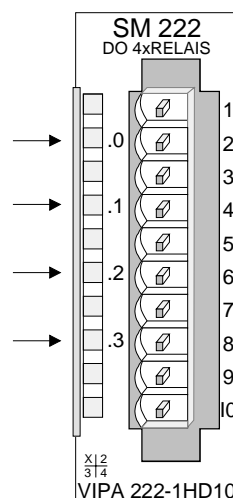
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

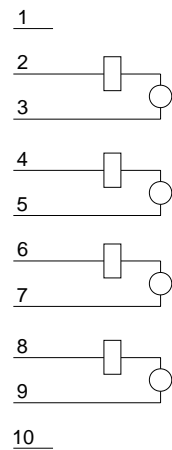
LED	Beschreibung
.0... .3	LED (grün) A.0 bis A.3 sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert



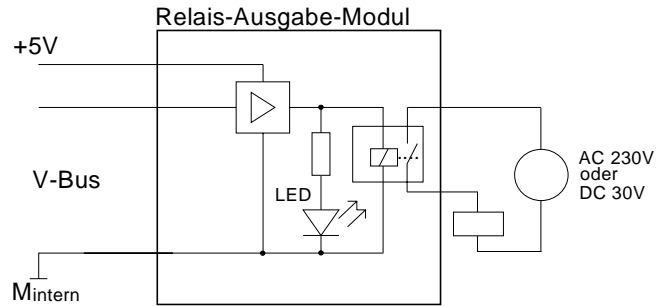
Pin	Belegung
1	nicht belegt
2+3	Relais-Ausg. A.0
4+5	Relais-Ausg. A.1
6+7	Relais-Ausg. A.2
8+9	Relais-Ausg. A.3
10	nicht belegt

**Anschlussbild und Prinzipschaltbild**

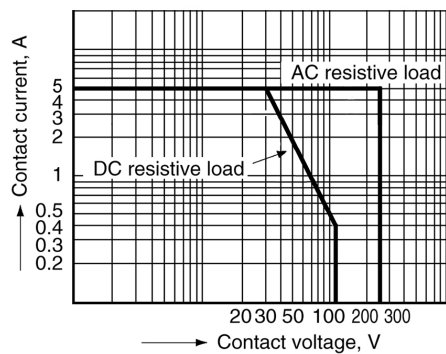
**Anschlussbild**



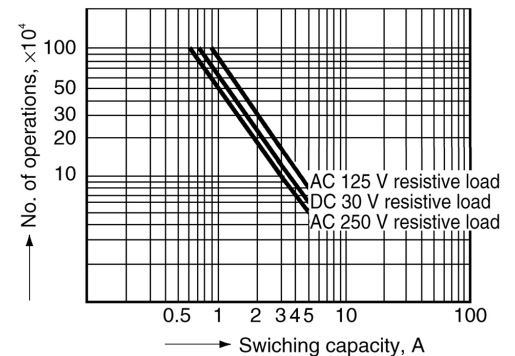
**Prinzipschaltbild**



**Maximale Schaltleistung**



**Lebensdauer**



**Technische Daten**

Elektrische Daten	VIPA 222-1HD10
Anzahl der Ausgänge	4 über Relais
Nennlastspannung	AC 230V oder max. DC 30V
max. Ausgangsstrom	AC 230V: 5A / DC 30V: 5A
Stromaufnahme über Rückwandbus	150mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Schaltfrequenz	max. 100Hz
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	1Byte (Bit 0 ... Bit 3)
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g

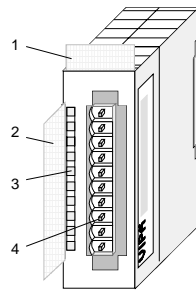
## DO 4xRelais bistabil

**Bestelldaten** DO 4xRelais bistabil VIPA 222-1HD20

**Beschreibung** Das digitale Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über bistabile Relais-Ausgänge an die Prozessebene. Das Modul wird über den Rückwandbus versorgt. Es hat 4 getrennte Kanäle, die als Schalter arbeiten. Der Schaltzustand bleibt bei Spannungsausfall des übergeordneten Bussystems erhalten.

- Eigenschaften**
- 4 galvanisch getrennte Relais-Ausgänge
  - Versorgung über Rückwandbus
  - Externe Lastspannung AC 230V / DC 30V (mischbar)
  - Max. Ausgangsstrom pro Kanal 16A (AC 230V / DC 30V)
  - Geeignet für Motoren, Lampen, Magnetventile und Gleichstromschütze

**Aufbau**

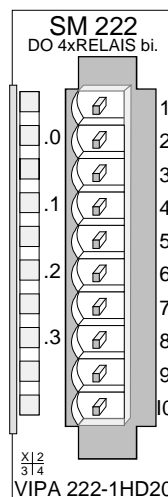


- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bitadresse mit Beschreibung
- [3] LEDs (ohne Funktion)
- [4] Steckerleiste

**Ausgabebyte / Steckerbelegung**

Bit	Beschreibung
Bit 0	set A.0
Bit 1	set A.1
Bit 2	set A.2
Bit 3	set A.3
Bit 4	reset A.0
Bit 5	reset A.1
Bit 6	reset A.2
Bit 7	reset A.3

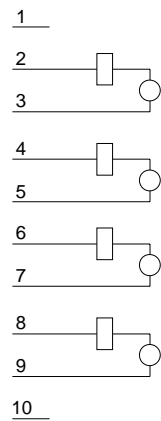
Durch Setzen der Bits 0...3 wird der entsprechende Kanal aktiviert.  
 Ein Setzen der Bits 4..7 bewirkt das Rücksetzen des entsprechenden Ausgangs nach frühestens 50ms.



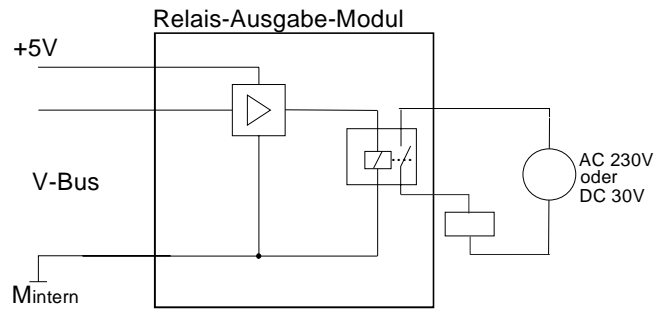
Pin	Belegung
1	nicht belegt
2+3	Relais-Ausg. A.0
4+5	Relais-Ausg. A.1
6+7	Relais-Ausg. A.2
8+9	Relais-Ausg. A.3
10	nicht belegt

**Anschlussbild und Prinzipschaltbild**

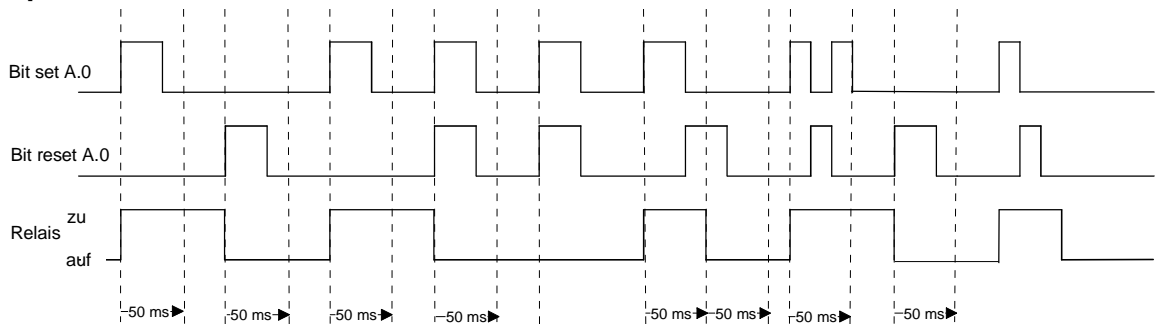
**Anschlussbild**



**Prinzipschaltbild**



**Signalflussplan**



**Hinweis!**

Bitte beachten Sie, dass ein gesetzter Relais-Ausgang frühestens nach 50ms zurückgesetzt werden kann, sofern kein Set-Signal ansteht!

**Technische Daten**

Elektrische Daten	VIPA 222-1HD20
Anzahl der Ausgänge	4 über Relais
Nennlastspannung	AC 230V oder DC 30V
max. Ausgangsstrom je Kanal	AC 230V: 16A / DC 30V: 16A
Stromaufnahme über Rückwandbus	40mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Schaltfrequenz	max. 100Hz
Statusanzeige	-
Programmierdaten	
Eingabedaten	-
Ausgabedaten	1Byte
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g

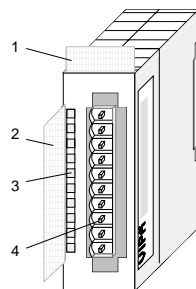
## DO 8xSolid State COM

**Bestelldaten** DO 8xSolid State COM VIPA 222-1FF00

**Beschreibung** Das Solid-State-Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über Halbleiterrelais-Ausgänge an die Prozessebene. Das Modul wird über den Rückwandbus versorgt. Es hat 8 über die Lastspannung verbundene Kanäle, die als Schalter arbeiten und ihren Zustand über LEDs anzeigen. Solid State-Relais schalten im Nulldurchgang der Lastspannung (Wechselspannung).

- Eigenschaften**
- 8 Solid State-Ausgänge mit Aktivanzeige der Kanäle durch LED
  - Lange Lebensdauer da Lastspannung (wenn Wechselspannung) im Nulldurchgang geschaltet wird
  - Externe Lastspannung AC 230V oder DC 400V
  - Max. Ausgangsstrom pro Kanal 0,5A (AC 230V / DC 400V)
  - Geeignet für Kleinmotoren, Lampen, Magnetventile und Schütze

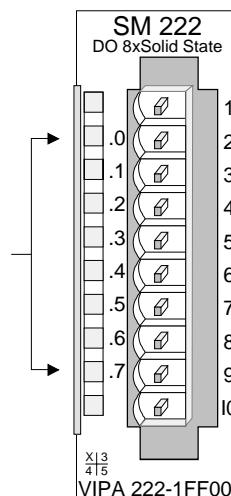
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

LED	Beschreibung
.0 ... .7	LED (grün) A.0 bis A.7 sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert

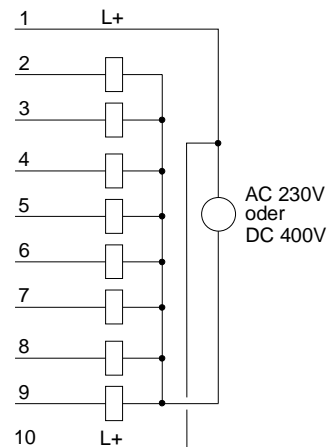


Pin	Belegung
1	Versorgungsspg.
2	Ausgang A.0
3	Ausgang A.1
4	Ausgang A.2
5	Ausgang A.3
6	Ausgang A.4
7	Ausgang A.5
8	Ausgang A.6
9	Ausgang A.7
10	Versorgungsspg.

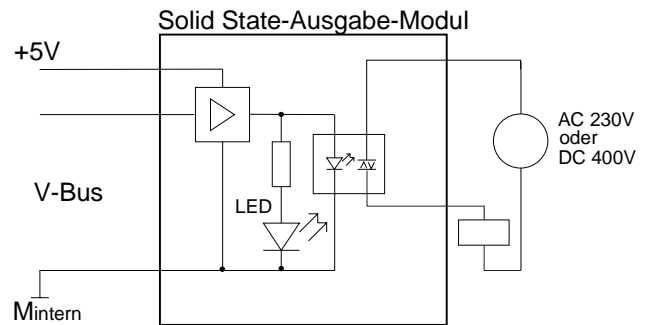


Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 222-1FF00
Anzahl der Ausgänge	8 über Solid State
Nennlastspannung	AC 230V oder DC 400V
max. Ausgangsstrom je Kanal	AC 230V: 0,5A / DC 400V: 0,5A
Kontaktwiderstand	typ. 2,1Ω , max. 3,2Ω
Stromaufnahme über Rückwandbus	140mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Schaltfrequenz	alle 50ms
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	1Byte (Bit 0 ... Bit 7)
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g

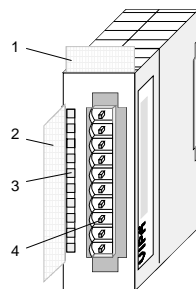
## DO 4xSolid State

**Bestelldaten** DO 4xSolid State VIPA 222-1FD10

**Beschreibung** Das digitale Ausgabe-Modul erfasst die binären Steuersignale aus dem übergeordneten Bussystem und transportiert sie über Halbleiterrelais-Ausgänge an die Prozessebene. Das Modul wird über den Rückwandbus versorgt. Es hat 4 getrennte Kanäle, die als Schalter arbeiten und ihren Zustand durch Leuchtdioden anzeigen. Bei aktiven Verbrauchern sind diese extern zu versorgen.

- Eigenschaften**
- 4 galvanisch getrennte Solid State-Ausgänge
  - Versorgung über Rückwandbus
  - Externe Lastspannung AC 230V oder DC 400V
  - Max. Ausgangsstrom pro Kanal 0,5A (AC 230V / DC 400V)
  - Geeignet für Motoren, Lampen, Magnetventile und Gleichstromschütze
  - Aktivanzeige der Kanäle durch LED

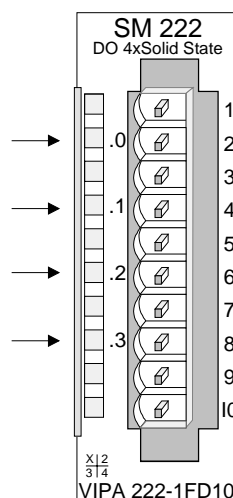
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bitadresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

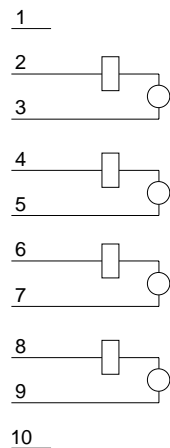
LED	Beschreibung
.0... .3	LED (grün) A.0 bis A.3 sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert



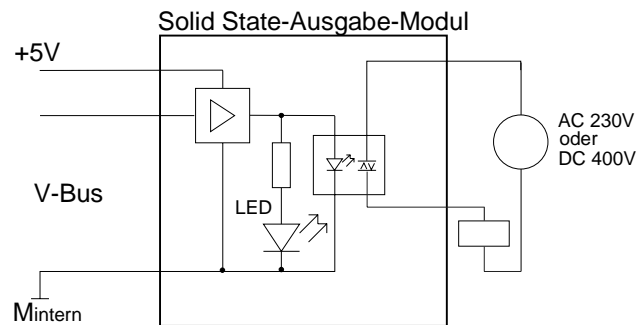
Pin	Belegung
1	nicht belegt
2+3	Ausgang A.0
4+5	Ausgang A.1
6+7	Ausgang A.2
8+9	Ausgang A.3
10	nicht belegt

Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 222-1FD10
Anzahl der Ausgänge	4 über Solid State
Nennlastspannung	AC 230V oder DC 400V
max. Ausgangstrom je Kanal	AC 230V: 0,5A / DC 400V: 0,5A
Stromaufnahme über Rückwandbus	100mA
Spannungsversorgung	5V über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Schaltfrequenz	max. 100Hz
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	1Byte (Bit 0 ... Bit 3)
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	80g



## Teil 15 Digitale Ein-/Ausgabe-Module

### Überblick

Inhalt dieses Kapitels ist der Aufbau und die Funktionsweise der digitalen Ein-/Ausgabe-Module von VIPA.

Nachfolgend sind beschrieben:

- Systemübersicht der Digitalen Ein-/Ausgabe-Module
- Eigenschaften
- Aufbau
- Anschluss- und Prinzipschaltbild
- Technische Daten

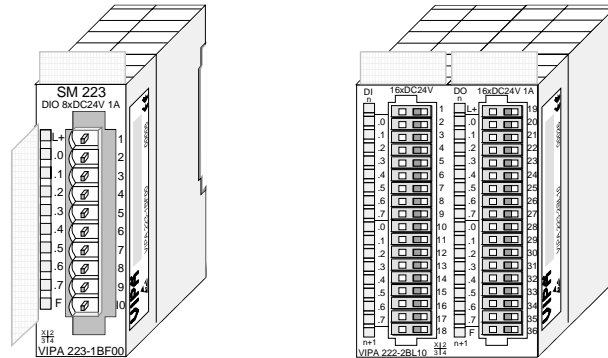
### Inhalt

Thema	Seite
<b>Teil 15 Digitale Ein-/Ausgabe-Module</b> .....	<b>15-1</b>
Systemübersicht .....	15-2
Sicherheitshinweise zum Einsatz der DIO-Module .....	15-2
DIO 8xDC 24V 1A.....	15-3
DI 16xDC 24V, DO 16xDC 24V 1A .....	15-5

## Systemübersicht

### Ein-/Ausgabe-Module SM 223

Nachfolgend finden Sie eine Übersicht der bei VIPA zur Zeit erhältlichen Digitalen Ein-/Ausgabe-Module:



### Bestelldaten Ein-/ Ausgabe-Module

Typ	Bestellnummer	Seite
DIO 8xDC 24V 1A	VIPA 223-1BF00	15-3
DI 16xDC 24V, DO 16xDC 24V 1A	VIPA 223-2BL10	15-5

## Sicherheitshinweise zum Einsatz der DIO-Module



### Achtung!

Bitte beachten Sie, dass die an einem Ausgabe-Kanal anliegende Spannung immer  $\leq$  der über L+ anliegenden Versorgungsspannung ist.

Weiter ist zu beachten, dass aufgrund der Parallelschaltung von Ein- und Ausgabe-Kanal je Gruppe ein gesetzter Ausgang über ein anliegendes Eingabesignal versorgt werden kann.

Auch bei ausgeschalteter Versorgungsspannung und anliegendem Eingangssignal bleibt so ein gesetzter Ausgang aktiv.

Bei Nichtbeachtung kann dies zur Zerstörung des Moduls führen.

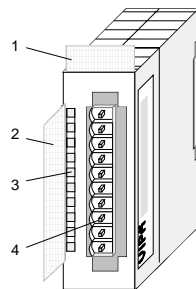
# DIO 8xDC 24V 1A

**Bestelldaten** DIO 8xDC 24V 1A VIPA 223-1BF00

**Beschreibung** Das Modul ist ein Misch-Modul. Es besitzt 8 Kanäle, die als Ein- oder Ausgabe-Kanal eingesetzt werden können. Der Zustand der Kanäle wird über LEDs angezeigt. Jeder Kanal besitzt eine Diagnosefunktion, d.h. sobald ein Ausgang aktiv ist, wird der zugehörige Eingang auf "1" gesetzt. Bei einem Kurzschluss an der Last, wird der Eingang auf "0" gezogen und durch Auswertung des Eingangs kann der Fehler erkannt werden.

- Eigenschaften**
- 8 Kanäle, potenzialgetrennt zum Rückwandbus (als Ein- oder Ausgang)
  - Diagnosefunktion
  - Eingangs-Nennspannung DC 24V / Versorgungsspannung DC 24V
  - Ausgangsstrom 1A
  - LED für Fehlermeldung bei Überlast, Überhitzung oder Kurzschluss
  - Aktivanzeige der Kanäle durch LED

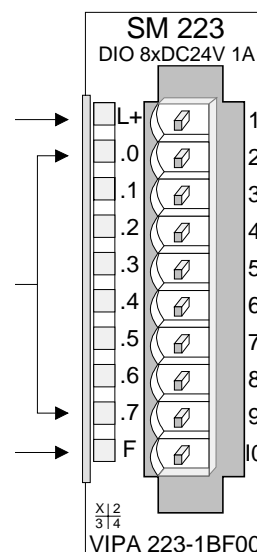
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

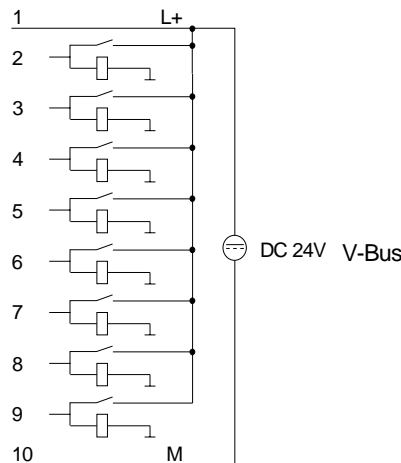
LED	Beschreibung
L+	LED (gelb) Versorgungsspannung liegt an
.0 ... .7	LED (grün) sobald Eingangssignal "1" oder Ausgang aktiv ist, wird die entsprechende LED angesteuert
F	LED (rot) Fehler bei Überlast, Überhitzung oder Kurzschluss



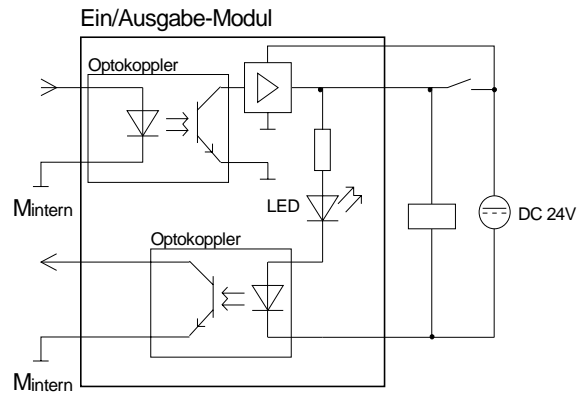
Pin	Belegung
1	Versorgungsspannung +DC 24V
2	Kanal K.0
3	Kanal K.1
4	Kanal K.2
5	Kanal K.3
6	Kanal K.4
7	Kanal K.5
8	Kanal K.6
9	Kanal K.7
10	Versorgungsspannung Masse

**Anschlussbild und  
Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Technische Daten**

Elektrische Daten	VIPA 223-1BF00
Anzahl der Kanäle	8
Nennlastspannung	DC 24V (18...35V) über ext. Netzteil
Stromaufnahme an L+ ohne Last (alle A.x=aus)	50mA
Ausgangsstrom je Kanal	1A kurzschlussfest
Summenstrom	12A
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	0 ... 5V
Signalspannung "1"	15 ... 28,8V
Eingangsfiler Zeitverzögerung	3ms
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	65mA
Datenbreite im Prozessabbild	1Byte PAA, 1Byte PAE
Statusanzeige	über LEDs auf der Frontseite
<b>Programmierdaten</b>	
Eingabedaten	1Byte
Ausgabedaten	1Byte
Parameterdaten	-
Diagnosedaten	-
<b>Maße und Gewicht</b>	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	50g



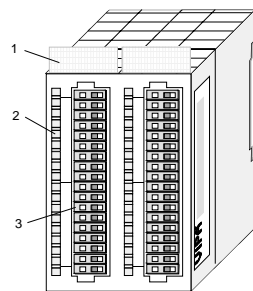
# DI 16xDC 24V, DO 16xDC 24V 1A

**Bestelldaten** DI 16xDC 24V, DO 16xDC 24V 1A VIPA 223-1BL10

**Beschreibung** Das Modul hat 32 zum Rückwandbus potenzialgetrennte Kanäle, von denen 16 als Eingänge und 16 als Ausgänge arbeiten. Der Zustand der Kanäle wird über LEDs angezeigt.

- Eigenschaften**
- 32 Kanäle davon 16 Input und 16 Output
  - Eingangs-Nennspannung DC 24V
  - Versorgungsspannung DC 24V (extern) für Ausgänge
  - Ausgangsstrom 1A pro Kanal
  - LED für Fehlermeldung bei Überlast, Überhitzung oder Kurzschluss
  - Aktivanzeige der Kanäle durch LED

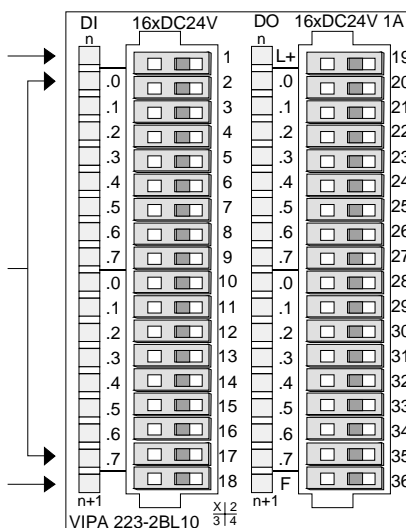
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bitadresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

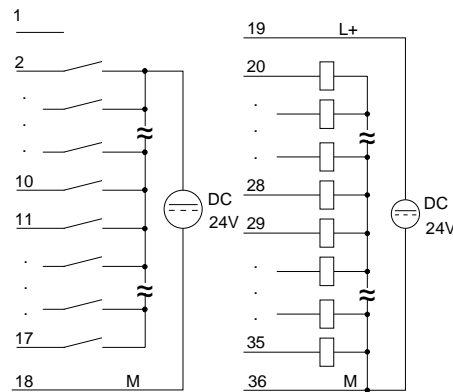
**Statusanzeige  
Steckerbelegung**

LED	Beschreibung	Pin	Belegung
L+	LED (gelb) Versorgungsspannung liegt an	1	nicht belegt
.0 ... .7	LED (grün) E.0 ... E.7 (je Byte) A.0 ... A.7 (je Byte) bei Signal "1" (Eingang) bzw. aktivem Ausgang wird die entsprechende LED angesteuert	2	Eingang E.0
F	LED (rot) Fehler bei Überlast, Überhitzung oder Kurzschluss	17	Eingang E.15
		18	Masse Eingänge
		19	Versorgungsspg. +24V
		20	Ausgang A.0
		35	Ausgang A.15
		36	Versorgungsspg. Masse Ausgänge

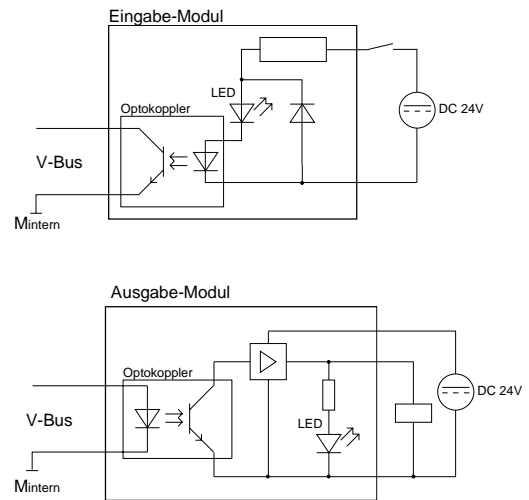


**Anschlussbild und  
Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Technische Daten**

Elektrische Daten	VIPA 223-2BL10
Anzahl der Kanäle	32
Nennlastspannung	DC 24V (18...35V) über ext. Netzteil
Stromaufnahme an L+ ohne Last (alle A.x=aus)	10mA
Ausgangsstrom je Kanal	1A kurzschlussfest
max. Kontaktbelastung je Leiste	10A
Nenneingangsspannung	DC 24V (18 ... 28,8V)
Signalspannung "0"	0 ... 5V
Signalspannung "1"	15 ... 28,8V
Eingangsfiler Zeitverzögerung	3ms
Eingangsstrom	typ. 7mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme über Rückwandbus	100mA
Datenbreite im Prozessabbild	2Byte PAA, 2Byte PAE
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	2Byte
Ausgabedaten	2Byte
Parameterdaten	-
Diagnosedaten	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x76
Gewicht	100g

## Teil 16 Analoge Eingabe-Module

### Überblick

Inhalt dieses Kapitels ist der Aufbau und die Funktionsweise der analogen Eingabe-Module von VIPA.

Nachfolgend sind beschrieben:

- Systemübersicht der analogen Eingabe-Module
- Eigenschaften
- Aufbau
- Anschluss- und Prinzipschaltbild
- Technische Daten

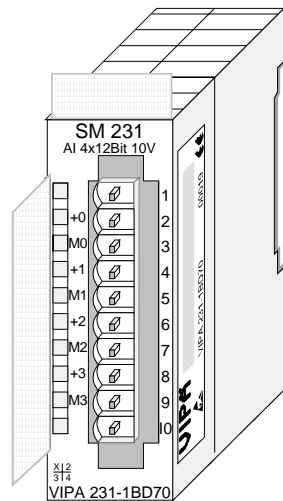
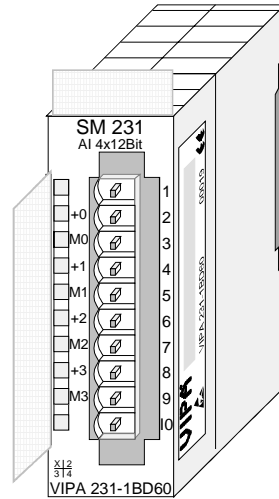
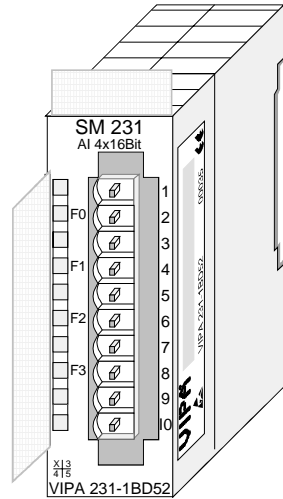
### Inhalt

Thema	Seite
<b>Teil 16 Analoge Eingabe-Module</b> .....	<b>16-1</b>
Systemübersicht .....	16-2
Allgemeines .....	16-4
AI 4x16Bit, Multi-Input.....	16-5
AI 4x12Bit, 4 ... 20mA, potenzialgetrennt.....	16-16
AI 4x12Bit, $\pm 10V$ , potenzialgetrennt.....	16-19
AI 4x16Bit f .....	16-22
AI 8x16Bit .....	16-32

# Systemübersicht

## Eingabe-Module SM 231

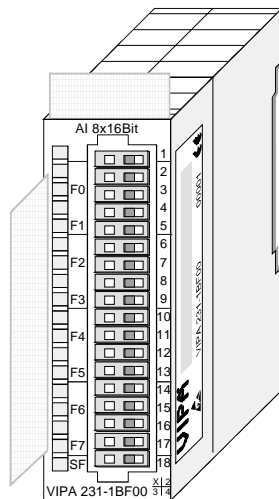
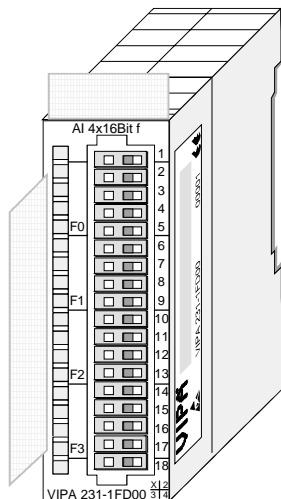
Nachfolgend finden Sie eine Übersicht der bei VIPA zur Zeit erhältlichen analogen Eingabe-Module:



### Bestelldaten Eingabe-Module

Typ	Bestellnummer	Seite
AI4x16Bit, Multiinput	VIPA 231-1BD52	16-5
AI4x12Bit, 4 ... 20mA, potenzialgetrennt	VIPA 231-1BD60	16-16
AI4x12Bit, ±10V, potenzialgetrennt	VIPA 231-1BD70	16-19

**Eingabe-Module  
SM 231**



**Bestelldaten  
Eingabe-Module**

Typ	Bestellnummer	Seite
AI4x16Bit f	VIPA 231-1FD00	16-22
AI8x16Bit	VIPA 231-1BF00	16-32

## Allgemeines

### Leitungen für Analogsignale

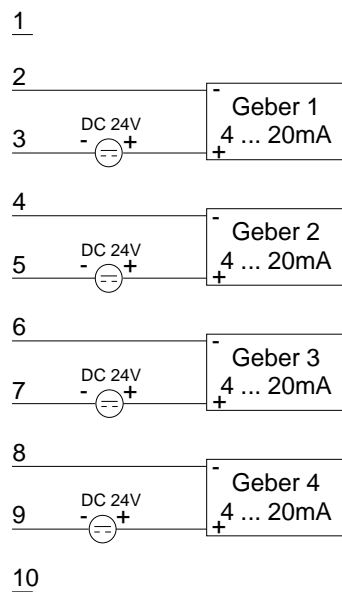
Für die Analogsignale sollten Sie geschirmte und paarweise verdrehte Leitungen verwenden. Hierdurch verringern Sie die Störbeeinflussung. Den Schirm der Analogleitungen sollten Sie an beiden Leitungsenden erden. Wenn Potenzialunterschiede zwischen den Leitungsenden bestehen, kann ein Potenzialausgleichstrom fließen, der die Analogsignale stören könnte. In diesem Fall sollten Sie den Schirm nur an einem Leitungsende erden.

### Anschließen von Messwertgebern

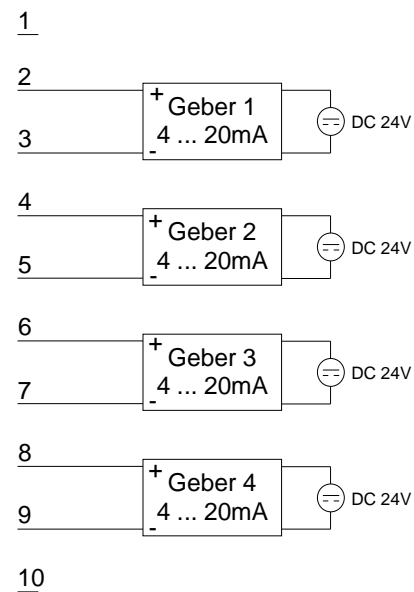
Die analogen Eingabemodule bieten vielfältige Anschlussmöglichkeiten für 2-Draht- und 4-Draht-Messumformer.

Bitte beachten Sie, dass die Messumformer extern zu versorgen sind. Schleifen Sie bei 2-Draht-Messumformern eine externe Spannungsversorgung in Ihre Messleitung ein. Folgende Abbildung soll den Anschluss von 2- und 4-Draht-Messumformern verdeutlichen:

#### 2-Draht-Anschluss



#### 4-Draht-Anschluss



#### Hinweis!

Bitte achten Sie beim Anschluss der Messumformer auf richtige Polarität! Schließen Sie nicht benutzte Eingänge kurz, indem Sie den positiven Anschluss und Kanal-Masse des jeweiligen Kanals miteinander verbinden.

### Parametrierung und Diagnose zur Laufzeit

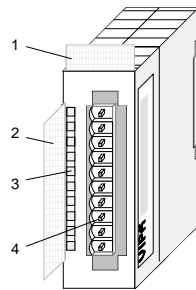
Durch Einsatz der SFCs 55, 56 und 57 können Sie zur Laufzeit über die CPU 21x die Parameter in Ihren Analog-Modulen ändern.

Zur Diagnoseauswertung zur Laufzeit stehen Ihnen die beiden SFCs 51 und 59 zur Verfügung. Hiermit können Sie detaillierte Diagnoseinformationen abrufen und auf die Diagnose reagieren.

## AI 4x16Bit, Multi-Input

<b>Bestelldaten</b>	AI 4x16Bit Multi-Input	VIPA 231-1BD52
<b>Beschreibung</b>	<p>Das Modul besitzt 4 Eingänge, deren Funktion einzeln parametrierbar sind. Im Prozessabbild belegt das Modul insgesamt 8Byte Eingangsdaten (2Byte pro Kanal).</p> <p>Die Kanäle auf dem Modul sind gegenüber dem Rückwandbus mittels DC/DC-Wandlern und Optokopplern galvanisch getrennt.</p>	
<b>Eigenschaften</b>	<ul style="list-style-type: none"> <li>• die Kanäle können unterschiedlich parametrierbar werden und sind abschaltbar</li> <li>• die Massen der Kanäle sind nicht galvanisch verbunden und dürfen bis zu 5V Spannungsdifferenz erreichen</li> <li>• LED für Drahtbruchererkennung und Überstrom im Strommessbereich</li> <li>• Diagnosefunktion</li> </ul>	

### Aufbau



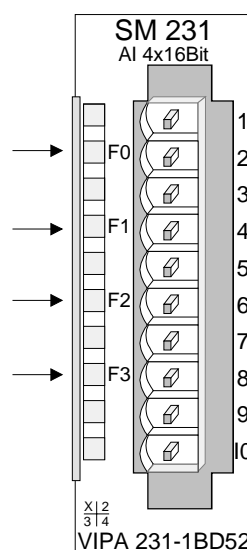
- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bitadresse mit Beschreibung
- [3] LEDs
- [4] Steckerleiste

### Statusanzeige Steckerbelegung

#### LED Beschreibung

F0 ... F3 LED (rot):  
leuchtet bei Drahtbruch in den 4...20mA Strommessbereichen

blinkt ab einem Strom > 40mA in allen Strommessbereichen



#### Pin Belegung

1	bei Vierleiteranschluss Kanal 0
2	+ Kanal 0
3	Masse Kanal 0
4	+ Kanal 1
5	Masse Kanal 1
6	+ Kanal 2
7	Masse Kanal 2
8	+ Kanal 3
9	Masse Kanal 3
10	bei Vierleiteranschluss Kanal 2

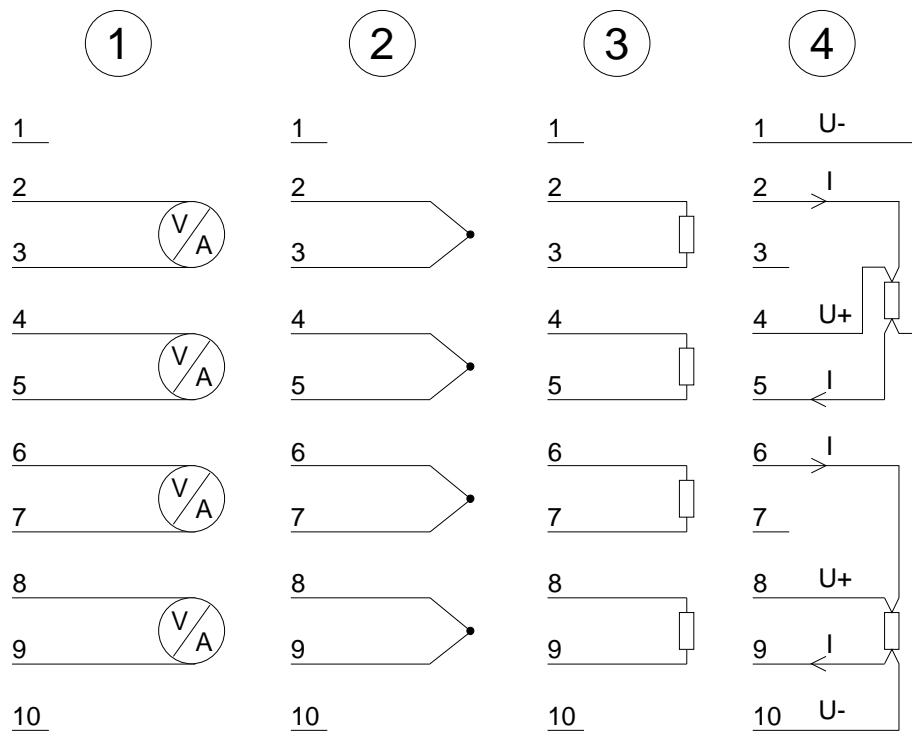
**Anschlussbilder**

Die nachfolgende Abbildung zeigt die Anschlussmöglichkeiten für die verschiedenen Messbereiche. Die Zuordnung zu den Messbereichen entnehmen Sie bitte der Spalte "Anschl." in der Tabelle "Funktions-Nr. Zuordnung" auf den Folgeseiten.



**Hinweis!**

Bitte beachten Sie, dass das Modul 231-1BD52 eine Weiterentwicklung des 231-1BD50 ist. Die Messfunktionen beginnen nun nicht mehr mit 00h, sondern verschieben sich um eins auf 01h.



**Achtung!**

Vorübergehend nicht benutzte Eingänge sind bei aktiviertem Kanal mit der zugehörigen Masse zu verbinden. Sind nicht benutzte Kanäle über FFh abgeschaltet, ist diese Maßnahme nicht erforderlich.



**Hinweis!**

Jeder Kanal ist einzeln parametrierbar. Für die Parametrierung stehen 10Byte Parametrierdaten zur Verfügung. Sie werden permanent abgelegt und bleiben auch im ausgeschalteten Zustand erhalten.



**Funktions-Nr.  
Zuordnung**

Die Zuweisung einer Funktions-Nr. zu einem Kanal erfolgt über die Parametrierung. Mit der Funktions-Nr. 00h wird die in den permanent abgelegten Parametrierdaten enthaltene Funktions-Nr. nicht beeinflusst.

Durch Angabe von FFh können Sie den entsprechenden Kanal deaktivieren.

Nr.	Funktion	Messbereich / Darstellung	Messtoleranz	Anschl.
00h	Die Funktions-Nr. in den permanent abgelegten Parametrierdaten wird nicht geändert.			
01h	Pt100 im Zweileiteranschluss	-200 .. +850°C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)3)</sup> ±1°C	(3)
02h	Pt1000 im Zweileiteranschluss	-200 .. +500°C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)3)</sup> ±1°C	(3)
03h	NI100 im Zweileiteranschluss	-50 .. +250°C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)3)</sup> ±1°C	(3)
04h	NI1000 im Zweileiteranschluss	-50 .. +250°C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)3)</sup> ±1°C	(3)
05h	Widerstandsmessung 60Ohm Zweileiter	- / 60Ω= Endwert (32767)	<sup>1)2)3)</sup> ±0,2% vom Endwert	(3)
06h	Widerstandsmessung 600Ohm Zweileiter	- / 600Ω = Endwert (32767)	<sup>1)2)3)</sup> ±0,1% vom Endwert	(3)
07h	Widerstandsmessung 3000Ohm Zweileiter	- / 3000Ω = Endwert (32767)	<sup>1)2)3)</sup> ±0,1% vom Endwert	(3)
08h	Widerstandsmessung 6000Ohm Zweileiter	- / 6000Ω = Endwert (32767)	<sup>1)2)3)</sup> ±0,1% vom Endwert	(3)
09h	Pt100 im Vierleiteranschluss	-200 .. +850°C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)</sup> ±0,5°C	(4)
0Ah	Pt1000 im Vierleiteranschluss	-200 .. +500°C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)</sup> ±0,5°C	(4)
0Bh	NI100 im Vierleiteranschluss	-50 .. +250°C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)</sup> ±0,5°C	(4)
0Ch	NI1000 im Vierleiteranschluss	-50 .. +250°C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)</sup> ±0,5°C	(4)
0Dh	Widerstandsmessung 60Ohm Vierleiter	- / 60Ω= Endwert (32767)	<sup>1)2)</sup> ±0,1% vom Endwert	(4)
0Eh	Widerstandsmessung 600Ohm Vierleiter	- / 600Ω= Endwert (32767)	<sup>1)2)</sup> ±0,05% vom Endwert	(4)
0Fh	Widerstandsmessung 3000Ohm Vierleiter	- / 3000Ω = Endwert (32767)	<sup>1)2)</sup> ±0,05% vom Endwert	(4)
10h	Thermoelement Typ J , Kompensation extern	-210 °C .. 850 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)4)</sup> ±1°C	(2)

Fortsetzung ...

... Fortsetzung

11h	Thermoelement Typ K, Kompensation extern	-270 °C .. 1200 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)4)</sup> ±1,5°C	(2)
12h	Thermoelement Typ N, Kompensation extern	-200 °C .. 1300 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)4)</sup> ±1,5°C	(2)
13h	Thermoelement Typ R, Kompensation extern	-50 °C .. 1760 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)4)</sup> ±4°C	(2)
14h	Thermoelement Typ T, Kompensation extern	-270 °C .. 400 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)4)</sup> ±1,5°C	(2)
15h	Thermoelement Typ S, Kompensation extern	-50 °C .. 1760 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)4)</sup> ±5°C	(2)
18h	Thermoelement Typ J, Kompensation intern	-210 °C .. 850 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)5)</sup> ±1,5°C	(2)
19h	Thermoelement Typ K, Kompensation intern	-270 °C .. 1200 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)5)</sup> ±2°C	(2)
1Ah	Thermoelement Typ N, Kompensation intern	-200 °C .. 1300 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)5)</sup> ±2°C	(2)
1Bh	Thermoelement Typ R, Kompensation intern	-50 °C .. 1760 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)5)</sup> ±5°C	(2)
1Ch	Thermoelement Typ T, Kompensation intern	-270 °C .. 400 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)5)</sup> ±2°C	(2)
1Dh	Thermoelement Typ S, Kompensation intern	-50 °C .. 1760 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)4)</sup> ±5°C	(2)
27h	Spannung 0...50mV S7-Format von Siemens	0...50mV / 59,25mV = Ende des nutzbaren Bereichs im Übersteuerungsbereich (32767) 0...50mV = Nennwert (0...27648) Zweierkomplement	<sup>1)</sup> ±0,1% vom Endwert	(1)
28h	Spannung ±10V S7-Format von Siemens	±11,85V / 11,85V= Ende Übersteuerungsbereich (32767) -10...10V= Nennbereich (-27648...27648) -11,85V= Ende Untersteuerungsbereich (-32767) Zweierkomplement	<sup>1)</sup> ±0,05% vom Endwert	(1)
29h	Spannung ±4V S7-Format von Siemens	±4,74V / 4,74V = Ende Übersteuerungsbereich (32767) -4...4V = Nennbereich (-27648...27648) -4,74V = Ende Untersteuerungsbereich (-32767) Zweierkomplement	<sup>1)</sup> ±0,05% vom Endwert	(1)
2Ah	Spannung ±400mV S7-Format von Siemens	±0,474V / 474mV = Ende Übersteuerungsbereich (32767) -400...400mV = Nennbereich (-27648...27648) -474mV = Ende Untersteuerungsbereich (-32767) Zweierkomplement	<sup>1)</sup> ±0,1% vom Endwert	(1)
2Bh	Spannung ±10V S5-Format von Siemens	±11,85V / 12,5V = Ende Übersteuerungsbereich (20480) -10...10V = Nennbereich (-16384...16384) -12,5V = Ende Untersteuerungsbereich (-20480) Betrag und Vorzeichen	<sup>1)</sup> ±0,2% vom Endwert	(1)

Fortsetzung ...

... Fortsetzung

2Ch	Strom $\pm 20\text{mA}$ S7-Format von Siemens	$\pm 23,70\text{mA}$ / 23,70mA = Ende Übersteuerungsbereich (32767) -20...20mA = Nennwert (-27648...27648) -23,70mA = Ende Untersteuerungsbereich (-32767) Zweierkomplement	<sup>1)</sup> $\pm 0,05\%$ vom Endwert	(1)
2Dh	Strom 4...20mA S7-Format von Siemens	1,185 .. +22,96mA / 22,96mA = Ende Übersteuerungsbereich (32767) 4...20mA = Nennbereich (0...27648) 0mA = Ende Untersteuerungsbereich (-5530) Zweierkomplement	<sup>1)</sup> $\pm 0,05\%$ vom Endwert	(1)
2Eh	Strom 4...20mA S5-Format von Siemens	1,185 .. +22,96mA / 22,96mA = Ende Übersteuerungsbereich (20480) 20mA = Nennbereich (0...16384) 0mA = Ende Untersteuerungsbereich (-4096) Betrag und Vorzeichen	<sup>1)</sup> $\pm 0,2\%$ vom Endwert	(1)
2Fh	Strom $\pm 20\text{mA}$ S5-Format von Siemens	$\pm 23,70\text{mA}$ / 23,70mA = Ende Übersteuerungsbereich (19456) -20...20mA = Nennwert (-16384...16384) -23,70mA = Ende Untersteuerungsbereich (-19456) Betrag und Vorzeichen	<sup>1)</sup> $\pm 0,05\%$ vom Endwert	(1)
32h	Widerstandsmessung 6000Ohm Vierleiter	- / 6000 $\Omega$ = Endwert (32767)	<sup>1)2)</sup> $\pm 0,05\%$ vom Endwert	(4)
33h	Widerstandsmessung 6000Ohm Vierleiter	- / 6000 $\Omega$ = Endwert (6000)	<sup>1)2)</sup> $\pm 0,05\%$ vom Endwert	(4)
35h	Widerstandsmessung 60Ohm Zweileiter	- / 60 $\Omega$ = Endwert (6000)	<sup>1)2)3)</sup> $\pm 0,2\%$ vom Endwert	(3)
36h	Widerstandsmessung 600Ohm Zweileiter	- / 600 $\Omega$ = Endwert (6000)	<sup>1)2)3)</sup> $\pm 0,1\%$ vom Endwert	(3)
37h	Widerstandsmessung 3000Ohm Zweileiter	- / 3000 $\Omega$ = Endwert (30000)	<sup>1)2)3)</sup> $\pm 0,1\%$ vom Endwert	(3)
38h	Widerstandsmessung 6000Ohm Zweileiter	- / 6000 $\Omega$ = Endwert (6000)	<sup>1)2)3)</sup> $\pm 0,1\%$ vom Endwert	(3)
<sup>6)</sup> 3Ah	Strom $\pm 20\text{mA}$ S5-Format von Siemens	$\pm 23,70\text{mA}$ / 23,70mA = Ende Übersteuerungsbereich (19456) -20...20mA = Nennwert (-16384...16384) -23,70mA = Ende Untersteuerungsbereich (-19456) Zweierkomplement	<sup>1)</sup> $\pm 0,05\%$ vom Endwert	(1)
<sup>6)</sup> 3Bh	Spannung $\pm 10\text{V}$ S5-Format von Siemens	$\pm 11,85\text{V}$ / 12,5V = Ende Übersteuerungsbereich (20480) -10...10V = Nennbereich (-16384...16384) -12,5V = Ende Untersteuerungsbereich (-20480) Zweierkomplement	<sup>1)</sup> $\pm 0,2\%$ vom Endwert	(1)
3Dh	Widerstandsmessung 60Ohm Vierleiter	- / 60 $\Omega$ = Endwert (6000)	<sup>1)2)</sup> $\pm 0,1\%$ vom Endwert	(4)
3Eh	Widerstandsmessung 600Ohm Vierleiter	- / 600 $\Omega$ = Endwert (6000)	<sup>1)2)</sup> $\pm 0,05\%$ vom Endwert	(4)
3Fh	Widerstandsmessung 3000Ohm Vierleiter	- / 3000 $\Omega$ = Endwert (30000)	<sup>1)2)</sup> $\pm 0,05\%$ vom Endwert	(4)

Fortsetzung ...

... Fortsetzung

57h	Spannung 0...50mV	0...50mV / 59.25mV = Ende Übersteuerungsbereich (5925) 0...50mV = Nennbereich (0...5000) Zweierkomplement	<sup>1)</sup> ±0,1% vom Endwert	(1)
58h	Spannung ±10V	±11,85V / 11,85V= Ende Übersteuerungsbereich (11850) -10...10V= Nennbereich (-10000...10000) -11,85V= Ende Untersteuerungsbereich (-11850) Zweierkomplement	<sup>1)</sup> ±0,05% vom Endwert	(1)
59h	Spannung ±4V	±4,74V / 4,74V = Ende Übersteuerungsbereich (47400) -4...4V = Nennbereich (-40000...40000) -4,74V = Ende Untersteuerungsbereich (-47400) Zweierkomplement	<sup>1)</sup> ±0,05% vom Endwert	(1)
5Ah	Spannung ±400mV	±0,474V / 474mV = Ende Übersteuerungsbereich (47400) -400...400mV = Nennbereich (-40000...40000) -474mV = Ende Untersteuerungsbereich (-47400) Zweierkomplement	<sup>1)</sup> ±0,1% vom Endwert	(1)
5Ch	Strom ±20mA	±23,70mA / 23,70mA = Ende Übersteuerungsbereich (23700) -20...20mA = Nennwert (-20000...20000) -23,70mA = Ende Untersteuerungsbereich (-23700) Zweierkomplement	<sup>1)</sup> ±0,05% vom Endwert	(1)
5Dh	Strom 4...20mA	1,185 .. +22,96mA / 22,96mA = Ende Übersteuerungsbereich (18960) 4...20mA = Nennbereich (0...16000) 0mA = Ende Untersteuerungsbereich (-4000) Zweierkomplement	<sup>1)</sup> ±0,05% vom Endwert	(1)
FFh	Kanal nicht aktiv (abgeschaltet)			

<sup>1)</sup> ermittelt bei Umgebungstemperatur 25°C, Geschwindigkeit 15 Wandlungen/s

<sup>2)</sup> ausgenommen sind Fehler durch Ungenauigkeit des Gebers

<sup>3)</sup> ausgenommen sind Fehler durch Übergangswiderstände an Kontakten sowie Leitungswiderstände

<sup>4)</sup> die Kompensation der Kaltstelle muss extern durchgeführt werden

<sup>5)</sup> die Kompensation der Kaltstelle wird intern durchgeführt indem die Temperatur des Frontsteckers berücksichtigt wird, die Thermoelementleiter sind unmittelbar am Frontstecker anzuschließen, ggf. muss mit Thermoelement-Verlängerungskabel verlängert werden.

<sup>6)</sup> Verfügbar ab Hardware-Ausgabestand 11



### Hinweis!

Das Modul ist ab Werk auf den Messbereich "Spannung ±10V" eingestellt.

**Zahlendarstellung im S5-Format von Siemens**

Die Eingabedaten werden im S5-Format von Siemens in einem Wort abgelegt. Das Wort setzt sich zusammen aus dem binären Wert und den Informationsbits.

*Zahlendarstellung:*

Byte	Bit 7 ... Bit 0
0	Bit 0: Überlaufbit 0: Wert liegt im Messbereich 1: Messbereich überschritten Bit 1: Fehlerbit (gesetzt bei internem Fehler) Bit 2: Tätigkeitsbit (immer 0) Bit 3 ... 7: Binärer Messwert
1	Bit 0 ... 6: Binärer Messwert Bit 7: Vorzeichen 0 positiv 1 negativ

**+/- 10V (Zweierkomplement)**

Spannung	Dezimal	Hex
-10V	-16384	C000
-5V	-8192	E000
0V	0	0000
+5V	+8192	2000
+10V	+16384	4000

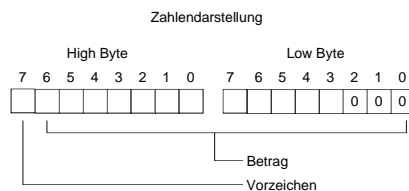
Formeln für die Berechnung:

$$Wert = 16384 \cdot \frac{U}{10}, \quad U = Wert \cdot \frac{10}{16384}$$

U: Spannungswert, Wert: Dezimalwert

**+/- 10V (Betrag und Vorzeichen)**

Spannung	Dezimal	Hex
-10V	-16384	C000
-5V	-8192	A000
0V	0	0000
+5V	+8192	2000
+10V	+16384	4000



**4....20mA (Betrag und Vorzeichen)**

Strom	Dezimal	Hex
+4mA	0	0000
+12mA	+8192	2000
+20mA	+16384	4000



**+/- 20mA (Zweierkomplement)**

Strom	Dezimal	Hex
-20mA	-16384	C000
-10mA	-8192	E000
0mA	0	0000
+10mA	+8192	2000
+20mA	+16384	4000

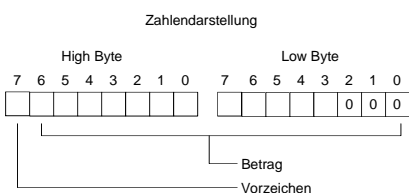
Formeln für die Berechnung:

$$Wert = 16384 \cdot \frac{I}{20}, \quad I = Wert \cdot \frac{20}{16384}$$

I: Stromwert, Wert: Dezimalwert

**+/- 20mA (Betrag und Vorzeichen)**

Strom	Dezimal	Hex
-20mA	-16384	C000
-10mA	-8192	A000
0mA	0	0000
+10mA	+8192	2000
+20mA	+16384	4000



**Zahlendarstellung im S7-Format von Siemens**

Die Darstellung des Analogwertes erfolgt im Zweierkomplement.

Zahlendarstellung:

Byte	Bit 7 ... Bit 0
0	Bit 0 ... 7: Binärer Messwert
1	Bit 0 ... 6: Binärer Messwert Bit 7: Vorzeichen 0 positiv 1 negativ

**+/- 10V**

Spannung	Dezimal	Hex
-10V	-27648	9400
-5V	-13824	CA00
0V	0	0
+5V	13824	3600
+10V	+27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U}{10}, \quad U = Wert \cdot \frac{10}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**0...10V**

Spannung	Dezimal	Hex
0V	0	0000
5V	8192	2000
10V	16384	4000

Formeln für die Berechnung:

$$Wert = 16384 \cdot \frac{U}{10}, \quad U = Wert \cdot \frac{10}{16384}$$

U: Spannungswert, Wert: Dezimalwert

**1...5V**

Spannung	Dezimal	Hex
+1V	0	0
+3V	+13824	3600
+5V	+27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U - 1}{4}, \quad U = Wert \cdot \frac{4}{27648} + 1$$

U: Spannungswert, Wert: Dezimalwert

**+/-4V**

Spannung	Dezimal	Hex
-4V	-27648	9400
0V	0	0
4V	27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U}{4}, \quad U = Wert \cdot \frac{4}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**+/-400mV**

Spannung	Dezimal	Hex
-400mV	-27648	9400
0V	0	0
400mV	27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U}{400}, \quad U = Wert \cdot \frac{400}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**4...20mA**

Strom	Dezimal	Hex
+4mA	0	0
+12mA	+13824	3600
+20mA	+27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{I - 4}{16}, \quad I = Wert \cdot \frac{16}{27648} + 4$$

I: Stromwert, Wert: Dezimalwert

**+/- 20mA**

Strom	Dezimal	Hex
-20mA	-27648	9400
-10mA	-13824	CA00
0mA	0	0
+10mA	+13824	3600
+20mA	+27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{I}{20}, \quad I = Wert \cdot \frac{20}{27648}$$

I: Stromwert, Wert: Dezimalwert

**Messdaten erfassen**

Während der Messung werden die Messdaten im Dateneingabebereich abgelegt. Die Zuordnung der Messdaten zu einem Messwert und die jeweiligen Toleranzen finden Sie in der Tabelle oben.

Die nachfolgende Tabelle zeigt den Aufbau des Dateneingabebereichs:

*Dateneingabebereich:*

Byte	Bit 7 ... Bit 0
0	High-Byte Kanal 0
1	Low-Byte Kanal 0
2	High-Byte Kanal 1
3	Low-Byte Kanal 1
4	High-Byte Kanal 2
5	Low-Byte Kanal 2
6	High-Byte Kanal 3
7	Low-Byte Kanal 3

**Hinweis!**

Bei Vierleitermessung werden nur die Kanäle 0 und 2 verwendet.

**Parametrierdaten**

Jeder Kanal ist einzeln parametrierbar. Für die Parametrierung stehen 10Byte Parametrierdaten zur Verfügung. Sie werden permanent abgelegt und bleiben auch im ausgeschalteten Zustand erhalten.

Die nachfolgende Tabelle zeigt den Aufbau des Parameterbereichs:

*Parameterbereich:*

Byte	Bit 7 ... Bit 0	Default
0	Diagnosealarm-Byte: Bit 0 ... 5: reserviert Bit 6: 0: Diagnosealarm gesperrt 1: Diagnosealarm freigegeben Bit 7: reserviert	00h
1	reserviert	00h
2	Funktions-Nr. Kanal 0 (siehe Tabelle)	2Dh
3	Funktions-Nr. Kanal 1 (siehe Tabelle)	2Dh
4	Funktions-Nr. Kanal 2 (siehe Tabelle)	2Dh
5	Funktions-Nr. Kanal 3 (siehe Tabelle)	2Dh
6	Option-Byte Kanal 0	00h
7	Option-Byte Kanal 1	00h
8	Option-Byte Kanal 2	00h
9	Option-Byte Kanal 3	00h

**Parameter**

*Diagnosealarm*

Mit Bit 6 von Byte 0 können Sie den Diagnosealarm freigeben. Im Fehlerfall werden an Ihr übergeordnetes Master-System 4 Diagnose-Bytes geschickt.

*Funktions-Nr.*

Tragen Sie hier für jeden Kanal die Funktions-Nummer Ihrer Messfunktion ein. Die Zuordnung der Funktions-Nummer zu einer Messfunktion entnehmen Sie bitte der obigen Tabelle.

*Option-Byte*

Hier können Sie für jeden Kanal die Wandlergeschwindigkeit einstellen. Des Weiteren sind Selektions- und Hüllfunktion implementiert.



**Hinweis!**

Bitte beachten Sie, dass bei höheren Wandlergeschwindigkeiten die Auflösung aufgrund der kürzeren Integrationszeit sinkt.

Das Datenübergabeformat bleibt gleich. Lediglich die unteren Bits (LSBs) sind für den Analogwert nicht mehr aussagekräftig.

*Aufbau Option-Byte:*

Byte	Bit 7 ... Bit 0	Auflösung	Default
6 ... 9	Option-Byte: Bit 0 ... 3: Geschwindigkeit* 0000 15 Wandlungen/s 0001 30 Wandlungen/s 0010 60 Wandlungen/s 0011 123 Wandlungen/s 0100 168 Wandlungen/s 0101 202 Wandlungen/s 0110 3,7 Wandlungen/s 0111 7,5 Wandlungen/s Bit 4 ... 5: Selektionsfunktion 00 deaktiviert 01 2 aus 3 Werten verwenden 10 4 aus 6 Werten verwenden Bit 6 ... 7: Hüllfunktion 00 deaktiviert 01 Hülle ± 8 10 Hülle ±16	16 16 15 14 12 10 16 16	00h

\*) Die Angaben beziehen sich auf 1-Kanal-Betrieb. Wenn Sie die angegebenen Wandlergeschwindigkeiten durch die Anzahl der aktiven Kanäle dividieren, erhalten Sie bei Mehrkanalbetrieb die Wandlergeschwindigkeit pro Kanal.



**Diagnosedaten**

Sobald Sie die Alarmfreigabe in Byte 0 des Parameterbereichs aktiviert haben, werden im Fehlerfall 4 Diagnose-Bytes mit fester Belegung an das übergeordnete System übergeben. Bitte beachten Sie, dass für die Diagnose nur die ersten zwei Bytes verwendet werden. Die restlichen zwei Bytes werden nicht benutzt. Die Diagnose-Bytes haben folgende Belegung:

*Diagnosedaten:*

Byte	Bit 7 ... Bit 0	Default
0	Bit 0: Störung im Modul Bit 1: konstant 0 Bit 2: Fehler extern Bit 3: Kanalfehler vorhanden Bit 4 ... 7: reserviert	-
1	Bit 0 ... 3 Modulkategorie 0101 Analogmodul Bit 4: Kanalinformation vorhanden	-
2 ... 3	nicht belegt	-

**Technische Daten**

Elektrische Daten	VIPA 231-1BD52
Anzahl der Eingänge	4 differenzielle Eingänge
Eingangswiderstand	> 2M $\Omega$ (Spannungsbereich) < 50 $\Omega$ (Strombereich)
Eingangsbereiche	
- Thermoelemente	Typ J, K, N, R, S, T
- Widerstandsthermometer	Pt100, Pt1000, NI100, NI1000
- Widerstandsmessung	60 $\Omega$ , 600 $\Omega$ , 3k $\Omega$
- Spannungsmessung	0...50mV, 0...10V, $\pm$ 4mV, $\pm$ 4V, $\pm$ 10V
- Strommessung	4...20mA, $\pm$ 20mA
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme	240mA über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	8Byte (1 Wort pro Kanal)
Ausgabedaten	-
Parameterdaten	10Byte
Diagnosedaten	4Byte
Prozessalarmdaten	-
Maße und Gewicht	
Abmessungen (BxHxT)	25,4 x 76 x 76 mm
Gewicht	100g

## AI 4x12Bit, 4 ... 20mA, potenzialgetrennt

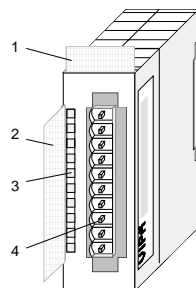
**Bestelldaten** AI 4x12Bit, 4...20mA, potenzialgetrennt VIPA 231-1BD60

**Beschreibung** Das Modul besitzt 4 Eingänge, die fest auf Strommessung (4 ... 20mA) eingestellt sind. Im Prozessabbild belegt das Modul insgesamt 8Byte Eingangsdaten (2Byte pro Kanal).

Die Messwerte werden im S5-Format von Siemens ausgegeben. Die Kanäle auf dem Modul sind mittels DC/DC-Wandlern und Trennverstärkern sowohl zum Rückwandbus als auch untereinander potenzialgetrennt.

- Eigenschaften**
- 4 Eingänge, deren Kanäle zum Rückwandbus und untereinander potenzialgetrennt sind (galvanische Trennung der Kanäle durch Trennverstärker)
  - Strommessung fest eingestellt
  - keine Parametrierung erforderlich
  - Geeignet für Geber mit 4 ... 20mA
  - LEDs für Drahtbrucherkennung

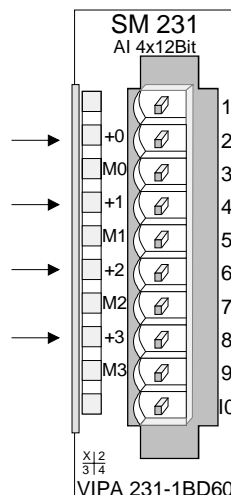
**Aufbau**



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

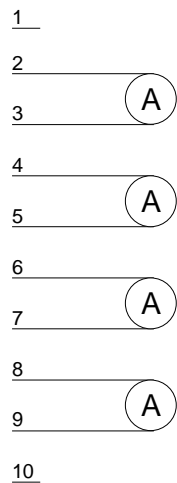
LED	Beschreibung
+0 ... +3	LED (rot) Drahtbrucherkennung LED leuchtet bei Drahtbruch bzw. wenn kein Geber angeschlossen ist.



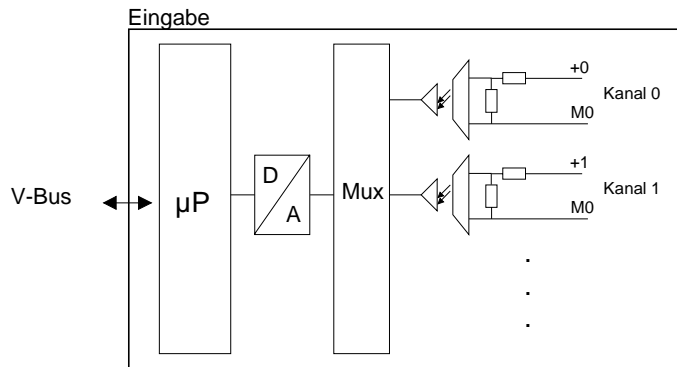
Pin	Belegung
1	
2	pos. Anschluss K.0
3	Masse Kanal 0
4	pos. Anschluss K.1
5	Masse Kanal 1
6	pos. Anschluss K.2
7	Masse Kanal 2
8	pos. Anschluss K.3
9	Masse Kanal 3
10	

**Anschlussbild und Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Drahtbruchkennung**

Die Drahtbruchkennung ist immer aktiv. Im Falle eines Drahtbruchs bzw. wenn kein Geber angeschlossen ist leuchtet die LED des entsprechenden Kanals. Das Modul ist nicht diagnosefähig.

**Zahlendarstellung**

Die Eingabedaten werden im S5-Format von Siemens in einem Wort abgelegt. Das Wort setzt sich zusammen aus dem binären Wert und Informationsbits:

*Zahlendarstellung:*

Byte	Bit 7 ... Bit 0
0	Bit 0: Überlaufbit 0: Wert liegt im Messbereich 1: Messbereich überschritten Bit 1: Fehlerbit (gesetzt bei internem Fehler) Bit 2: Tätigkeitsbit (immer 0) Bit 3 ... 7: <b>Binärer Messwert</b> (siehe nachfolgende Tabelle)
1	Bit 0 ... 6: <b>Binärer Messwert</b> (siehe nachfolgende Tabelle) Bit 7: Vorzeichen 0 positiv 1 negativ

In der nachfolgend aufgeführten Tabelle finden Sie die Zuordnung der Binärwerte zu den entsprechenden Messwerten.

**Zahlendarstellung im S5-Format von Siemens**

Messwert in mA	Einheiten	Binärer Messwert	T	F	Ü	Bereich
24,0	2560	0 1 0 1 0 0 0 0 0 0 0 0 0 0	0	0	0	Übersteuerungsbereich
20,016	2049	0 1 0 0 0 0 0 0 0 0 0 0 0 1	0	0	0	
20,0	2048	0 1 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0	Nennbereich
19,98	2047	0 0 1 1 1 1 1 1 1 1 1 1 1 1	0	0	0	
12,0	1024	0 0 1 0 0 0 0 0 0 0 0 0 0 0	0	0	0	
8,0	512	0 0 0 1 0 0 0 0 0 0 0 0 0 0	0	0	0	
6,0	256	0 0 0 0 1 0 0 0 0 0 0 0 0 0	0	0	0	
5,0	128	0 0 0 0 0 1 0 0 0 0 0 0 0 0	0	0	0	
4,016	2	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	0	0	0	
4,008	1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	0	0	0	
4	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0	
3,984	-2	1 1 1 1 1 1 1 1 1 1 1 1 1 1 0	0	0	0	Untersteuerungsbereich
3,0	-128	1 1 1 1 1 1 0 0 0 0 0 0 0 0	0	0	0	
2,0	-256	1 1 1 1 1 0 0 0 0 0 0 0 0 0	0	0	0	
1,0	-384	1 1 1 1 0 1 0 0 0 0 0 0 0 0	0	0	0	
0,0	-512	1 1 1 1 0 0 0 0 0 0 0 0 0 0	0	0	0	

**Technische Daten**

Elektrische Daten	VIPA 231-1BD60
Anzahl der Eingänge	4 einzeln potenzialgetrennt
Strombereich	4 ... 20mA
Eingangsfiler Zeitverzögerung	3ms
Eingangswiderstand	20Ω
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme	280mA über Rückwandbus
Potenzialtrennung	ja, jeder Kanal einzeln Isolation geprüft mit 500Veff
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	8Byte (1 Wort pro Kanal)
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Prozessalarmdaten	-
Maße und Gewicht	
Abmessungen (BxHxT)	25,4 x 76 x 76 mm
Gewicht	120g

## AI 4x12Bit, $\pm 10V$ , potenzialgetrennt

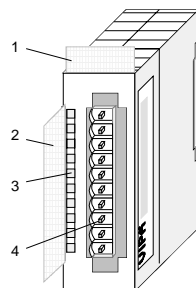
**Bestelldaten** AI 4x12Bit,  $\pm 10V$ , potenzialgetrennt VIPA 231-1BD70

**Beschreibung** Das Modul besitzt 4 Eingänge, die fest auf Spannungsmessung ( $\pm 10V$ ) eingestellt sind. Im Prozessabbild belegt das Modul insgesamt 8Byte Eingangsdaten (2Byte pro Kanal). Die Messwerte werden im S5-Format von Siemens ausgegeben. Die Kanäle auf dem Modul sind mittels DC/DC-Wandlern und Trennverstärkern sowohl zum Rückwandbus als auch untereinander potenzialgetrennt.

**Eigenschaften**

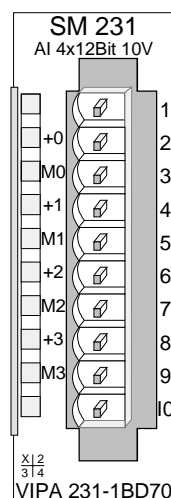
- 4 Eingänge, deren Kanäle zum Rückwandbus und untereinander potenzialgetrennt sind (galvanische Trennung der Kanäle durch Trennverstärker)
- Spannungsmessung fest eingestellt
- keine Parametrierung erforderlich
- Geeignet für Geber mit  $\pm 10V$

### Aufbau



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

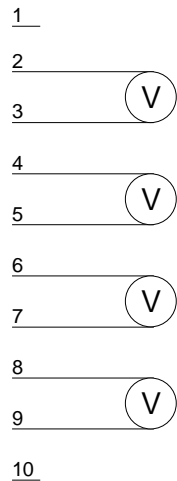
### Steckerbelegung



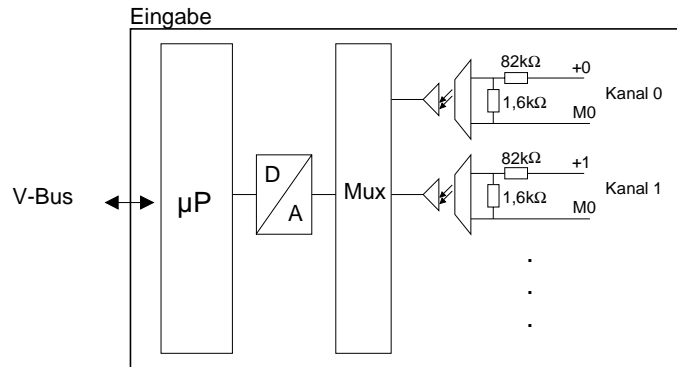
Pin	Belegung
1	
2	pos. Anschluss Kanal 0
3	Masse Kanal 0
4	pos. Anschluss Kanal 1
5	Masse Kanal 1
6	pos. Anschluss Kanal 2
7	Masse Kanal 2
8	pos. Anschluss Kanal 3
9	Masse Kanal 3
10	

**Anschlussbild und Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Zahlendarstellung**

Die Eingabedaten werden im S5-Format von Siemens in einem Wort abgelegt. Das Wort setzt sich zusammen aus dem binären Wert und Informationsbits:

*Zahlendarstellung:*

Byte	Bit 7 ... Bit 0
0	Bit 0: Überlaufbit 0: Wert liegt im Messbereich 1: Messbereich überschritten Bit 1: Fehlerbit (gesetzt bei internem Fehler) Bit 2: Tätigkeitsbit (immer 0) Bit 3 ... 7: <b>Binärer Messwert</b> (siehe nachfolgende Tabelle)
1	Bit 0 ... 6: <b>Binärer Messwert</b> (siehe nachfolgende Tabelle) Bit 7: Vorzeichen 0 positiv 1 negativ

In der nachfolgend aufgeführten Tabelle finden Sie die Zuordnung der Binärwerte zu den entsprechenden Messwerten.

**Zahlendarstellung  
im S5-Format von  
Siemens**

Messwert in V	Ein- heiten	Binärer Messwert	T	F	Ü	Bereich
12,5	2560	0 1 0 1 0 0 0 0 0 0 0 0 0 0	0	0	0	Übersteuerungs- bereich
10,005	2049	0 1 0 0 0 0 0 0 0 0 0 0 0 1	0	0	0	
10,0	2048	0 1 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0	Nennbereich
5	1024	0 0 1 0 0 0 0 0 0 0 0 0 0 0	0	0	0	
2,5	512	0 0 0 1 0 0 0 0 0 0 0 0 0 0	0	0	0	
1,25	256	0 0 0 0 1 0 0 0 0 0 0 0 0 0	0	0	0	
0,625	128	0 0 0 0 0 1 0 0 0 0 0 0 0 0	0	0	0	
0,005	1	0 0 0 0 0 0 0 0 0 0 0 0 0 1	0	0	0	
0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0	
-0,005	-1	1 1 1 1 1 1 1 1 1 1 1 1 1 1	0	0	0	
-0,625	-128	1 1 1 1 1 1 0 0 0 0 0 0 0 0	0	0	0	
-1,25	-256	1 1 1 1 1 0 0 0 0 0 0 0 0 0	0	0	0	
-2,5	-512	1 1 1 1 0 0 0 0 0 0 0 0 0 0	0	0	0	
-5	-1024	1 1 1 0 0 0 0 0 0 0 0 0 0 0	0	0	0	
-10,0	-2048	1 1 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0	
-10,005	-2049	1 0 1 1 1 1 1 1 1 1 1 1 1 1	0	0	0	
-12	-2560	1 0 1 1 0 0 0 0 0 0 0 0 0 0	0	0	0	

**Technische Daten**

Elektrische Daten	VIPA 231-1BD70
Anzahl der Eingänge	4 einzeln potenzialgetrennt
Spannungsbereich	±10V
Eingangsfiler Zeitverzögerung	3ms
Eingangswiderstand	83,5kΩ
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme	300mA über Rückwandbus
Potenzialtrennung	ja, jeder Kanal einzeln Isolation geprüft mit 500Veff
Programmierdaten	
Eingabedaten	8Byte (1 Wort pro Kanal)
Ausgabedaten	-
Parameterdaten	-
Diagnosedaten	-
Prozessalarmdaten	-
Maße und Gewicht	
Abmessungen (BxHxT)	25,4 x 76 x 76 mm
Gewicht	120g

## AI 4x16Bit f

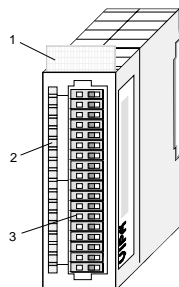
**Bestelldaten** AI 4x16Bit f VIPA 231-1FD00

**Beschreibung** Das Modul besitzt 4 schnelle (f=fast) Eingänge, deren Funktion einzeln parametrierbar sind. Im Prozessabbild belegt das Modul insgesamt 8Byte Eingangsdaten (2Byte pro Kanal).

Die Kanäle auf dem Modul sind gegenüber dem Rückwandbus mittels DC/DC-Wandlern und Optokopplern galvanisch getrennt.

- Eigenschaften**
- die Zykluszeit ist bei Einsatz aller 4 Kanäle < 1ms
  - die Kanäle können unterschiedlich parametrierbar werden und sind abschaltbar
  - LED für Drahtbruchererkennung und Überstrom im Strommessbereich
  - Diagnosefunktion
  - Auflösung 16Bit

### Aufbau



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] LED-Statusanzeige
- [3] Steckerleiste

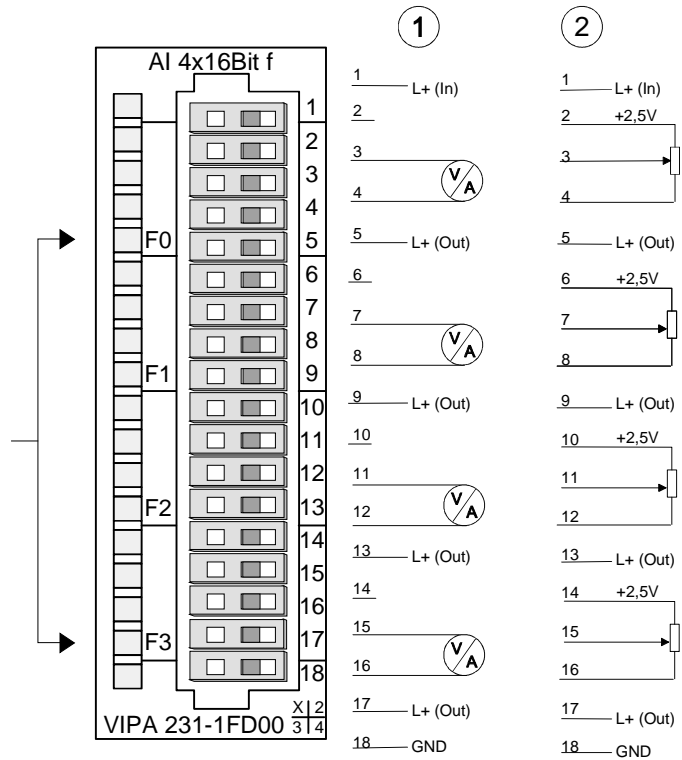


**Statusanzeige  
Steckerbelegung**

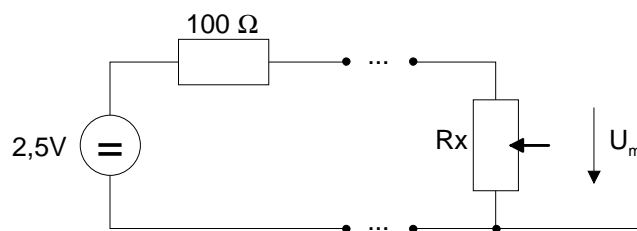
**LED Beschreibung**

F0 ... F3 LED (rot):  
leuchtet, wenn bei der Strommessung der Wert außerhalb vom Bereich 4...20mA liegt (Drahtbruch oder Überlast).

**Anschlussbild**



Der Innenwiderstand  $R_i$  der 2,5V Spannungsquelle beträgt  $100\Omega$ . Für den Anschluss eines Spannungsteilers ergibt sich folgendes Ersatzschaltbild:



**Hinweis!**

Vorübergehend nicht benutzte Eingänge sind bei aktiviertem Kanal mit der zugehörigen Masse zu verbinden. Sind nicht benutzte Kanäle über FFh abgeschaltet, ist diese Maßnahme nicht erforderlich.

**Funktions-Nr.  
Zuordnung**

Die Zuweisung einer Funktions-Nr. zu einem Kanal erfolgt über die Parametrierung. Mit der Funktions-Nr. 00h wird die in den permanent abgelegten Parametrierdaten enthaltene Funktions-Nr. nicht beeinflusst.

Durch Angabe von FFh können Sie den entsprechenden Kanal deaktivieren.

Nr.	Funktion	Messbereich / Darstellung	Messtoleranz	Anschl.
00h	keinen Einfluss auf die permanent abgelegten Parametrierdaten			
28h	Spannung $\pm 10V$ S7-Format von Siemens (Defaulteinstellung)	-10V ... -9,9V (-27648 ... -27371) <sup>1)</sup> -9,9 ... 9,9 (-27370 ... 27370) -10...10V= Nennbereich (-27648...27648) 9,9 ... 10V (27371 ... 27648) <sup>1)</sup>	0,2% vom Endwert	(1), (2)
29h	Spannung $\pm 4V$ S7-Format von Siemens	$\pm 4,70V$ / 4,70V = Ende Übersteuerungsbereich (32511) -4...4V = Nennbereich (-27648...27648) -4,70V = Ende Untersteuerungsbereich (-32512) Zweierkomplement	0,2% vom Endwert	(1), (2)
2Ah	Spannung $\pm 400mV$ S7-Format von Siemens	$\pm 0,47V$ / 470mV = Ende Übersteuerungsbereich (32511) -400...400mV = Nennbereich (-27648...27648) -470mV = Ende Untersteuerungsbereich (-32512) Zweierkomplement	0,4% vom Endwert	(1)
2Ch	Strom $\pm 20mA$ S7-Format von Siemens	$\pm 23,51mA$ / 23,51mA = Ende Übersteuerungsbereich (32511) -20...20mA = Nennwert (-27648...27648) -23,51mA = Ende Untersteuerungsbereich (-32512) Zweierkomplement	0,2% vom Endwert	(1)
2Dh	Strom 4...20mA S7-Format von Siemens	1,185 .. +22,81mA / 22,81mA = Ende Übersteuerungsbereich (32511) 4...20mA = Nennbereich (0...27648) 1,18mA = Ende Untersteuerungsbereich (-4864) Zweierkomplement	0,5% vom Endwert	(1)
58h	Spannung $\pm 10V$	-10 ... -9,9V (-10000 ... -9901) <sup>1)</sup> -9,9 ... 9,9V (-9900 ... 9900) -10...10V= Nennbereich (-10000...10000) 9,9 ... 10V (9901 ... 10000) <sup>1)</sup>	0,2% vom Endwert	(1), (2)
59h	Spannung $\pm 4V$	$\pm 4,95V$ / 4,95V = Ende Übersteuerungsbereich (4950) -4...4V = Nennbereich (-4000...4000) -4,95V = Ende Untersteuerungsbereich (-4950) Zweierkomplement	0,2% vom Endwert	(1), (2)
5Ah	Spannung $\pm 400mV$	$\pm 0,495V$ / 495mV = Ende Übersteuerungsbereich (4950) -400...400mV = Nennbereich (-4000...4000) -495mV = Ende Untersteuerungsbereich (-4950) Zweierkomplement	0,4% vom Endwert	(1)
5Ch	Strom $\pm 20mA$	$\pm 25mA$ / 25mA = Ende Übersteuerungsbereich (25000) -20...20mA = Nennwert (-20000...20000) -25mA = Ende Untersteuerungsbereich (-25000) Zweierkomplement	0,2% vom Endwert	(1)
5Dh	Strom 4...20mA	0,8 .. +24,00mA / 24,00mA = Ende Übersteuerungsbereich (20000) 4...20mA = Nennbereich (0...16000) 0,8mA = Ende Untersteuerungsbereich (-3200) Zweierkomplement	0,5% vom Endwert	(1)
FFh	Kanal nicht aktiv (abgeschaltet)			

1) hängt vom Kalibrierungsfaktor ab, der Wert ist nicht garantiert

**Hinweis!**

Das Modul ist ab Werk auf den Messbereich "Spannung ±10V" im S7-Format von Siemens eingestellt.

**Zahlendarstellung im S7-Format von Siemens**

Die Darstellung des Analogwertes erfolgt im Zweierkomplement.

*Zahlendarstellung:*

Byte	Bit 7 ... Bit 0
0	Bit 0 ... 7: Binärer Messwert
1	Bit 0 ... 6: Binärer Messwert Bit 7: Vorzeichen 0 positiv 1 negativ

**+/- 10V**

Spannung	Dezimal	Hex
-10V	-27648	9400
-5V	-13824	CA00
0V	0	0
+5V	13824	3600
+10V	+27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{U}{10}, \quad U = \text{Wert} \cdot \frac{10}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**+/-4V**

Spannung	Dezimal	Hex
-4V	-27648	9400
0V	0	0
4V	27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{U}{4}, \quad U = \text{Wert} \cdot \frac{4}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**+/-400mV**

Spannung	Dezimal	Hex
-400mV	-27648	9400
0V	0	0
400mV	27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{U}{400}, \quad U = \text{Wert} \cdot \frac{400}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**4...20mA**

Strom	Dezimal	Hex
+4mA	0	0
+12mA	+13824	3600
+20mA	+27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{I - 4}{16}, \quad I = \text{Wert} \cdot \frac{16}{27648} + 4$$

I: Stromwert, Wert: Dezimalwert

**+/- 20mA**

Strom	Dezimal	Hex
-20mA	-27648	9400
-10mA	-13824	CA00
0mA	0	0
+10mA	+13824	3600
+20mA	+27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{I}{20}, \quad I = \text{Wert} \cdot \frac{20}{27648}$$

I: Stromwert, Wert: Dezimalwert

**Messdaten erfassen**

Während der Messung werden die Messdaten im Dateneingabebereich abgelegt. Die Zuordnung der Messdaten zu einem Messwert und die jeweiligen Toleranzen finden Sie in der Tabelle oben.

Die nachfolgende Tabelle zeigt den Aufbau des Dateneingabebereichs:

*Dateneingabebereich:*

Byte	Bit 7 ... Bit 0
0	High-Byte Kanal 0
1	Low-Byte Kanal 0
2	High-Byte Kanal 1
3	Low-Byte Kanal 1
4	High-Byte Kanal 2
5	Low-Byte Kanal 2
6	High-Byte Kanal 3
7	Low-Byte Kanal 3

**Parametrierdaten**

Jeder Kanal ist einzeln parametrierbar. Für die Parametrierung stehen 32Byte Parametrierdaten zur Verfügung. Sie werden permanent abgelegt und bleiben auch im ausgeschalteten Zustand erhalten.

Die nachfolgende Tabelle zeigt den Aufbau des Parameterbereichs:

*Parameterbereich:*

Byte	Bit 7 ... Bit 0	Default
0	Diagnosealarm-Byte: Bit 0 ... 5: reserviert Bit 6: 0: Diagnosealarm gesperrt 1: Diagnosealarm freigegeben Bit 7: reserviert	00h
1	Grenzwertüberwachung: Bit 0: Grenzwertüberwachung Kanal 0 Bit 1: Grenzwertüberwachung Kanal 1 Bit 2: Grenzwertüberwachung Kanal 2 Bit 3: Grenzwertüberwachung Kanal 3 Bit 4 ... 7: reserviert	00h
2	Funktions-Nr. Kanal 0 (siehe Tabelle)	28h
3	Funktions-Nr. Kanal 1 (siehe Tabelle)	28h
4	Funktions-Nr. Kanal 2 (siehe Tabelle)	28h
5	Funktions-Nr. Kanal 3 (siehe Tabelle)	28h
6-9	reserviert	00h

*Fortsetzung ...*

... Fortsetzung Parameterbereich

Byte	Bit 7 ... Bit 0	Default
10	Bit 0 ... 2: Mittelwert 000: deaktiviert 001: Mittelwert über 2 Werte 010: Mittelwert über 4 Werte 011: Mittelwert über 8 Werte 100: Mittelwert über 16 Werte 101, 011, 111: deaktiviert Bit 3 ... 7: reserviert	00h
11-15	reserviert	00h
16	Kanal 0, oberer Grenzwert, High-Byte	7Fh
17	Kanal 0, oberer Grenzwert, Low-Byte	FFh
18	Kanal 0, unterer Grenzwert, High-Byte	80h
19	Kanal 0, unterer Grenzwert, Low-Byte	00h
20	Kanal 1, oberer Grenzwert, High-Byte	7Fh
21	Kanal 1, oberer Grenzwert, Low-Byte	FFh
22	Kanal 1, unterer Grenzwert, High-Byte	80h
23	Kanal 1, unterer Grenzwert, Low-Byte	00h
24	Kanal 2, oberer Grenzwert, High-Byte	7Fh
25	Kanal 2, oberer Grenzwert, Low-Byte	FFh
26	Kanal 2, unterer Grenzwert, High-Byte	80h
27	Kanal 2, unterer Grenzwert, Low-Byte	00h
28	Kanal 3, oberer Grenzwert, High-Byte	7Fh
29	Kanal 3, oberer Grenzwert, Low-Byte	FFh
30	Kanal 3, unterer Grenzwert, High-Byte	80h
31	Kanal 3, unterer Grenzwert, Low-Byte	00h

**Diagnosedaten**

Die Diagnosedaten sind 12Byte lang und stehen in den Datensätzen 0 und 1 des Systemdatenbereichs.

Sobald Sie die Diagnosealarmfreigabe in Byte 0 des Parameterbereichs aktiviert haben, wird im Fehlerfall *Datensatz 0* an das übergeordnete System übergeben.

Der *Datensatz 0* hat eine feste Belegung und eine Länge von 4Byte. Die Inhalte von *Datensatz 0* können Sie im Diagnosefenster der CPU im Klartext ausgeben lassen.

Zur erweiterten Diagnose während der Laufzeit haben Sie auch die Möglichkeit, über die SFCs 51 und 59 den 12Byte großen *Datensatz 1* auszuwerten.

**Diagnose auswerten**

Bei anstehender Diagnose unterbricht die CPU ihr Anwenderprogramm und verzweigt in den OB82. In diesem OB können Sie durch entsprechende Programmierung mit den SFCs 51 und 59 detaillierte Diagnoseinformationen abrufen und auf die Diagnose reagieren.

Nach Abarbeitung des OB82 wird die Bearbeitung des Anwenderprogramms wieder fortgesetzt. Die Diagnosedaten sind bis zum Verlassen des OB82 konsistent.

**Datensatz 0**

*Byte 0 bis 3:*

*Datensatz 0 (Byte 0 bis 3):*

Byte	Bit 7 ... Bit 0	Default
0	Bit 0: Störung im Modul Bit 1: reserviert Bit 2: Fehler extern Bit 3: Kanalfehler vorhanden Bit 4 ... 6: reserviert Bit 7: Falsche Parameter im Modul	00h
1	Bit 0 ... 3: Modulklasse 0101 Analogmodul Bit 4: Kanalinformation vorhanden Bit 5 ... 7: reserviert	15h
2	nicht belegt	00h
3	Bit 0 ... 5: reserviert Bit 6: Prozessalarm verloren (siehe Prozessalarm) Bit 7: reserviert	00h

**Datensatz 1***Byte 0 bis 11:*

Der Datensatz 1 enthält die 4Byte des Datensatzes 0 und zusätzlich 8Byte modulspezifische Diagnosedaten.

Die Diagnosebytes haben folgende Belegung:

*Datensatz 1 (Byte 0 bis 11):*

Byte	Bit 7 ... Bit 0	Default
0 ... 3	Inhalte Datensatz 0 (siehe vorherige Seite)	-
4	Bit 0 ... 6: Kanaltyp 70h: Digitaleingabe 71h: Analogeingabe 72h: Digitalausgabe 73h: Analogausgabe Bit 7: reserviert	71h
5	Bit 0 ... 7: Anzahl der Diagnosebits, die das Modul pro Kanal ausgibt	04h
6	Bit 0 ... 7: Anzahl der gleichartigen Kanäle eines Moduls	04h
7	Bit 0: Kanalfehler Kanal 0 Bit 1: Kanalfehler Kanal 1 Bit 2: Kanalfehler Kanal 2 Bit 3: Kanalfehler Kanal 3 Bit 4 ... 7: reserviert	00h
8	Bit 0: reserviert Bit 1: Parametrierfehler Kanal 0 Bit 2 ... 4: reserviert Bit 5: Parametrierfehler Kanal 1 Bit 6, 7: reserviert	00h
9	Bit 0: reserviert Bit 1: Parametrierfehler Kanal 2 Bit 2 ... 4: reserviert Bit 5: Parametrierfehler Kanal 3 Bit 6, 7: reserviert	00h
10 ... 11	reserviert	00h

**Prozessalarm**

Der obere und untere Grenzwert ist für jeden Kanal parametrierbar. Beim Parametrieren ist zu berücksichtigen, dass die Grenzwertüberwachung im Parameterbyte 1 freigegeben sein muss.

Verlässt das Signal den definierten Arbeitsbereich, so wird ein Prozessalarm ausgelöst. In der CPU wird der Prozessalarmbaustein (OB40) aufgerufen.

Die 4Byte der Prozessalarmzusatzinformation sind wie folgt belegt:

*Prozessalarmzusatzinformation*

Byte	Bit 7 ... Bit 0	Default
0	Bit 0: oberer Grenzwert überschritten Kanal 0 Bit 1: oberer Grenzwert überschritten Kanal 1 Bit 2: oberer Grenzwert überschritten Kanal 2 Bit 3: oberer Grenzwert überschritten Kanal 3 Bit 4 ... 7: reserviert	00h
1	Bit 0: unterer Grenzwert unterschritten Kanal 0 Bit 1: unterer Grenzwert unterschritten Kanal 1 Bit 2: unterer Grenzwert unterschritten Kanal 2 Bit 3: unterer Grenzwert unterschritten Kanal 3 Bit 4 ... 7: reserviert	00h
2	reserviert	00h
3	reserviert	00h

**Hinweis!**

Wenn der Prozessalarm von der CPU noch nicht quittiert worden ist und ein neuer Prozessalarm vom gleichen Typ in diesem Kanal auftritt, so wird ein Diagnosealarm mit der Information "Prozessalarm verloren" (Diagnosedaten Byte 3) ausgelöst.



**Technische Daten**

Elektrische Daten	VIPA 231-1FD00
Anzahl der Eingänge	4 differentielle Eingänge
Zykluszeit (alle Kanäle)	< 1ms
Eingangsbereiche	
- Spannungsmessung	$\pm 400\text{mV}$ , $\pm 4\text{V}$ , $\pm 10\text{V}$
- Strommessung	4...20mA, $\pm 20\text{mA}$
Eingangswiderstand	> $2\text{M}\Omega$ (Spannungsbereich) < $57\Omega$ (Strombereich)
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme	300mA über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	8Byte (1 Wort pro Kanal)
Ausgabedaten	-
Parameterdaten	32Byte
Diagnosedaten	12Byte
Prozessalarmdaten	4Byte
Maße und Gewicht	
Abmessungen (BxHxT)	25,4 x 76 x 76 mm
Gewicht	100g

## AI 8x16Bit

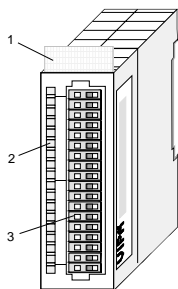
**Bestelldaten** AI 8x16Bit VIPA 231-1BF00

**Beschreibung** Das analoge Eingabe-Modul wandelt analoge Signale aus dem Prozess in digitale Signale für die interne Verarbeitung um. Als Geber können Thermoelemente Typ J, K, T und Widerstandsthermometer PT100, angeschlossen werden. Das Modul besitzt 8 Eingänge, deren Funktion paarweise parametrierbar ist.

**Eigenschaften**

- 8 Analogeingänge
- Überprüfung auf Drahtbruch
- Messwertauflösung 15Bit + Vorzeichen

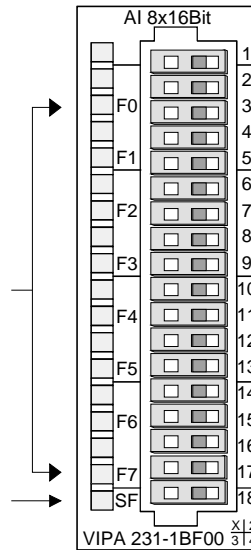
### Aufbau



- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] LED-Statusanzeige
- [3] Steckerleiste

## Statusanzeige Steckerbelegung

LED	Beschreibung	Pin	Belegung
F0 ... F7	LED (rot): Fehlermeldung je Kanal	1	nicht benutzt
		2	+ Kanal 0
		3	Masse Kanal 0
		4	+ Kanal 1
		5	Masse Kanal 1
		6	+ Kanal 2
		7	Masse Kanal 2
		8	+ Kanal 3
		9	Masse Kanal 3
		10	+ Kanal 4
		11	Masse Kanal 4
		12	+ Kanal 5
		13	Masse Kanal 5
		14	+ Kanal 6
		15	Masse Kanal 6
		16	+ Kanal 7
		17	Masse Kanal 7
		18	nicht benutzt
SF	LED (rot) Sammelfehler		

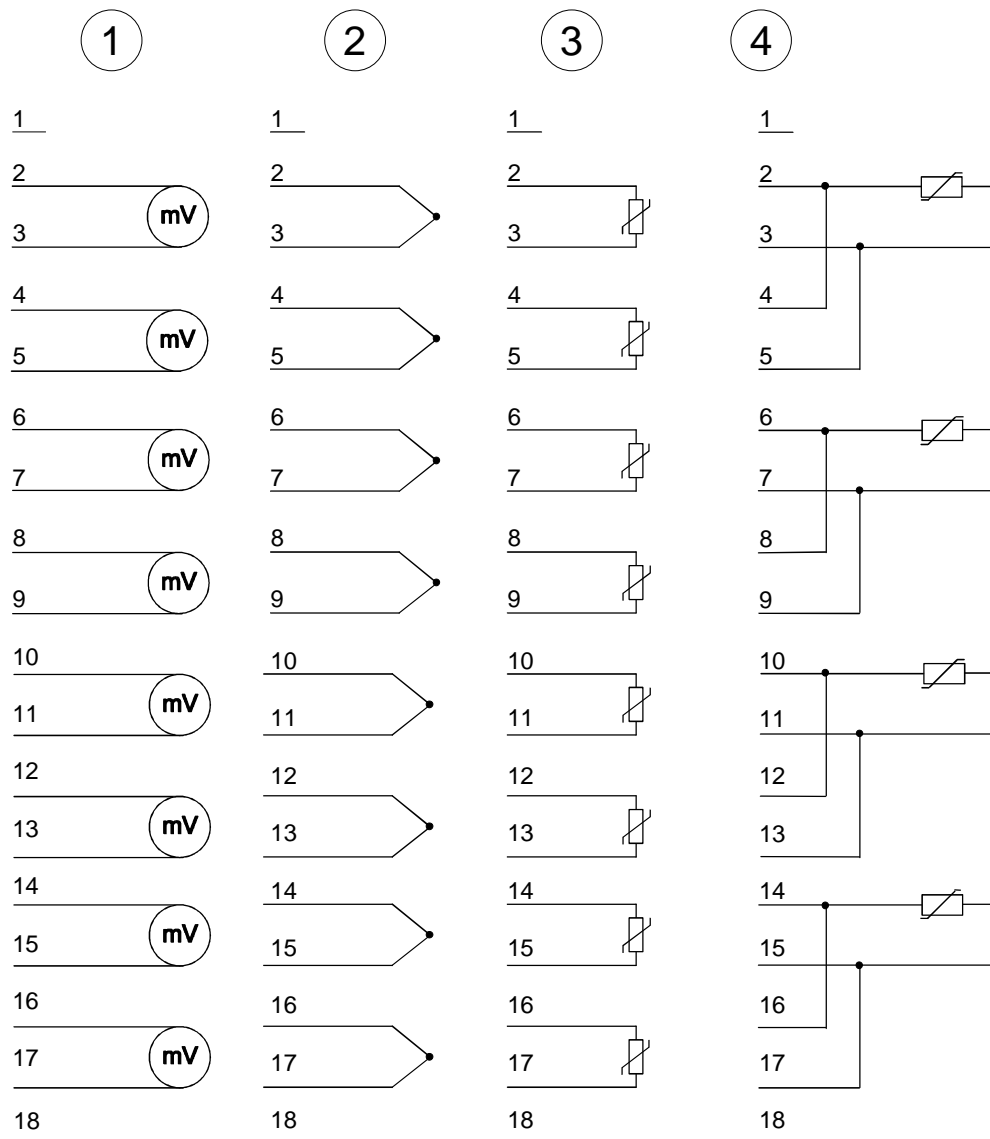


### Hinweis!

Vorübergehend nicht benutzte Eingänge sind bei aktiviertem Kanal mit der zugehörigen Masse zu verbinden.

Sind nicht benutzte Kanäle abgeschaltet, ist diese Maßnahme nicht erforderlich.

Anschlussbilder



**Funktions-Nr.  
Zuordnung**

Die Zuweisung einer Funktions-Nr. zu einem Kanal erfolgt über die Parametrierung. Mit der Funktions-Nr. 00h wird die in den permanent abgelegten Parametrierdaten enthaltene Funktions-Nr. nicht beeinflusst.

Durch Angabe von FFh können Sie den entsprechenden Kanal deaktivieren.

Nr.	Funktion	Messbereich / Darstellung	Messtoleranz bezogen auf Nennbereich	Anschl.
00h	keinen Einfluss auf die permanent abgelegten Parametrierdaten			
01h	RTD PT100 im Zweileiteranschluss	-200 .. +850°C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)3)</sup> ±0,1%	(3)
61h	RTD PT100 im Zweileiteranschluss	-328 .. 1562°F / in Einheit 1/10°F, Zweierkomplement	<sup>1)2)3)</sup> ±0,1%	(3)
09h	RTD PT100 im Vierleiteranschluss	-200 .. +850°C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)</sup> ±0,1%	(4)
69h	RTD PT100 im Vierleiteranschluss	-328 .. 1562°F / in Einheit 1/10°F, Zweierkomplement	<sup>1)2)</sup> ±0,1%	(4)
10h	Thermoelement Typ J , Kompensation extern	0 °C .. 1000 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)4)</sup> ±0,1%	(2)
40h	Thermoelement Typ J , Kompensation extern	32 .. 1832°F / in Einheit 1/10°F, Zweierkomplement	<sup>1)2)4)</sup> ±0,1%	(2)
11h	Thermoelement Typ K , Kompensation extern	0 °C .. 1300 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)4)</sup> ±0,1%	(2)
41h	Thermoelement Typ K , Kompensation extern	32 .. 2372°F / in Einheit 1/10°F, Zweierkomplement	<sup>1)2)4)</sup> ±0,1%	(2)
14h	Thermoelement Typ T , Kompensation extern	-200 °C .. +400 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)4)</sup> -200...-60,1 ±0,5% -60...400 ±0,2%	(2)
44h	Thermoelement Typ T , Kompensation extern	-328 .. 752°F / in Einheit 1/10°F, Zweierkomplement	<sup>1)2)4)</sup> -328...-76,1 ±0,5% -76...752 ±0,2%	(2)
18h	Thermoelement Typ J , Kompensation intern	0 °C .. 1000 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)5)</sup> ±1,0%	(2)
48h	Thermoelement Typ J , Kompensation intern	32 .. 1832°F / in Einheit 1/10°F, Zweierkomplement	<sup>1)2)5)</sup> ±1,0%	(2)
19h	Thermoelement Typ K , Kompensation intern	0 °C .. 1300 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)5)</sup> ±1,0%	(2)
49h	Thermoelement Typ K , Kompensation intern	32 .. 2372°F / in Einheit 1/10°F, Zweierkomplement	<sup>1)2)5)</sup> ±1,0%	(2)
1Ch	Thermoelement Typ T , Kompensation intern	-200 °C .. +400 °C / in Einheit 1/10°C, Zweierkomplement	<sup>1)2)5)</sup> ±2,0%	(2)
4Ch	Thermoelement Typ T , Kompensation intern	-328 .. 752°F / in Einheit 1/10°F, Zweierkomplement	<sup>1)2)5)</sup> ±2,0%	(2)
26h	Spannung 0...60mV	0...60mV = Nennbereich (0-27648)	<sup>1)</sup> ±0,1%	(1)
56h	Spannung 0...60mV	0...60mV = Nennbereich (0-6000) in Einheit 1/100mV	<sup>1)</sup> ±0,1%	(1)
FFh	Kanal nicht aktiv (abgeschaltet)			

<sup>1)</sup> ermittelt bei Umgebungstemperatur 25°C, Geschwindigkeit 15 Wandlungen/s

<sup>2)</sup> ausgenommen sind Fehler durch Ungenauigkeit des Gebers

<sup>3)</sup> ausgenommen sind Fehler durch Übergangswiderstände an Kontakten sowie Leitungswiderstände

<sup>4)</sup> die Kompensation der Kaltstelle muss extern durchgeführt werden

<sup>5)</sup> die Kompensation der Kaltstelle wird intern durchgeführt indem die Temperatur des Frontsteckers berücksichtigt wird. Die Thermoelementleiter sind unmittelbar am Frontstecker anzuschließen, ggf. muss mit Thermoelement-Verlängerungskabel verlängert werden.

**Messdaten erfassen**

Während der Messung werden die Messdaten im Dateneingabebereich abgelegt. Die Zuordnung der Messdaten zu einem Messwert und die jeweiligen Toleranzen finden Sie in der Tabelle oben.

Die nachfolgende Tabelle zeigt den Aufbau des Dateneingabebereichs:

*Dateneingabebereich:*

Byte	Bit 7 ... Bit 0
0	High-Byte Kanal 0
1	Low-Byte Kanal 0
2	High-Byte Kanal 1
3	Low-Byte Kanal 1
4	High-Byte Kanal 2
5	Low-Byte Kanal 2
6	High-Byte Kanal 3
7	Low-Byte Kanal 3
8	High-Byte Kanal 4
9	Low-Byte Kanal 4
10	High-Byte Kanal 5
11	Low-Byte Kanal 5
12	High-Byte Kanal 6
13	Low-Byte Kanal 6
14	High-Byte Kanal 7
15	Low-Byte Kanal 7

**Hinweis!**

Bei Vierleitermessung werden nur die Kanäle 0, 2, 4 und 6 verwendet.

**Parametrierdaten**

Die Kanäle sind paarweise parametrierbar. Für die Parametrierung stehen 10Byte Parametrierdaten zur Verfügung. Sie werden permanent abgelegt und bleiben auch im ausgeschalteten Zustand erhalten.

Die nachfolgende Tabelle zeigt den Aufbau des Parameterbereichs:

*Parameterbereich:*

Byte	Bit 7 ... Bit 0	Default
0	Diagnosealarm-Byte: Bit 0: 0: Drahtbruchererkennung Kanal 0/1 aus 1: Drahtbruchererkennung Kanal 0/1 ein Bit 1: 0: Drahtbruchererkennung Kanal 2/3 aus 1: Drahtbruchererkennung Kanal 2/3 ein Bit 2: 0: Drahtbruchererkennung Kanal 4/5 aus 1: Drahtbruchererkennung Kanal 4/5 ein Bit 3: 0: Drahtbruchererkennung Kanal 6/7 aus 1: Drahtbruchererkennung Kanal 6/7 ein Bit 4, 5: reserviert Bit 6: 0: Diagnosealarm gesperrt 1: Diagnosealarm freigegeben Bit 7: reserviert	0Fh
1	reserviert	00h
2	Funktions-Nr. Kanal 0/1 (siehe Tabelle)	26h
3	Funktions-Nr. Kanal 2/3 (siehe Tabelle)	26h
4	Funktions-Nr. Kanal 4/5 (siehe Tabelle)	26h
5	Funktions-Nr. Kanal 6/7 (siehe Tabelle)	26h
6	Option-Byte Kanal 0/1	00h
7	Option-Byte Kanal 2/3	00h
8	Option-Byte Kanal 4/5	00h
9	Option-Byte Kanal 6/7	00h

**Parameter**

*Diagnosealarm*

Mit Bit 6 von Byte 0 können Sie den Diagnosealarm freigeben. Im Fehlerfall werden an Ihr übergeordnetes Master-System 4 Diagnose-Bytes geschickt.

*Funktions-Nr.*

Tragen Sie hier für je 2 Kanäle die Funktions-Nummer Ihrer Messfunktion ein. Die Zuordnung der Funktions-Nummer zu einer Messfunktion entnehmen Sie bitte der obigen Tabelle.

*Option-Byte*

Hier können Sie für je 2 Kanäle die Wandlergeschwindigkeit einstellen.



**Hinweis!**

Bitte beachten Sie, dass bei höheren Wandlergeschwindigkeiten die Auflösung aufgrund der kürzeren Integrationszeit sinkt.

Das Datenübergabeformat bleibt gleich. Lediglich die unteren Bits (LSBs) sind für den Analogwert nicht mehr aussagekräftig.

*Aufbau Option-Byte:*

Byte	Bit 7 ... Bit 0	Auflösung	Default
6 ... 9	Option-Byte: Bit 0 ... 3: Geschwindigkeit*		00h
	0000 15,0 Wandlungen/s	16	
	0001 30,1 Wandlungen/s	16	
	0010 60,0 Wandlungen/s	15	
	0011 123,2 Wandlungen/s	14	
	0100 168,9 Wandlungen/s	12	
	0101 202,3 Wandlungen/s	10	
	0110 3,76 Wandlungen/s	16	
	0111 7,51 Wandlungen/s	16	
	Bit 4 ... 7: reserviert		

\*) Die Angaben beziehen sich auf 1-Kanal-Betrieb. Wenn Sie die angegebenen Wandlergeschwindigkeiten durch die Anzahl der aktiven Kanäle dividieren, erhalten Sie bei Mehrkanalbetrieb die Wandlergeschwindigkeit pro Kanal.



**Diagnosedaten**

Die Diagnosedaten sind 12Byte lang und stehen in den Datensätzen 0 und 1 des Systemdatenbereichs.

Sobald Sie die Diagnosealarmfreigabe in Byte 0 des Parameterbereichs aktiviert haben, wird im Fehlerfall *Datensatz 0* an das übergeordnete System übergeben.

Der *Datensatz 0* hat eine feste Belegung und eine Länge von 4Byte. Die Inhalte von *Datensatz 0* können Sie im Diagnosefenster der CPU im Klartext ausgeben lassen.

Zur erweiterten Diagnose während der Laufzeit haben Sie auch die Möglichkeit, über die SFCs 51 und 59 den 12Byte großen *Datensatz 1* auszuwerten.

**Diagnose auswerten**

Bei anstehender Diagnose unterbricht die CPU ihr Anwenderprogramm und verzweigt in den OB82. In diesem OB können Sie durch entsprechende Programmierung mit den SFCs 51 und 59 detaillierte Diagnoseinformationen abrufen und auf die Diagnose reagieren.

Nach Abarbeitung des OB82 wird die Bearbeitung des Anwenderprogramms wieder fortgesetzt. Die Diagnosedaten sind bis zum Verlassen des OB82 konsistent.

**Datensatz 0**

*Byte 0 bis 3:*

*Datensatz 0 (Byte 0 bis 3):*

Byte	Bit 7 ... Bit 0	Default
0	Bit 0: Störung im Modul Bit 1: reserviert Bit 2: Fehler extern Bit 3: Kanalfehler vorhanden Bit 4 ... 6: reserviert Bit 7: Falsche Parameter im Modul	00h
1	Bit 0 ... 3: Modulklasse 0101 Analogmodul Bit 4: Kanalinformation vorhanden Bit 5 ... 7: reserviert	15h
2	nicht belegt	00h
3	Bit 0 ... 5: reserviert Bit 6: Prozessalarm verloren (siehe Prozessalarm) Bit 7: reserviert	00h

**Datensatz 1**

*Byte 0 bis 11:*

Der Datensatz 1 enthält die 4Byte des Datensatzes 0 und zusätzlich 8Byte modulspezifische Diagnosedaten.

Die Diagnosebytes haben folgende Belegung:

*Datensatz 1 (Byte 0 bis 11):*

Byte	Bit 7 ... Bit 0	Default
0 ... 3	Inhalte Datensatz 0 (siehe vorherige Seite)	-
4	Bit 0 ... 6: Kanaltyp 70h: Digitaleingabe 71h: Analogeingabe 72h: Digitalausgabe 73h: Analogausgabe Bit 7: reserviert	71h
5	Bit 0 ... 7: Anzahl der Diagnosebits, die das Modul pro Kanal ausgibt	04h
6	Bit 0 ... 7: Anzahl der gleichartigen Kanäle eines Moduls	04h
7	Bit 0: Kanalfehler Kanal 0 Bit 1: Kanalfehler Kanal 1 Bit 2: Kanalfehler Kanal 2 Bit 3: Kanalfehler Kanal 3 Bit 4: Kanalfehler Kanal 4 Bit 5: Kanalfehler Kanal 5 Bit 6: Kanalfehler Kanal 6 Bit 7: Kanalfehler Kanal 7	00h
8	Bit 0: Drahtbruch Kanal 0 Bit 1: Parametrierungsfehler Kanal 0 Bit 2: Meßbereichsunterschreitung Kanal 0 Bit 3: Meßbereichsüberschreitung Kanal 0 Bit 4: Drahtbruch Kanal 1 Bit 5: Parametrierungsfehler Kanal 1 Bit 6: Meßbereichsunterschreitung Kanal 1 Bit 7: Meßbereichsüberschreitung Kanal 1	00h
9	Bit 0: Drahtbruch Kanal 2 Bit 1: Parametrierungsfehler Kanal 2 Bit 2: Meßbereichsunterschreitung Kanal 2 Bit 3: Meßbereichsüberschreitung Kanal 2 Bit 4: Drahtbruch Kanal 3 Bit 5: Parametrierungsfehler Kanal 3 Bit 6: Meßbereichsunterschreitung Kanal 3 Bit 7: Meßbereichsüberschreitung Kanal 3	00h

*Fortsetzung ...*

... Fortsetzung

Byte	Bit 7 ... Bit 0	Default
10	Bit 0: Drahtbruch Kanal 4 Bit 1: Parametrierungsfehler Kanal 4 Bit 2: Meßbereichsunterschreitung Kanal 4 Bit 3: Meßbereichsüberschreitung Kanal 4 Bit 4: Drahtbruch Kanal 5 Bit 5: Parametrierungsfehler Kanal 5 Bit 6: Meßbereichsunterschreitung Kanal 5 Bit 7: Meßbereichsüberschreitung Kanal 5	00h
11	Bit 0: Drahtbruch Kanal 6 Bit 1: Parametrierungsfehler Kanal 6 Bit 2: Meßbereichsunterschreitung Kanal 6 Bit 3: Meßbereichsüberschreitung Kanal 6 Bit 4: Drahtbruch Kanal 7 Bit 5: Parametrierungsfehler Kanal 7 Bit 6: Meßbereichsunterschreitung Kanal 7 Bit 7: Meßbereichsüberschreitung Kanal 7	00h

**Technische Daten**

Elektrische Daten	VIPA 231-1BF00
Anzahl der Eingänge	8
Eingangswiderstand	> 2MΩ
Eingangsbereiche	
- Thermoelemente	Typ J, K, T
- Widerstandsthermometer	Pt100
- Spannungsmessung	0...60mV
Spannungsversorgung	5V über Rückwandbus
Stromaufnahme	280mA über Rückwandbus
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Verlustleistung	typ. 1,3W
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	16Byte (1 Wort pro Kanal)
Ausgabedaten	-
Parameterdaten	10Byte
Diagnosedaten	12Byte
Prozessalarmdaten	-
Maße und Gewicht	
Abmessungen (BxHxT)	25,4x76x76mm
Gewicht	120g



## Teil 17 Analoge Ausgabe-Module

### Überblick

Inhalt dieses Kapitels ist der Aufbau und die Funktionsweise der analogen Ausgabe-Module von VIPA.

Nachfolgend sind beschrieben:

- Systemübersicht der analogen Ausgabe-Module
- Eigenschaften
- Aufbau
- Anschluss- und Prinzipschaltbild
- Technische Daten

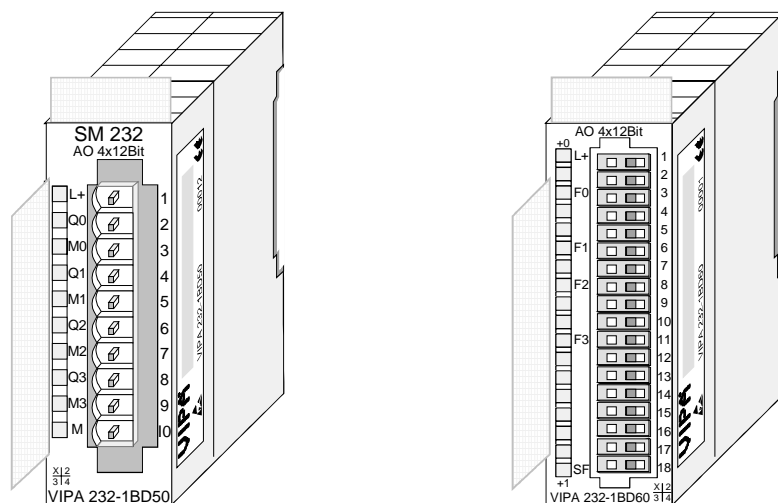
### Inhalt

Thema	Seite
<b>Teil 17 Analoge Ausgabe-Module</b> .....	<b>17-1</b>
Systemübersicht .....	17-2
Allgemeines .....	17-3
AO 4x12Bit, Multi-Output .....	17-4
AO 4x12Bit f, Multi-Output .....	17-12

## Systemübersicht

### Ausgabe-Module SM 232

Nachfolgend finden Sie eine Übersicht der bei VIPA zur Zeit erhältlichen Analogen Ausgabe-Module:



### Bestelldaten Ausgabe-Module

Typ	Bestellnummer	Seite
AO 4x12Bit, Multioutput	VIPA 232-1BD50	17-4
AO 4x12Bit f, Multioutput	VIPA 232-1BD60	17-12

## Allgemeines

### Leitungen für Analogsignale

Für die Analogsignale sollten Sie geschirmte und paarweise verdrehte Leitungen verwenden. Hierdurch verringern Sie die Störbeeinflussung. Den Schirm der Analogleitungen sollten Sie an beiden Leitungsenden erden. Wenn Potenzialunterschiede zwischen den Leitungsenden bestehen, so kann ein Potenzialausgleichstrom fließen, der die Analogsignale stören könnte. In diesem Fall sollten Sie den Schirm nur an einem Leitungsende erden.

### Anschließen von Lasten und Aktoren

Da auch die Aktoren extern zu versorgen sind, können Sie Aktoren in 2- und in 4-Draht-Technik anschließen. Bei der Ausgabe von Stellwerten an 2-Draht-Aktoren ist eine Spannungsquelle in die Steuerleitung einzuschleifen. 4-Draht-Aktoren sind extern zu versorgen.



#### **Hinweis!**

Bitte achten Sie beim Anschluss der Aktoren immer auf richtige Polarität! Lassen Sie die Ausgangsklemmen der nicht benutzten Kanäle unbeschaltet!

### Parametrierung und Diagnose zur Laufzeit

Durch Einsatz der SFCs 55, 56 und 57 können Sie zur Laufzeit über die CPU 21x die Parameter in Ihren Analog-Modulen ändern.

Zur Diagnoseauswertung zur Laufzeit stehen Ihnen die beiden SFCs 51 und 59 zur Verfügung. Hiermit können Sie detaillierte Diagnoseinformationen abrufen und auf die Diagnose reagieren.

# AO 4x12Bit, Multi-Output

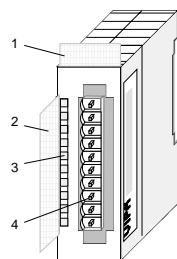
**Bestelldaten** AO 4x12Bit Multi-Output VIPA 232-1BD50

**Beschreibung** Das Modul besitzt 4 Ausgänge, deren Funktion Sie einzeln parametrieren können. Im Prozessabbild belegt das Modul insgesamt 8Byte Ausgangsdaten (2Byte pro Kanal). Die Werte sind im Zweierkomplement linksbündig vorzugeben.

Die Kanäle auf dem Modul sind gegenüber dem Rückwandbus und der ext. Versorgungsspannung mittels DC/DC-Wandlern und Optokopplern galvanisch getrennt. Das Modul ist extern mit DC 24V zu versorgen.

- Eigenschaften**
- 4 Ausgänge, deren Massen verbunden sind
  - für jeden Ausgang einzelparametrierbare Funktionalität
  - Geeignet für Aktoren mit den Eingängen ±10V, 1 ... 5V, 0 ... 10V, ±20mA, 4 ... 20mA oder 0 ... 20mA
  - Diagnose-LED und Diagnosefunktion

**Aufbau**



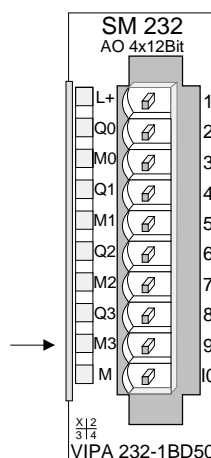
- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bitadresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige Steckerbelegung**

**LED Beschreibung**

- M3 Sammeldiagnose LED leuchtet rot wenn:
- Kurzschluss bei Spannungsausgabe
  - Drahtbruch bei Stromausgabe
  - CPU sich in STOP befindet
  - Buskoppler nicht mit Spannung versorgt wird

**Pin Belegung**



- 1 Versorgungsspg. DC 24V
- 2 + Kanal 0
- 3 Masse Kanal 0
- 4 + Kanal 1
- 5 Masse Kanal 1
- 6 + Kanal 2
- 7 Masse Kanal 2
- 8 + Kanal 3
- 9 Masse Kanal 3
- 10 Versorgungsspg. Masse



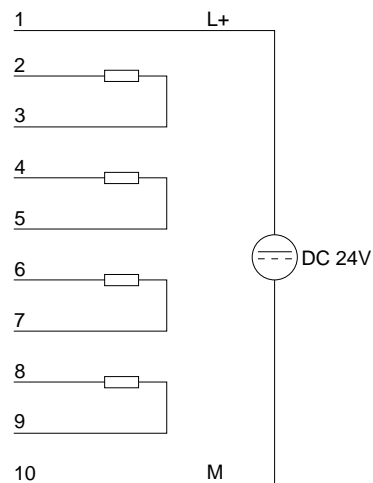
**Hinweis!**

Bitte beachten Sie, dass aus entwicklungsstechnischen Gründen die Diagnose-LED für das **komplette** Modul mit M3 beschriftet ist!

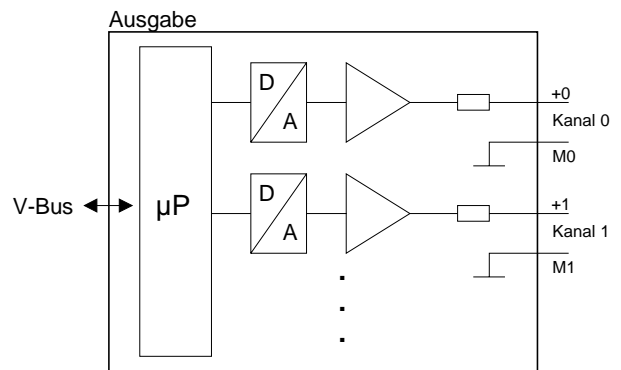


Anschlussbild und  
Prinzipschaltbild

## Anschlussbild



## Prinzipschaltbild

Funktions-Nr.  
Zuordnung

Die Zuweisung einer Funktions-Nr. zu einem Kanal erfolgt über die Parametrierung. Mit der Funktions-Nr. 00h erfolgt keine Ausgabe.

Nr.	Funktion	Ausgabebereich	Toleranz
00h	keine Ausgabe		
01h	Spannung $\pm 10V$ S5-Format von Siemens	$\pm 12,5V$ 12,5V = Ende Übersteuerungsbereich (20480) -10...10V = Nennbereich (-16384...16384) -12,5V = Ende Untersteuerungsbereich (-20480)	<sup>1)</sup> $\pm 0,2\%$ vom Endwert
02h	Spannung 1...5V S5-Format von Siemens	0...6V 6V = Ende Übersteuerungsbereich (20480) 1...5V = Nennbereich (0...16384) 0V = Ende Untersteuerungsbereich (-4096)	<sup>1)</sup> $\pm 0,2\%$ vom Endwert
05h	Spannung 0...10V S5-Format von Siemens	0...12,5V 12,5V = Ende Übersteuerungsbereich (20480) 0...10V = Nennbereich (0...16384) kein Untersteuerungsbereich	<sup>1)</sup> $\pm 0,2\%$ vom Endwert
09h	Spannung $\pm 10V$ S7-Format von Siemens (Zweierkomplement)	$\pm 11,85V$ 11,85V = Ende Übersteuerungsbereich (32767) -10V...10V = Nennbereich (-27648...27648) -11,85 = Ende Untersteuerungsbereich (-32767)	<sup>1)</sup> $\pm 0,2\%$ vom Endwert
0Ah	Spannung 1...5V S7-Format von Siemens (Zweierkomplement)	0...5,75V 5,75V = Ende Übersteuerungsbereich (32767) 1...5V = Nennbereich (0...27648) 0V = Ende Untersteuerungsbereich (-6912)	<sup>1)</sup> $\pm 0,2\%$ vom Endwert
0Dh	Spannung 0...10V S7-Format von Siemens (Zweierkomplement)	0...11,85V 11,85V = Ende Übersteuerungsbereich (32767) 0...10V = Nennbereich (0...27648) kein Untersteuerungsbereich	<sup>1)</sup> $\pm 0,2\%$ vom Endwert

Fortsetzung ...

... Fortsetzung

Nr.	Funktion	Ausgabebereich	Toleranz
03h	Strom $\pm 20\text{mA}$ S5-Format von Siemens	$\pm 25\text{mA}$ 25mA = Ende Übersteuerungsbereich (20480) -20...20mA = Nennbereich (-16384...16384) -25mA = Ende Untersteuerungsbereich (-20480)	<sup>1)</sup> $\pm 0,2\%$ vom Endwert
04h	Strom 4...20mA S5-Format von Siemens	0...25mA 25mA = Ende Übersteuerungsbereich (20480) 4...20mA = Nennbereich (0...16384) 0mA = Ende Untersteuerungsbereich (-4096)	<sup>1)</sup> $\pm 0,2\%$ vom Endwert
06h	Strom 0...20mA S5-Format von Siemens	0...25mA 25mA = Ende Übersteuerungsbereich (20480) 0...20mA = Nennbereich (0...16384) kein Untersteuerungsbereich	<sup>1)</sup> $\pm 0,2\%$ vom Endwert
0Bh	Strom $\pm 20\text{mA}$ S7-Format von Siemens (Zweierkomplement)	$\pm 23,70\text{mA}$ 23,70mA = Ende Übersteuerungsbereich (32767) -20...20mA = Nennbereich (-27648...27648) -23,70mA = Ende Untersteuerungsbereich (-32767)	<sup>1)</sup> $\pm 0,2\%$ vom Endwert
0Ch	Strom 4...20mA S7-Format von Siemens (Zweierkomplement)	0...22,96mA 22,96mA = Ende Übersteuerungsbereich (32767) 4...20mA = Nennbereich (0...27648) 0mA = Ende Untersteuerungsbereich (-5530)	<sup>1)</sup> $\pm 0,2\%$ vom Endwert
0Eh	Strom 0...20mA S7-Format von Siemens (Zweierkomplement)	0...23,70mA 23,70mA = Ende Übersteuerungsbereich (32767) 0...20mA = Nennbereich (0...27648) kein Untersteuerungsbereich	<sup>1)</sup> $\pm 0,2\%$ vom Endwert

<sup>1)</sup> ermittelt bei Umgebungstemp. 25°C, Geschwindigkeit 15 Wandlungen/s

**Zahlendarstellung im S5-Format von Siemens**

Die Eingabedaten werden im S5-Format von Siemens in einem Wort abgelegt. Das Wort setzt sich zusammen aus dem binären Wert und den Informationsbits.

*Zahlendarstellung:*

Byte	Bit 7 ... Bit 0
0	Bit 0: Überlaufbit 0: Wert liegt im Messbereich 1: Messbereich überschritten Bit 1: Fehlerbit (gesetzt bei internem Fehler) Bit 2: Tätigkeitsbit (immer 0) Bit 3 ... 7: Binärer Messwert
1	Bit 0 ... 6: Binärer Messwert Bit 7: Vorzeichen 0: positiv 1: negativ

**+/- 10V**

Spannung	Dezimal	Hex
-10V	-16384	C000
-5V	-8192	E000
0V	0	0
+5V	8192	2000
+10V	+16384	4000

Formeln für die Berechnung:

$$\text{Wert} = 16384 \cdot \frac{U}{10}, \quad U = \text{Wert} \cdot \frac{10}{16384}$$

U: Spannungswert, Wert: Dezimalwert

**0...10V**

Spannung	Dezimal	Hex
0V	0	0000
5V	8192	2000
10V	16384	4000

Formeln für die Berechnung:

$$\text{Wert} = 16384 \cdot \frac{U}{10}, \quad U = \text{Wert} \cdot \frac{10}{16384}$$

U: Spannungswert, Wert: Dezimalwert

**1...5V**

Spannung	Dezimal	Hex
+1V	0	0
+3V	+8192	2000
+5V	+16384	4000

Formeln für die Berechnung:

$$\text{Wert} = 16384 \cdot \frac{U - 1}{4}, \quad U = \text{Wert} \cdot \frac{4}{16384} + 1$$

U: Spannungswert, Wert: Dezimalwert

**4....20mA**

Strom	Dezimal	Hex
+4mA	0	0
+12mA	+8192	2000
+20mA	+16384	4000

Formeln für die Berechnung:

$$\text{Wert} = 16384 \cdot \frac{I - 4}{16}, \quad I = \text{Wert} \cdot \frac{16}{16384} + 4$$

I: Stromwert, Wert: Dezimalwert

**+/- 20mA**

Strom	Dezimal	Hex
-20mA	-16384	C000
-10mA	-8192	E000
0mA	0	0
+10mA	+8192	2000
+20mA	+16384	4000

Formeln für die Berechnung:

$$\text{Wert} = 16384 \cdot \frac{I}{20}, \quad I = \text{Wert} \cdot \frac{20}{16384}$$

I: Stromwert, Wert: Dezimalwert

**Zahlendarstellung im S7-Format von Siemens**

Die Darstellung des Analogwertes erfolgt im Zweierkomplement.

Zahlendarstellung:

Byte	Bit 7 ... Bit 0
0	Bit 0 ... 7: Binärer Messwert
1	Bit 0 ... 6: Binärer Messwert Bit 7: Vorzeichen 0: positiv 1: negativ

**+/- 10V**

Spannung	Dezimal	Hex
-10V	-27648	9400
-5V	-13824	CA00
0V	0	0
+5V	13824	3600
+10V	+27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U}{10}, \quad U = Wert \cdot \frac{10}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**0...10V**

Spannung	Dezimal	Hex
0V	0	0000
5V	+13824	3600
10V	+27648	6C00

Formeln für die Berechnung:

$$Wert = 16384 \cdot \frac{U}{10}, \quad U = Wert \cdot \frac{10}{16384}$$

U: Spannungswert, Wert: Dezimalwert

**1...5V**

Spannung	Dezimal	Hex
+1V	0	0
+3V	+13824	3600
+5V	+27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U - 1}{4}, \quad U = Wert \cdot \frac{4}{27648} + 1$$

U: Spannungswert, Wert: Dezimalwert

**+/-4V**

Spannung	Dezimal	Hex
-4V	-27648	9400
0V	0	0
4V	27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U}{4}, \quad U = Wert \cdot \frac{4}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**+/-400mV**

Spannung	Dezimal	Hex
-400mV	-27648	9400
0V	0	0
400mV	27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U}{400}, \quad U = Wert \cdot \frac{400}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**4...20mA**

Strom	Dezimal	Hex
+4mA	0	0
+12mA	+13824	3600
+20mA	+27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{I - 4}{16}, \quad I = Wert \cdot \frac{16}{27648} + 4$$

I: Stromwert, Wert: Dezimalwert

**+/- 20mA**

Strom	Dezimal	Hex
-20mA	-27648	9400
-10mA	-13824	CA00
0mA	0	0
+10mA	+13824	3600
+20mA	+27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{I}{20}, \quad I = Wert \cdot \frac{20}{27648}$$

I: Stromwert, Wert: Dezimalwert

**Datenausgabe**

Die auszugebenden Werte tragen Sie im Datenausgabebereich ein. Die Zuordnung Ihres Ausgabewertes zu einem entsprechenden Strom- bzw. Spannungswert können Sie für jeden Kanal über Funktions-Nr. parametrieren.

Den Aufbau des Datenausgabebereichs sehen Sie in der nachfolgenden Tabelle:

*Datenausgabebereich:*

Byte	Bit 7 ... Bit 0
0	High-Byte Kanal 0
1	Low-Byte Kanal 0
2	High-Byte Kanal 1
3	Low-Byte Kanal 1
4	High-Byte Kanal 2
5	Low-Byte Kanal 2
6	High-Byte Kanal 3
7	Low-Byte Kanal 3

**Parametrierdaten**

Für die Parametrierung stehen 6Byte Parametrierdaten zur Verfügung. Die Parametrierdaten werden permanent abgelegt und bleiben auch im ausgeschalteten Zustand erhalten.

Die nachfolgende Tabelle zeigt den Aufbau der Parametrierdaten:

*Parameterbereich:*

Byte	Bit 7 ... Bit 0
0	Diagnosealarm-Byte: Bit 0 ... 5: reserviert Bit 6: 0: Diagnosealarm gesperrt 1: Diagnosealarm freigegeben Bit 7: reserviert
1	reserviert
2	Funktions-Nr. Kanal 0
3	Funktions-Nr. Kanal 1
4	Funktions-Nr. Kanal 2
5	Funktions-Nr. Kanal 3

**Parameter***Diagnosealarm*

Mit Bit 6 von Byte 0 können Sie den Diagnosealarm freigeben. Im Fehlerfall werden an Ihr übergeordnetes System 4 Diagnose-Bytes geschickt.

*Funktions-Nr.*

Tragen Sie hier für jeden Kanal die Funktions-Nummer Ihrer Ausgabe-funktion ein. Die Zuordnung der Funktions-Nummer zu einer Ausgabe-funktion entnehmen Sie bitte der Funktions-Nr. Zuordnung.

**Diagnosedaten**

Sobald Sie die Alarmfreigabe in Byte 0 des Parameterbereichs aktiviert haben, werden im Fehlerfall 4 Diagnose-Bytes mit fester Belegung an das übergeordnete System übergeben. Bitte beachten Sie, dass für die Diagnose nur die ersten zwei Bytes verwendet werden. Die restlichen zwei Bytes werden nicht benutzt.

Die Diagnose-Bytes haben folgende Belegung:

*Diagnosedaten:*

Byte	Bit 7 ... Bit 0	Default
0	Bit 0: Störung im Modul Bit 1: konstant 0 Bit 2: Fehler extern Bit 3: Kanalfehler vorhanden Bit 4 ... 7: reserviert	-
1	Bit 0 ... 3: Modulkasse 0101 Analogmodul Bit 4: Kanalinformation vorhanden	-
2	nicht belegt	-
3	nicht belegt	-

**Technische Daten**

Elektrische Daten	VIPA 232-1BD50
Anzahl der Ausgänge	4
Spannungsbereich	$\pm 10V$ , 1 ... 5V, 0 ... 10V
Strombereich	$\pm 20mA$ , 4 ... 20mA, 0 ... 20mA
Aktorwiderstand	min. $1k\Omega$ (Spannungsbereich) max. $500\Omega$ (Strombereich)
Kurzschlussstrom	70mA
Spannungsversorgung	5V über Rückwandbus 24V $\pm 20\%$ extern
Stromaufnahme	Rückwandbus: 30mA DC 24V extern: 200mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	8Byte (1 Wort pro Kanal)
Parameterdaten	6Byte
Diagnosedaten	4Byte
Maße und Gewicht	
Abmessungen (BxHxT)	25,4x76x76mm
Gewicht	100g

## AO 4x12Bit f, Multi-Output

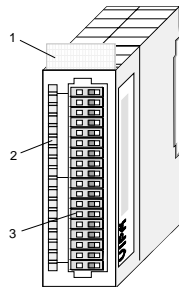
**Bestelldaten** AO 4x12Bit f, Multi-Output VIPA 232-1BD60

**Beschreibung** Das Modul besitzt 4 schnelle (f=fast) Ausgänge, deren Funktion Sie einzeln parametrieren können. Im Prozessabbild belegt das Modul insgesamt 8Byte Ausgangsdaten (2Byte pro Kanal).

Die Kanäle auf dem Modul sind gegenüber dem Rückwandbus und der ext. Versorgungsspannung mittels DC/DC-Wandlern und Optokopplern galvanisch getrennt. Das Modul ist extern mit 24V DC zu versorgen.

- Eigenschaften**
- die Zykluszeit ist bei Einsatz aller 4 Kanäle  $< 600\mu\text{s}$
  - 4 Ausgänge, deren Massen verbunden sind
  - für jeden Ausgang einzeln parametrierbare Funktionalität
  - Geeignet für Aktoren mit den Eingängen  $\pm 10\text{V}$ ,  $0 \dots 10\text{V}$ ,  $-10\text{V} \dots 0$ ,  $0 \dots 20\text{mA}$
  - Diagnose-LED und Diagnosefunktion
  - Auflösung 12Bit

**Aufbau**



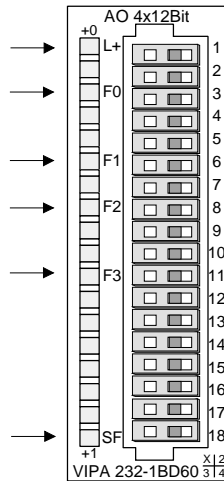
- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] LED-Statusanzeige
- [3] Steckerleiste



**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

- L+** LED (gelb)  
Versorgungsspannung liegt an
- F0...F3** LED (grün)  
für aktivierte Kanäle
- SF** LED (rot)  
Sammelmeldung Fehler

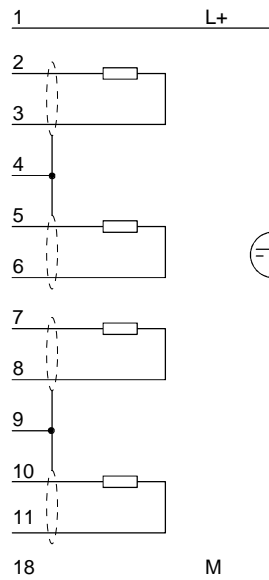


**Pin Belegung**

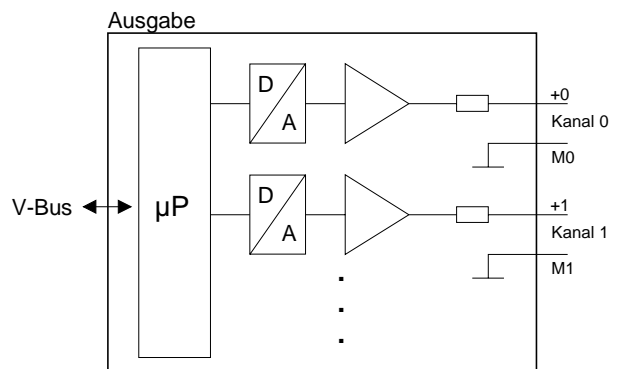
- 1** Versorgungsspg. DC 24V
- 2** + Kanal 0
- 3** Masse Kanal 0
- 4** Schirm Kanal 0, 1
- 5** + Kanal 1
- 6** Masse Kanal 1
- 7** + Kanal 2
- 8** Masse Kanal 2
- 9** Schirm Kanal 2, 3
- 10** + Kanal 3
- 11** Masse Kanal 3
- 12...17** nicht belegt
- 18** Versorgungsspg. Masse

**Anschlussbild und  
Prinzipschaltbild**

**Anschlussbild**



**Prinzipschaltbild**



**Funktions-Nr.  
Zuordnung**

Die Zuweisung einer Funktions-Nr. zu einem Kanal erfolgt über die Parametrierung. Mit der Funktions-Nr. 00h wird die in den permanent abgelegten Parametrierdaten enthaltene Funktions-Nr. nicht beeinflusst.

Durch Angabe von FFh können Sie den entsprechenden Kanal deaktivieren.

Nr.	Funktion	Ausgabebereich
00h	keinen Einfluss auf die permanent abgelegten Parametrierdaten	
01h	Spannung $\pm 10V$ S5-Format von Siemens	$\pm 12,5V$ 12,5V = Ende Übersteuerungsbereich (20480) -10...10V = Nennbereich (-16384...16384) -12,5V = Ende Untersteuerungsbereich (-20480)
05h	Spannung 0...10V S5-Format von Siemens	0...12,5V 12,5V = Ende Übersteuerungsbereich (20480) 0...10V = Nennbereich (0...16384) kein Untersteuerungsbereich
06h	Strom 0...20mA S5-Format von Siemens	0...25mA 25mA = Ende Übersteuerungsbereich (20480) 0...20mA = Nennbereich (0...16384) kein Untersteuerungsbereich
07h	Spannung -10...0V S5-Format von Siemens	-12,5V...0V -12,5V = Ende Untersteuerungsbereich (-20480) -10V...0 = Nennbereich (-16384...0) kein Übersteuerungsbereich
09h	Spannung $\pm 10V$ S7-Format von Siemens (Zweierkomplement)	$\pm 11,76V$ 11,76V = Ende Übersteuerungsbereich (32511) -10V...10V = Nennbereich (-27648...27648) -11,76V = Ende Untersteuerungsbereich (-32512)
0Dh	Spannung 0...10V S7-Format von Siemens (Zweierkomplement)	0...11,76V 11,76V = Ende Übersteuerungsbereich (32511) 0...10V = Nennbereich (0...27648) kein Untersteuerungsbereich
0Eh	Strom 0...20mA S7-Format von Siemens (Zweierkomplement)	0...23,52mA 23,52mA = Ende Übersteuerungsbereich (32511) 0...20mA = Nennbereich (0...27648) kein Untersteuerungsbereich
0Fh	Spannung -10...0V S7-Format von Siemens (Zweierkomplement)	-11,76V...0V -11,76V = Ende Untersteuerungsbereich (-32512) -10V...0V = Nennbereich (-27648...0) kein Übersteuerungsbereich
FFh	Kanal nicht aktiv (abgeschaltet)	

**Hinweis!**

Im Über- bzw. Untersteuerungsbereich wird in allen Modi der Wert 0 ausgegeben. Der Innenwiderstand der Kalibrierungsanordnung für Spannungsbereiche beträgt 2,7k $\Omega$ . Der Innenwiderstand der Kalibrierungsanordnung für Strombereiche beträgt 35 $\Omega$ .

Das Modul ist ab Werk auf den Messbereich "Spannung  $\pm 10V$ " im S7-Format von Siemens eingestellt.

**Zahlendarstellung im S5-Format von Siemens**

Die Eingabedaten werden im S5-Format von Siemens in einem Wort abgelegt. Das Wort setzt sich zusammen aus dem binären Wert und den Informationsbits.

*Zahlendarstellung:*

Byte	Bit 7 ... Bit 0
0	Bit 0: Überlaufbit 0: Wert liegt im Messbereich 1: Messbereich überschritten Bit 1: Fehlerbit (gesetzt bei internem Fehler) Bit 2: Tätigkeitsbit (immer 0) Bit 3 ... 7: Binärer Messwert
1	Bit 0 ... 6: Binärer Messwert Bit 7: Vorzeichen 0: positiv 1: negativ

**+/- 10V**

Spannung	Dezimal	Hex
-10V	-16384	C000
-5V	-8192	E000
0V	0	0000
+5V	8192	2000
+10V	+16384	4000

Formeln für die Berechnung:

$$\text{Wert} = 16384 \cdot \frac{U}{10}, \quad U = \text{Wert} \cdot \frac{10}{16384}$$

U: Spannungswert, Wert: Dezimalwert

**0...10V**

Spannung	Dezimal	Hex
0V	0	0000
5V	8192	2000
10V	16384	4000

Formeln für die Berechnung:

$$\text{Wert} = 16384 \cdot \frac{U}{10}, \quad U = \text{Wert} \cdot \frac{10}{16384}$$

U: Spannungswert, Wert: Dezimalwert

**-10...0V**

Spannung	Dezimal	Hex
-10V	-16384	C000
-5V	-8192	E000
0V	0	0000

Formeln für die Berechnung:

$$\text{Wert} = 16384 \cdot \frac{U}{10}, \quad U = \text{Wert} \cdot \frac{10}{16384}$$

U: Spannungswert, Wert: Dezimalwert

**0....20mA**

Strom	Dezimal	Hex
+0mA	0	0000
+10mA	+8192	2000
+20mA	+16384	4000

Formeln für die Berechnung:

$$\text{Wert} = 16384 \cdot \frac{I - 4}{16}, \quad I = \text{Wert} \cdot \frac{16}{16384} + 4$$

I: Stromwert, Wert: Dezimalwert

**Zahlendarstellung im S7-Format von Siemens**

Die Darstellung des Analogwertes erfolgt im Zweierkomplement.

Zahlendarstellung:

Byte	Bit 7 ... Bit 0
0	Bit 0 ... 7: Binärer Messwert
1	Bit 0 ... 6: Binärer Messwert Bit 7: Vorzeichen 0: positiv 1: negativ

+/- 10V

Spannung	Dezimal	Hex
-10V	-27648	9400
-5V	-13824	CA00
0V	0	0
+5V	13824	3600
+10V	+27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U}{10}, \quad U = Wert \cdot \frac{10}{27648}$$

U: Spannungswert, Wert: Dezimalwert

0...10V

Spannung	Dezimal	Hex
0V	0	0000
5V	13824	3600
10V	27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U}{10}, \quad U = Wert \cdot \frac{10}{27648}$$

U: Spannungswert, Wert: Dezimalwert

-10...0V

Spannung	Dezimal	Hex
-10V	-27648	9400
-5V	-13824	CA00
0V	0	0000

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{U}{10}, \quad U = Wert \cdot \frac{10}{27648}$$

U: Spannungswert, Wert: Dezimalwert

0....20mA

Strom	Dezimal	Hex
+0mA	0	0
+10mA	+13824	3600
+20mA	+27648	6C00

Formeln für die Berechnung:

$$Wert = 27648 \cdot \frac{I - 4}{16}, \quad I = Wert \cdot \frac{16}{27648} + 4$$

I: Stromwert, Wert: Dezimalwert

**Datenausgabe**

Die auszugebenden Werte tragen Sie im Datenausgabebereich ein. Die Zuordnung Ihres Ausgabewertes zu einem entsprechenden Strom- bzw. Spannungswert können Sie für jeden Kanal über die Funktions-Nr. parametrieren.

Den Aufbau des Datenausgabebereichs sehen Sie in der nachfolgenden Tabelle:

*Datenausgabebereich:*

Byte	Bit 7 ... Bit 0
0	High-Byte Kanal 0
1	Low-Byte Kanal 0
2	High-Byte Kanal 1
3	Low-Byte Kanal 1
4	High-Byte Kanal 2
5	Low-Byte Kanal 2
6	High-Byte Kanal 3
7	Low-Byte Kanal 3

**Parametrierdaten**

Für die Parametrierung stehen 6Byte Parametrierdaten zur Verfügung. Die Parametrierdaten werden permanent abgelegt und bleiben auch im ausgeschalteten Zustand erhalten.

Die nachfolgende Tabelle zeigt den Aufbau der Parametrierdaten:

*Parameterbereich:*

Byte	Bit 7 ... Bit 0
0	Diagnosealarm-Byte: Bit 0 ... 5: reserviert Bit 6: 0: Diagnosealarm gesperrt 1: Diagnosealarm freigegeben Bit 7: reserviert
1	reserviert
2	Funktions-Nr. Kanal 0
3	Funktions-Nr. Kanal 1
4	Funktions-Nr. Kanal 2
5	Funktions-Nr. Kanal 3

**Parameter**

*Diagnosealarm*

Mit Bit 6 von Byte 0 können Sie den Diagnosealarm freigeben. Im Fehlerfall werden an Ihr übergeordnetes System 4 Diagnose-Bytes geschickt.

*Funktions-Nr.*

Tragen Sie hier für jeden Kanal die Funktions-Nummer Ihrer Ausgabe-funktion ein. Die Zuordnung der Funktions-Nummer zu einer Ausgabe-funktion entnehmen Sie bitte der Funktions-Nr. Zuordnung.

**Diagnosedaten**

Sobald Sie die Alarmfreigabe in Byte 0 des Parameterbereichs aktiviert haben, werden im Fehlerfall 4 Diagnose-Bytes mit fester Belegung an das übergeordnete System übergeben. Bitte beachten Sie, dass für die Diagnose nur die ersten zwei Bytes verwendet werden. Die restlichen zwei Bytes werden nicht benutzt.

Die Diagnose-Bytes haben folgende Belegung:

*Diagnosedaten:*

Byte	Bit 7 ... Bit 0	Default
0	Bit 0: Störung im Modul Bit 1: reserviert Bit 2: Fehler extern Bit 3: Kanalfehler vorhanden Bit 4: Externe Hilfsspannung fehlt Bit 5: Drahtbruch bei Stromausgabe Kurzschluss bei Spannungsausgabe Bit 6: reserviert Bit 7: Falsche Parameter in Modul	-
1	Bit 0 ... 3: Modulklasse 0101 Analogmodul Bit 4: Kanalinformation vorhanden	-
2	nicht belegt	-
3	nicht belegt	-

**Technische Daten**

Elektrische Daten	VIPA 232-1BD60
Anzahl der Ausgänge	4
Zykluszeit (alle 4 Kanäle)	< 600µs
Spannungsbereich	±10V, 0 ... 10V, -10 ... 0V
Strombereich	0 ... 20mA
Aktorwiderstand	min. 500Ω (Spannungsbereich) max. 500Ω (Strombereich)
Kurzschlussstrom	30mA
Spannungsversorgung	5V über Rückwandbus 24V ±20% extern
Stromaufnahme	Rückwandbus: 50mA DC 24V extern: 200mA
Potenzialtrennung	500Veff (Feldspannung-Rückwandbus)
Statusanzeige	über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten	-
Ausgabedaten	8Byte (1 Wort pro Kanal)
Parameterdaten	6Byte
Diagnosedaten	4Byte
Maße und Gewicht	
Abmessungen (BxHxT)	25,4x76x76mm
Gewicht	100g





## Teil 18 Analoge Ein-/Ausgabe-Module

### Überblick

Inhalt dieses Kapitels ist der Aufbau und die Funktionsweise der analogen Ein-/Ausgabe-Module von VIPA.

Nachfolgend sind beschrieben:

- Systemübersicht der analogen Ein-/Ausgabe-Module
- Eigenschaften
- Aufbau
- Anschluss- und Prinzipschaltbild
- Parametrierdaten
- Funktionsnummernzuordnung
- Technische Daten

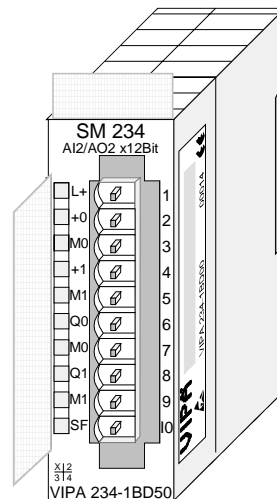
### Inhalt

Thema	Seite
<b>Teil 18 Analoge Ein-/Ausgabe-Module</b> .....	<b>18-1</b>
Systemübersicht .....	18-2
Sicherheitshinweis zur Bereichseinstellung.....	18-2
Allgemeines .....	18-3
AI 2/AO 2x12Bit, Multi-In-/Output.....	18-4

## Systemübersicht

### Ein-/Ausgabe-Module SM 234

Nachfolgend finden Sie eine Übersicht der bei VIPA zur Zeit erhältlichen analogen Ein-/Ausgabe-Module:



### Bestelldaten Ein-/Ausgabe- Module

Typ	Bestellnummer
AIO 2x12Bit, Multi In/Output	VIPA 234-1BD50

## Sicherheitshinweis zur Bereichseinstellung



### Achtung!

Bitte beachten Sie, dass das hier vorgestellte Modul keine hardwareseitige Absicherung gegen Falschparametrierung besitzt. Die Vorgabe des entsprechenden Mess- bzw. Ausgabe-Bereichs erfolgt ausschließlich bei der Projektierung.

Beispielsweise kann das Modul einen Defekt bekommen, sobald Sie bei projektierter Strommessung fälschlicherweise eine Spannung anlegen.

Bei der Projektierung des Moduls ist äußerste Sorgfalt geboten.

## Allgemeines

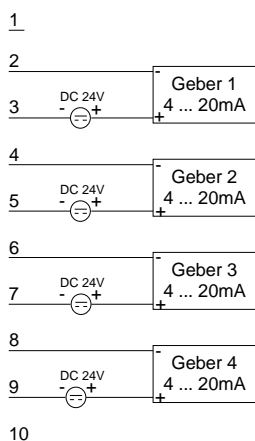
### Leitungen für Analogsignale

Für die Analogsignale sollten Sie geschirmte und paarweise verdrehte Leitungen verwenden. Hierdurch verringern Sie die Störbeeinflussung. Den Schirm der Analogleitungen sollten Sie an beiden Leitungsenden erden. Wenn Potenzialunterschiede zwischen den Leitungsenden bestehen, kann ein Potenzialausgleichstrom fließen, der die Analogsignale stören könnte. In diesem Fall sollten Sie den Schirm nur an einem Leitungsende erden.

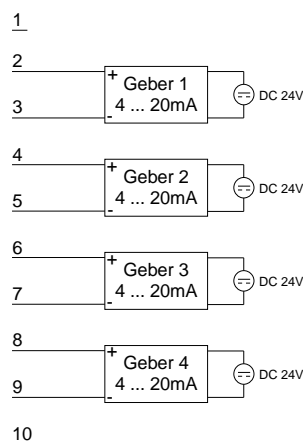
### Anschließen von Messwertgebern

Die analogen Eingabe-Module bieten vielfältige Anschlussmöglichkeiten für 2-Draht- und 4-Draht-Messumformer. Bitte beachten Sie, dass die Messumformer extern zu versorgen sind. Schleifen Sie bei 2-Draht-Messumformern eine externe Spannungsversorgung in Ihre Messleitung ein. Folgende Abbildung soll den Anschluss von 2- und 4-Draht-Messumformern verdeutlichen:

#### 2-Draht-Anschluss



#### 4-Draht-Anschluss



### Anschließen von Lasten und Aktoren

Da auch die Aktoren extern zu versorgen sind, können Sie Aktoren in 2- und in 4-Draht-Technik anschließen. Bei der Ausgabe von Stellwerten an 2-Draht-Aktoren ist eine Spannungsquelle in die Steuerleitung einzuschleifen. 4-Draht-Aktoren sind extern zu versorgen.



#### Hinweis!

Bitte achten Sie beim Anschluss der Aktoren immer auf richtige Polarität! Lassen Sie die Ausgangsklemmen der nicht benutzten Kanäle unbeschaltet!

### Parametrierung und Diagnose zur Laufzeit

Durch Einsatz der SFCs 55, 56 und 57 können Sie zur Laufzeit über die CPU 21x die Parameter in Ihren Analog-Modulen ändern. Zur Diagnoseauswertung zur Laufzeit stehen Ihnen die beiden SFCs 51 und 59 zur Verfügung. Hiermit können Sie detaillierte Diagnoseinformationen abrufen und auf die Diagnose reagieren.

## AI 2/AO 2x12Bit, Multi-In-/Output

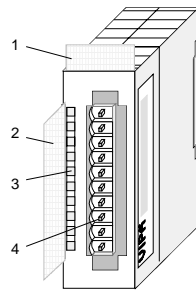
**Bestelldaten** AI 2/AO 2x12Bit Multi-In-/Output VIPA 234-1BD50

**Beschreibung** Das Modul besitzt 2 analoge Eingänge und 2 analoge Ausgänge, deren Funktion Sie einzeln parametrieren können. Im Prozessabbild belegt das Modul 4Byte Eingabe- und 4Byte Ausgabedaten

Die Kanäle auf dem Modul sind gegenüber dem Rückwandbus und der ext. Versorgungsspannung mittels DC/DC-Wandlern und Optokopplern galvanisch getrennt. Es ist extern mit DC 24V zu versorgen.

- Eigenschaften**
- 2 Eingänge und 2 Ausgänge, deren Masse verbunden ist
  - für jeden Ein-/Ausgang einzelparametrierbare Funktionalität
  - Geeignet für Geber bzw. Aktoren mit den Ein- bzw. Ausgangsbereichen: ±10V, 1 ... 5V, 0 ... 10V, ±20mA, 0 ... 20mA oder 4 ... 20mA
  - Diagnose-LED

**Aufbau**

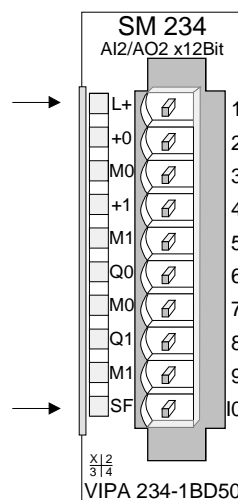


- [1] Beschriftungsstreifen für Modulbezeichnung
- [2] Beschriftungsstreifen für Bit-adresse mit Beschreibung
- [3] LED-Statusanzeige
- [4] Steckerleiste

**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

- L+ LED (gelb)  
Versorgungsspannung liegt an
- SF Sammelfehler LED (rot)  
Leuchtet sobald bei einem der Kanäle ein Fehler auftritt bzw. ein Eintrag in den Diagnosebytes stattgefunden hat.




Pin	Belegung
1	Versorgung.spg. + DC 24V
2	pos. Anschluss E.0
3	Masse Kanal 0
4	pos. Anschluss E.1
5	Masse Kanal 1
6	pos. Anschluss A.2
7	Masse Kanal 2
8	pos. Anschluss A.3
9	Masse Kanal 3
10	Versorgungsspg. Masse

**Anschlussbild und  
Prinzipschaltbild****Anschlussbild**

1 L+ DC24V


2

3 IN 


4

5 IN 

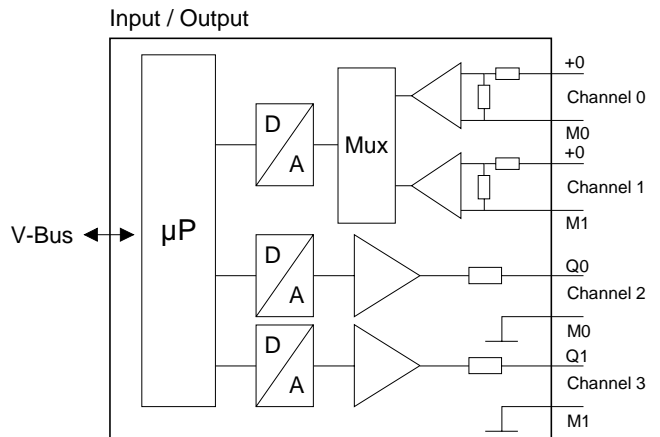
6

7 OUT 

8

9 OUT 

10 M

**Prinzipschaltbild****Achtung!**

Vorübergehend nicht benutzte Eingänge sind bei aktiviertem Kanal mit der zugehörigen Masse zu verbinden. Sind nicht benutzte Kanäle über FFh abgeschaltet, ist diese Maßnahme nicht erforderlich.

**Hinweis!**

Jeder Kanal ist einzeln parametrierbar. Für die Parametrierung stehen 12Byte Parametrierdaten zur Verfügung. Sie werden permanent abgelegt und bleiben auch im ausgeschalteten Zustand erhalten.

**Funktions-Nr.  
Zuordnung  
Eingabe-Bereich  
(Kanal 0, Kanal 1)**

Die Zuweisung einer Funktions-Nr. zu einem Kanal erfolgt über die Parametrierung. Mit der Funktions-Nr. 00h wird die in den permanent abgelegten Parametrierdaten enthaltene Funktions-Nr. nicht beeinflusst.

Durch Angabe von FFh können Sie den entsprechenden Kanal deaktivieren.



**Hinweis!**

Bei Überschreitung des Übersteuerungsbereichs erfolgt die Ausgabe des Werts 7FFFh (32767) und bei Unterschreitung des Untersteuerungsbereichs der Wert 8000h (-32768).

Nr.	Funktion	Messbereich / Darstellung
00h	Die Funktions-Nr. in den permanent abgelegten Parametrierdaten wird nicht geändert.	
3Bh	Spannung ±10V S5-Format von Siemens	±12,5V / 12,5V = Ende Übersteuerungsbereich (20480) -10...10V = Nennbereich (-16384...16384) -12,5V = Ende Untersteuerungsbereich (-20480) Zweierkomplement
2Bh	Spannung ±10V S5-Format von Siemens	±12,5V / 12,5V = Ende Übersteuerungsbereich (20480) -10...10V = Nennbereich (-16384...16384) -12,5V = Ende Untersteuerungsbereich (-20480) Betrag und Vorzeichen
72h	Spannung 1...5V S5-Format von Siemens	0...6V 6V = Ende Übersteuerungsbereich (20480) 1...5V = Nennbereich (0...16384) 0V = Ende Untersteuerungsbereich (-4096) Betrag und Vorzeichen
75h	Spannung 0...10V S5-Format von Siemens	0...12,5V 12,5V = Ende Übersteuerungsbereich (20480) 0...10V = Nennbereich (0...16384) kein Untersteuerungsbereich
28h	Spannung ±10V S7-Format von Siemens	±11,76V / 11,76V= Ende Übersteuerungsbereich (32511) -10...10V= Nennbereich (-27648...27648) -11,76V= Ende Untersteuerungsbereich (-32512) Zweierkomplement
7Ah	Spannung 1...5V S7-Format von Siemens	0...5,704V 5,704V = Ende Übersteuerungsbereich (32511) 1...5V = Nennbereich (0...27648) 0V = Ende Untersteuerungsbereich (-6912) Zweierkomplement
7Dh	Spannung 0...10V S7-Format von Siemens	0...11,76V 11,76V = Ende Übersteuerungsbereich (32511) 0...10V = Nennbereich (0...27648) kein Untersteuerungsbereich

... Fortsetzung

... Fortsetzung Funktions-Nr. Zuordnung Eingabe (Kanal 0, Kanal 1)

3Ah	Strom $\pm 20\text{mA}$ S5-Format von Siemens	$\pm 25,0\text{mA}$ / 25,0mA = Ende Übersteuerungsbereich (20480) -20...20mA = Nennwert (-16384...16384) -25,0mA = Ende Untersteuerungsbereich (-20480) Zweierkomplement
2Fh	Strom $\pm 20\text{mA}$ S5-Format von Siemens	$\pm 25,0\text{mA}$ / 25,0mA = Ende Übersteuerungsbereich (20480) -20...20mA = Nennwert (-16384...16384) -25,0mA = Ende Untersteuerungsbereich (-20480) Betrag und Vorzeichen
2Eh	Strom 4...20mA S5-Format von Siemens	0,8...+24,0mA / 24,0mA = Ende Übersteuerungsbereich (20480) 4 ... 20mA = Nennbereich (0...16384) 0,8mA = Ende Untersteuerungsbereich (-3277) Betrag und Vorzeichen
76h	Strom 0...20mA S5-Format von Siemens	0...25mA 25mA = Ende Übersteuerungsbereich (20480) 0...20mA = Nennbereich (0...16384) kein Untersteuerungsbereich
2Ch	Strom $\pm 20\text{mA}$ S7-Format von Siemens	$\pm 23,51\text{mA}$ / 23,51mA = Ende Übersteuerungsbereich (32511) -20...20mA = Nennwert (-27648...27648) -23,51mA = Ende Untersteuerungsbereich (-32512) Zweierkomplement
2Dh	Strom 4...20mA S7-Format von Siemens	1,185...+22,81mA / 22,81mA = Ende Übersteuerungsbereich (32511) 4...20mA = Nennbereich (0...27648) 1,18mA = Ende Untersteuerungsbereich (-4864) Zweierkomplement
7Eh	Strom 0...20mA S7-Format von Siemens	0...23,52mA 23,52mA = Ende Übersteuerungsbereich (32511) 0...20mA = Nennbereich (0...27648) kein Untersteuerungsbereich
FFh	Kanal nicht aktiv (abgeschaltet)	



### Hinweis!

Das Modul ist ab Werk auf den Messbereich "Spannung  $\pm 10\text{V}$ " im S7 Format von Siemens eingestellt.

**Funktions-Nr.  
Zuordnung  
Ausgabe-Bereich  
(Kanal 2, Kanal 3)**

Die Zuweisung einer Funktions-Nr. zu einem Kanal erfolgt über die Parametrierung. Mit der Funktions-Nr. 00h wird die in den permanent abgelegten Parametrierdaten enthaltene Funktions-Nr. nicht beeinflusst.

Durch Angabe von FFh können Sie den entsprechenden Kanal deaktivieren.



**Hinweis!**

Beim Verlassen des definierten Bereichs wird 0V bzw. 0A ausgegeben!

Nr.	Funktion	Ausgabebereich
00h	Die Funktions-Nr. in den permanent abgelegten Parametrierdaten wird nicht geändert.	
01h	Spannung $\pm 10V$ S5-Format von Siemens	$\pm 12,5V$ 12,5V = Ende Übersteuerungsbereich (20480) -10...10V = Nennbereich (-16384...16384) -12,5V = Ende Untersteuerungsbereich (-20480)
02h	Spannung 1...5V S5-Format von Siemens	0...6V 6V = Ende Übersteuerungsbereich (20480) 1...5V = Nennbereich (0...16384) 0V = Ende Untersteuerungsbereich (-4096)
05h	Spannung 0...10V S5-Format von Siemens	0...12,5V 12,5V = Ende Übersteuerungsbereich (20480) 0...10V = Nennbereich (0...16384) kein Untersteuerungsbereich
09h	Spannung $\pm 10V$ S7-Format von Siemens	$\pm 11,76V$ 11,76V = Ende Übersteuerungsbereich (32511) -10V...10V = Nennbereich (-27648...27648) -11,76 = Ende Untersteuerungsbereich (-32512) Zweierkomplement
0Ah	Spannung 1...5V S7-Format von Siemens	0...5,704V 5,704V = Ende Übersteuerungsbereich (32511) 1...5V = Nennbereich (0...27648) 0V = Ende Untersteuerungsbereich (-6912) Zweierkomplement
0Dh	Spannung 0...10V S7-Format von Siemens	0...11,76V 11,76V = Ende Übersteuerungsbereich (32511) 0...10V = Nennbereich (0...27648) kein Untersteuerungsbereich
03h	Strom $\pm 20mA$ S5-Format von Siemens	$\pm 25mA$ 25mA = Ende Übersteuerungsbereich (20480) -20...20mA = Nennbereich (-16384...16384) -25mA = Ende Untersteuerungsbereich (-20480)

Fortsetzung ...



... Fortsetzung Funktions-Nr. Zuordnung Ausgabe-Bereich (Kanal 2, Kanal 3)

04h	Strom 4...20mA S5-Format von Siemens	0...24mA 24mA = Ende Übersteuerungsbereich (20480) 4...20mA = Nennbereich (0...16384) 0mA = Ende Untersteuerungsbereich (-4096)
06h	Strom 0...20mA S5-Format von Siemens	0...25mA 25mA = Ende Übersteuerungsbereich (20480) 0...20mA = Nennbereich (0...16384) kein Untersteuerungsbereich
0Bh	Strom $\pm$ 20mA S7-Format von Siemens	$\pm$ 23,52mA 23,52mA = Ende Übersteuerungsbereich (32511) -20...20mA = Nennbereich (-27648...27648) -23,52mA = Ende Untersteuerungsbereich (-32512) Zweierkomplement
0Ch	Strom 4...20mA S7-Format von Siemens	0...22,81mA 22,81mA = Ende Übersteuerungsbereich (32511) 4...20mA = Nennbereich (0...27648) 0mA = Ende Untersteuerungsbereich (-6912) Zweierkomplement
0Eh	Strom 0...20mA S7-Format von Siemens	0...23,52mA 23,52mA = Ende Übersteuerungsbereich (32511) 0...20mA = Nennbereich (0...27648) kein Untersteuerungsbereich
FFh	Kanal nicht aktiv (abgeschaltet)	

**Zahlendarstellung im S5-Format von Siemens**

Die Eingabe- und Ausgabedaten werden im S5-Format von Siemens in einem Wort abgelegt. Das Wort setzt sich zusammen aus dem binären Wert und den Informationsbits.

*Zahlendarstellung:*

Byte	Bit 7 ... Bit 0
0	Bit 0: Überlaufbit 0: Wert liegt im Messbereich 1: Messbereich überschritten Bit 1: Fehlerbit (gesetzt bei internem Fehler) Bit 2: Tätigkeitsbit (immer 0) Bit 3 ... 7: Binärer Messwert
1	Bit 0 ... 6: Binärer Messwert Bit 7: Vorzeichen 0: positiv 1: negativ

**+/- 10V (Zweierkomplement)**

Spannung	Dezimal	Hex
-10V	-16384	C000
-5V	-8192	E000
0V	0	0000
+5V	+8192	2000
+10V	+16384	4000

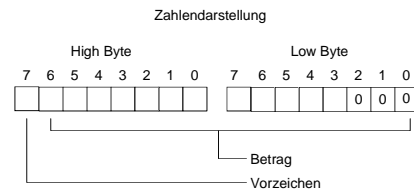
**Formeln für die Berechnung:**

$$Wert = 16384 \cdot \frac{U}{10}, \quad U = Wert \cdot \frac{10}{16384}$$

U: Spannungswert, Wert: Dezimalwert

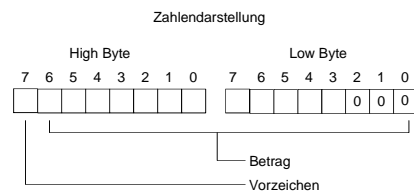
**+/- 10V (Betrag und Vorzeichen)**

Spannung	Dezimal	Hex
-10V	-16384	C000
-5V	-8192	A000
0V	0	0000
+5V	+8192	2000
+10V	+16384	4000



**4...20mA / 1 ... 5V (Betrag und Vorzeichen)**

Strom/Spannung	Dezimal	Hex
+4mA / 1V	0	0000
+12mA / 3V	+8192	2000
+20mA / 5V	+16384	4000



**+/- 20mA (Zweierkomplement)**

Strom	Dezimal	Hex
-20mA	-16384	C000
-10mA	-8192	E000
0mA	0	0000
+10mA	+8192	2000
+20mA	+16384	4000

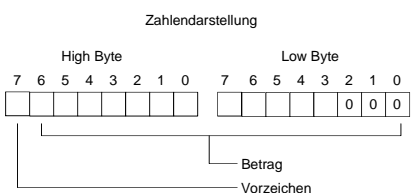
**Formeln für die Berechnung:**

$$Wert = 16384 \cdot \frac{I}{20}, \quad I = Wert \cdot \frac{20}{16384}$$

I: Stromwert, Wert: Dezimalwert

**+/- 20mA (Betrag und Vorzeichen)**

Strom	Dezimal	Hex
-20mA	-16384	C000
-10mA	-8192	A000
0mA	0	0000
+10mA	+8192	2000
+20mA	+16384	4000



**Zahlendarstellung  
im S7-Format von  
Siemens**

Die Darstellung des Analogwertes erfolgt im Zweierkomplement

**Zahlendarstellung:**

Byte	Bit 7 ... Bit 0
0	Bit 0 ... 7: Binärer Messwert
1	Bit 0 ... 6: Binärer Messwert Bit 7: Vorzeichen 0: positiv 1: negativ

**+/- 10V**

Spannung	Dezimal	Hex
-10V	-27648	9400
-5V	-13824	CA00
0V	0	0
+5V	13824	3600
+10V	+27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{U}{10}, \quad U = \text{Wert} \cdot \frac{10}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**0...10V**

Spannung	Dezimal	Hex
0V	0	0000
5V	13824	3600
10V	27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{U}{10}, \quad U = \text{Wert} \cdot \frac{10}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**1...5V**

Spannung	Dezimal	Hex
+1V	0	0
+3V	+13824	3600
+5V	+27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{U-1}{4}, \quad U = \text{Wert} \cdot \frac{4}{27648} + 1$$

U: Spannungswert, Wert: Dezimalwert

**+/-4V**

Spannung	Dezimal	Hex
-4V	-27648	9400
0V	0	0
4V	27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{U}{4}, \quad U = \text{Wert} \cdot \frac{4}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**+/-400mV**

Spannung	Dezimal	Hex
-400mV	-27648	9400
0V	0	0
400mV	27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{U}{400}, \quad U = \text{Wert} \cdot \frac{400}{27648}$$

U: Spannungswert, Wert: Dezimalwert

**4...20mA**

Strom	Dezimal	Hex
+4mA	0	0
+12mA	+13824	3600
+20mA	+27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{I-4}{16}, \quad I = \text{Wert} \cdot \frac{16}{27648} + 4$$

I: Stromwert, Wert: Dezimalwert

**+/- 20mA**

Strom	Dezimal	Hex
-20mA	-27648	9400
-10mA	-13824	CA00
0mA	0	0
+10mA	+13824	3600
+20mA	+27648	6C00

Formeln für die Berechnung:

$$\text{Wert} = 27648 \cdot \frac{I}{20}, \quad I = \text{Wert} \cdot \frac{20}{27648}$$

I: Stromwert, Wert: Dezimalwert

**Dateneingabe-  
/Datenausgabe-  
Bereich**

Für Ein- und Ausgabe-Bereich stehen Ihnen 4Byte zur Verfügung, die folgende Belegung haben:

*Dateneingabebereich:*

Während der Messung werden die Messdaten im Dateneingabebereich abgelegt.

Byte	Bit 7 ... Bit 0
0	High-Byte Kanal 0
1	Low-Byte Kanal 0
2	High-Byte Kanal 1
3	Low-Byte Kanal 1

**Hinweis!**

Bei Dreileiter- bzw. Vierleitermessung wird nur Kanal 0 verwendet.

*Datenausgabebereich*

Zur Ausgabe tragen Sie einen Wert im Datenausgabebereich ein.

Byte	Bit 7 ... Bit 0
0	High-Byte Kanal 2
1	Low-Byte Kanal 2
2	High-Byte Kanal 3
3	Low-Byte Kanal 3

**Parametrierdaten**

Für die Parametrierung stehen 12Byte Parametrierdaten zur Verfügung. Die Parametrierdaten werden permanent abgelegt und bleiben auch im ausgeschalteten Zustand erhalten.

Die nachfolgende Tabelle zeigt den Aufbau der Parametrierdaten:

*Parameterbereich:*

Byte	Bit 7 ... Bit 0	Default
0	Drahtbruchkennung und Diagnose: Bit 0: Drahtbruchkennung Kanal 0 0: deaktiviert 1: aktiviert Bit 1: Drahtbruchkennung Kanal 1 0: deaktiviert 1: aktiviert Bit 2 ... 5: reserviert Bit 6: Diagnosealarm 0: gesperrt 1: freigegeben Bit 7: reserviert	00h
1	Bit 0: reserviert Bit 1: reserviert Bit 2: CPU-Stop-Verhalten für Kanal 2 0: Ersatzwert Kanal 2 aufschalten <sup>*)</sup> 1: Letzten Wert Kanal 2 halten Bit 3: CPU-Stop-Verhalten für Kanal 3 0: Ersatzwert Kanal 3 aufschalten 1: Letzten Wert Kanal 3 halten Bit 4 ... 7: reserviert	00h
2	Funktions-Nr. Kanal 0 (siehe Tabelle der Eingangsbereiche)	28h
3	Funktions-Nr. Kanal 1 (siehe Tabelle der Eingangsbereiche)	28h
4	Funktions-Nr. Kanal 2 (siehe Tabelle der Ausgangsbereiche)	09h
5	Funktions-Nr. Kanal 3 (siehe Tabelle der Ausgangsbereiche)	09h
6	Optionen-Byte Kanal 0	00h
7	Optionen-Byte Kanal 1	00h
8	High-Byte Ersatzwert Kanal 2	00h
9	Low-Byte Ersatzwert Kanal 2	00h
10	High-Byte Ersatzwert Kanal 3	00h
11	Low-Byte Ersatzwert Kanal 3	00h

<sup>\*)</sup> Soll bei CPU-STOP der Ausgabekanal 0A bzw. 0V ausgeben, so ist folgender Ersatzwert bei Stromausgabe (4...20mA) bzw. Spannungsausgabe (1...5V) vorzugeben: E500h für das S7-Format von Siemens und F000h für das S5-Format von Siemens.

**Parameter***Drahtbruchkennung*

Die Drahtbruchkennung kommt beim Messbereich 4 ... 20mA zum Einsatz. Über die Bits 0 und 1 von Byte 0 können Sie die Drahtbruchererkennung für die Eingabekanäle aktivieren.

Sinkt im 4...20mA-Strommessbereich der Strom unter 0,8mA, wird ein Drahtbruch erkannt. Bei aktivierter Drahtbruchkennung und Diagnosealarm erfolgt eine Diagnosemeldung an das übergeordnete System.

*Diagnosealarm*

Mit Bit 6 von Byte 0 können Sie den Diagnosealarm freigeben. Im Fehlerfall wird an Ihr übergeordnetes System der 4Byte große *Datensatz 0* übergeben. Zur erweiterten Diagnose haben Sie dann die Möglichkeit den 12Byte großen *Datensatz 1* abzurufen.

Näheres hierzu finden Sie weiter unten unter "Diagnosedaten".

*CPU-Stop-Verhalten und Ersatzwert*

Mit Bit 2 und 3 von Byte 1 und Byte 8 ... 11 können Sie je Ausgabekanal das Verhalten des Moduls bei CPU-Stop vorgeben.

Über Byte 8 ... 11 geben Sie einen Ersatzwert vor, der am Analogausgang anzuliegen hat sobald die CPU in Stop geht.

Durch Setzen von Bit 2 bzw. 3 bleibt bei CPU-Stop der letzte Ausgabe-Wert am Ausgang stehen. Ein Rücksetzen schaltet den Ersatzwert auf.

*Funktions-Nr.*

Tragen Sie hier für jeden Kanal die Funktions-Nummer Ihrer Mess- bzw. Ausgabefunktion ein. Diese können Sie der entsprechenden Funktions-Nr.-Zuordnung aus der Tabelle für den Ein- bzw. Ausgabe-Bereich entnehmen.

*Optionen-Byte*

Hier können Sie für jeden Eingabe-Kanal die Wandlergeschwindigkeit einstellen. Bitte beachten Sie, dass bei höheren Wandlergeschwindigkeiten die Auflösung aufgrund der kürzeren Integrationszeit sinkt.

Das Datenübergabeformat bleibt gleich. Lediglich die unteren Bits (LSBs) sind für den Analogwert nicht mehr aussagekräftig.

*Aufbau Option-Byte:*

Byte	Bit 7 ... Bit 0	Auflösung	Default
6 ... 7	Bit 0 ... 3: Geschwindigkeit pro Kanal		00h
	0000 15 Wandlungen/s	16	
	0001 30 Wandlungen/s	16	
	0010 60 Wandlungen/s	15	
	0011 123 Wandlungen/s	14	
	0100 168 Wandlungen/s	12	
	0101 202 Wandlungen/s	10	
	0110 3,7 Wandlungen/s	16	
	0111 7,5 Wandlungen/s	16	
	Bit 4 ... 7: reserviert		

**Diagnosedaten**

Die Diagnosedaten sind 12Byte lang und stehen in den Datensätzen 0 und 1 des Systemdatenbereichs.

Sobald Sie die Diagnosealarmfreigabe in Byte 0 des Parameterbereichs aktiviert haben, wird im Fehlerfall *Datensatz 0* an das übergeordnete System übergeben.

Der *Datensatz 0* hat eine feste Belegung und eine Länge von 4Byte. Die Inhalte von *Datensatz 0* können Sie im Diagnosefenster der CPU im Klartext ausgeben lassen.

Zur erweiterten Diagnose während der Laufzeit haben Sie auch die Möglichkeit, über die SFCs 51 und 59 den 12Byte großen *Datensatz 1* auszuwerten.

**Diagnose auswerten**

Bei anstehender Diagnose unterbricht die CPU ihr Anwenderprogramm und verzweigt in den OB82. In diesem OB können Sie durch entsprechende Programmierung mit den SFCs 51 und 59 detaillierte Diagnoseinformationen abrufen und auf die Diagnose reagieren.

Nach Abarbeitung des OB82 wird die Bearbeitung des Anwenderprogramms wieder fortgesetzt. Die Diagnosedaten sind bis zum Verlassen des OB82 konsistent.

**Datensatz 0**

*Byte 0 bis 3:*

*Datensatz 0 (Byte 0 bis 3):*

Byte	Bit 7 ... Bit 0	Default
0	Bit 0: Störung im Modul Bit 1: reserviert Bit 2: Fehler extern Bit 3: Kanalfehler vorhanden Bit 4: externe Versorgungsspannung fehlt Bit 5 ... 6: reserviert Bit 7: Falsche Parameter im Modul	00h
1	Bit 0 ... 3: Modulklasse 0101 Analogmodul Bit 4: Kanalinformation vorhanden Bit 5 ... 7: reserviert	15h
2	reserviert	00h
3	reserviert	00h

**Datensatz 1***Byte 0 bis 11:*

Der Datensatz 1 enthält die 4Byte des Datensatzes 0 und zusätzlich 8Byte modulspezifische Diagnosedaten.

Die Diagnosebytes haben folgende Belegung:

*Datensatz 1 (Byte 0 bis 11):*

Byte	Bit 7 ... Bit 0	Default
0 ... 3	Inhalte Datensatz 0 (siehe vorherige Seite)	-
4	Bit 0 ... 6: Kanaltyp 70h: Digitaleingabe 71h: Analogeingabe 72h: Digitalausgabe 73h: Analogausgabe 74h: Analogein-/ausgabe Bit 7: reserviert	74h
5	Bit 0 ... 7: Anzahl der Diagnosebits, die das Modul pro Kanal ausgibt	08h
6	Bit 0 ... 7: Anzahl der gleichartigen Kanäle eines Moduls	04h
7	Bit 0: Kanalfehler Kanal 0 Bit 1: Kanalfehler Kanal 1 Bit 2: Kanalfehler Kanal 2 Bit 3: Kanalfehler Kanal 3 Bit 4 ... 7: reserviert	00h
8	Bit 0: Drahtbruch Kanal 0 Bit 1: Parametrierfehler Kanal 0 Bit 2: Messbereichsunterschreitung Kanal 0 Bit 3: Messbereichsüberschreitung Kanal 0 Bit 4 ... 7: reserviert	00h
9	Bit 0: Drahtbruch Kanal 1 Bit 1: Parametrierfehler Kanal 1 Bit 2: Messbereichsunterschreitung Kanal 1 Bit 3: Messbereichsüberschreitung Kanal 1 Bit 4 ... 7: reserviert	00h
10	Bit 0: Drahtbruch bei Stromausgabe bzw. Kurzschluss bei Spannungsausgabe Kanal 2 Bit 1: Parametrierungsfehler Kanal 2 Bit 2 ... 7: reserviert	00h
11	Bit 0: Drahtbruch bei Stromausgabe bzw. Kurzschluss bei Spannungsausgabe Kanal 3 Bit 1: Parametrierungsfehler Kanal 3 Bit 2 ... 7: reserviert	00h



## Technische Daten

Elektrische Daten	VIPA 234-1BD50							
Anzahl der Ein/Ausgänge	2/2							
Spannungsversorgung	DC5V über Rückwandbus DC24V (20,4 ... 28,8V)							
Stromaufnahme	Rückwandbus: 100mA DC 24V extern: 110mA							
Kurzschlussstrom	30mA							
Bereiche	±10V, 1 ... 5V, 0 ... 10V, ±20mA, 0 ... 20mA, 4 ... 20mA							
Analogwertbildung Eingänge	Wandlungszeit/Auflösung (pro Kanal)							
parametrierte Geschwindigkeit (Hz)	3,7	7,5	15	30	60	123	168	202
Grundwandlungszeit (ms)	268	135	69	35,5	19	10	8	6,75
Zusätzliche Wandlungszeit (wird einmal pro Zyklus abgearbeitet) (ms)	10	10	10	10	10	10	10	10
Zusätzliche Wandlungszeit für Drahtbruchüberwachung (ms)	6,5	6,5	6,5	6,5	6,5	6,5	6,5	6,5
Auflösung in Bit	16	16	16	16	15	14	12	10
Analogwertbildung Ausgänge	Auflösung (inkl. Übersteuerungsbit) ±10V, ±20mA 4 ... 20mA, 1 ... 5V 0 ... 10V, 0 ... 20mA Zykluszeit 11Bit + Vorzeichen 10Bit 11Bit 2,5ms							
Auflösung (inkl. Übersteuerungsbit)								
±10V, ±20mA								
4 ... 20mA, 1 ... 5V								
0 ... 10V, 0 ... 20mA	11Bit							
Zykluszeit	2,5ms							
Einschwingzeit	0,05ms 0,5ms 0,1ms							
- ohmsche Last								
- kapazitive Last								
- induktive Last	0,1ms							
Fehlergrenzen	Messbereich				Toleranz			
- Spannungsein- /-ausgang	±10V				±0,2%			
	0 ... 10V				±0,4%			
	1 ... 5V				±0,6%			
- Stromein- /-ausgang	±20mA				±0,3%			
	0 ... 20mA				±0,6%			
	4 ... 20mA				±0,8%			

Fortsetzung ...

... Fortsetzung Technische Daten

Elektrische Daten	
Daten zur Auswahl des Gebers - Spannungseingang - Stromeingang	100 k $\Omega$ 50 $\Omega$
Daten zur Auswahl des Aktors - Spannungsausgänge  - Stromausgänge	Bürdenwiderstand ohmsche Last - min. 1 k $\Omega$ kapazitive Last - max. 1 $\mu$ F ohmsche Last - max. 500 $\Omega$ kapazitive Last - max. 10 mH
Diagnosealarm Potenzialtrennung  Statusanzeige	parametrierbar 500Veff (Feldspannung-Rückwandbus) über LEDs auf der Frontseite
Programmierdaten	
Eingabedaten Ausgabedaten Parameterdaten Diagnosedaten	4Byte (1 Wort pro Kanal) 4Byte (1 Wort pro Kanal) 12Byte 12Byte
Maße und Gewicht	
Abmessungen (BxHxT) Gewicht	25,4x76x76mm 100g

## Teil 19 Systemerweiterungen

### Überblick

In diesem Kapitel werden die bei VIPA erhältlichen Zusatzkomponenten und Zubehörteile für das System 200V beschrieben.

Nach einer Gesamtübersicht folgt die Beschreibung der Buserweiterung, die das Aufteilen einer System 200V Zeile auf bis zu 4 Zeilen ermöglicht.

Der 4fach Fast Ethernet Mini-Switch ergänzt das System 200V im Bereich der Netzwerktechnologie.

Mit den Klemmen-Modulen endet das Kapitel. Hiermit können Sie neben dem Anschluss der Signalleitungen auch den Anschluss der Versorgungsspannung über Ihr System 200V führen.

Nachfolgend sind beschrieben:

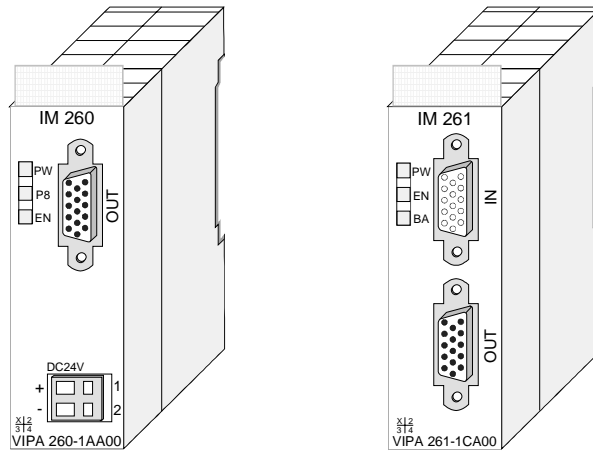
- Systemübersicht der Zusatzkomponenten
- Buserweiterung mit IM 260 und IM 261
- 4fach Mini-Switch CM 240
- Klemmen-Modul CM 201

### Inhalt

Thema	Seite
<b>Teil 19 Systemerweiterungen</b> .....	<b>19-1</b>
Systemübersicht .....	19-2
Buserweiterung IM 260, IM 261 .....	19-4
4fach Mini-Switch CM 240 .....	19-7
Klemmen-Modul CM 201 .....	19-10

# Systemübersicht

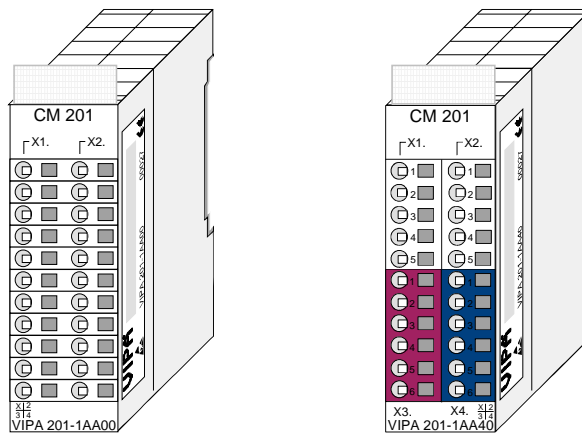
## Buserweiterung



## Bestelldaten Buserweiterung

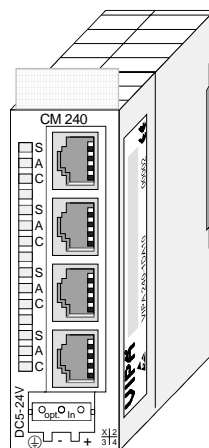
Typ	Bestellnummer	Beschreibung
IM 260	VIPA 260-1AA00	Basisanschaltung 1. Zeile
IM 261	VIPA 261-1CA00	Zeilenanschaltung 2. ... 4. Zeile
Kabel 0,5m	VIPA 260-1XY05	Verbindungskabel mit 0,5m Länge
Kabel 1m	VIPA 260-1XY10	Verbindungskabel mit 1m Länge
Kabel 1,5m	VIPA 260-1XY15	Verbindungskabel mit 1,5m Länge
Kabel 2m	VIPA 260-1XY20	Verbindungskabel mit 2m Länge
Kabel 2,5m	VIPA 260-1XY25	Verbindungskabel mit 2,5m Länge

## Klemmen-Module



## Bestelldaten Klemmen-Module

Typ	Bestellnummer	Beschreibung
CM 201	VIPA 201-1AA00	Doppelklemmen grau/grau
CM 201	VIPA 201-1AA10	Doppelklemmen grün-gelb/grau
CM 201	VIPA 201-1AA20	Doppelklemmen rot/blau
CM 201	VIPA 201-1AA40	Vierfachklemmen grau/rot/blau

**4fach Mini-Switch****Bestelldaten  
4fach Mini-Switch**

Typ	Bestellnummer	Beschreibung
CM 240	VIPA 240-1DA10	4fach Mini-Switch
	VIPA 970-0CM00	optionaler Frontstecker bei externer Spannungsversorgung DC 5-24V

**Hinweis Bestell-Nr.-Umstellung!**

Aufgrund einer Best-Nr.-Umstellung ergeben sich für den 4fach-Mini-Switch folgende Änderungen:

Alte Bestell-Nr.	Neue Bestell-Nr.
243-1DA10	240-1DA10

## Buserweiterung IM 260, IM 261

Das System, bestehend aus IM 260, IM 261 und Verbindungskabel, stellt eine Zeilenerweiterung dar, die Ihnen das Aufteilen Ihres System 200V auf bis zu 4 Zeilen ermöglicht.

Das System darf nur in einem zentralen System 200V, mit einem PC 288 oder einer CPU als Kopfstation, eingesetzt werden!

Zur Buserweiterung ist immer die Basisanschlusung IM 260 erforderlich. An die Basisanschlusung können Sie über entsprechende Verbindungskabel bis zu 3 weitere System 200V Zeilen über die Zeilenanschlusung IM 261 ankoppeln.



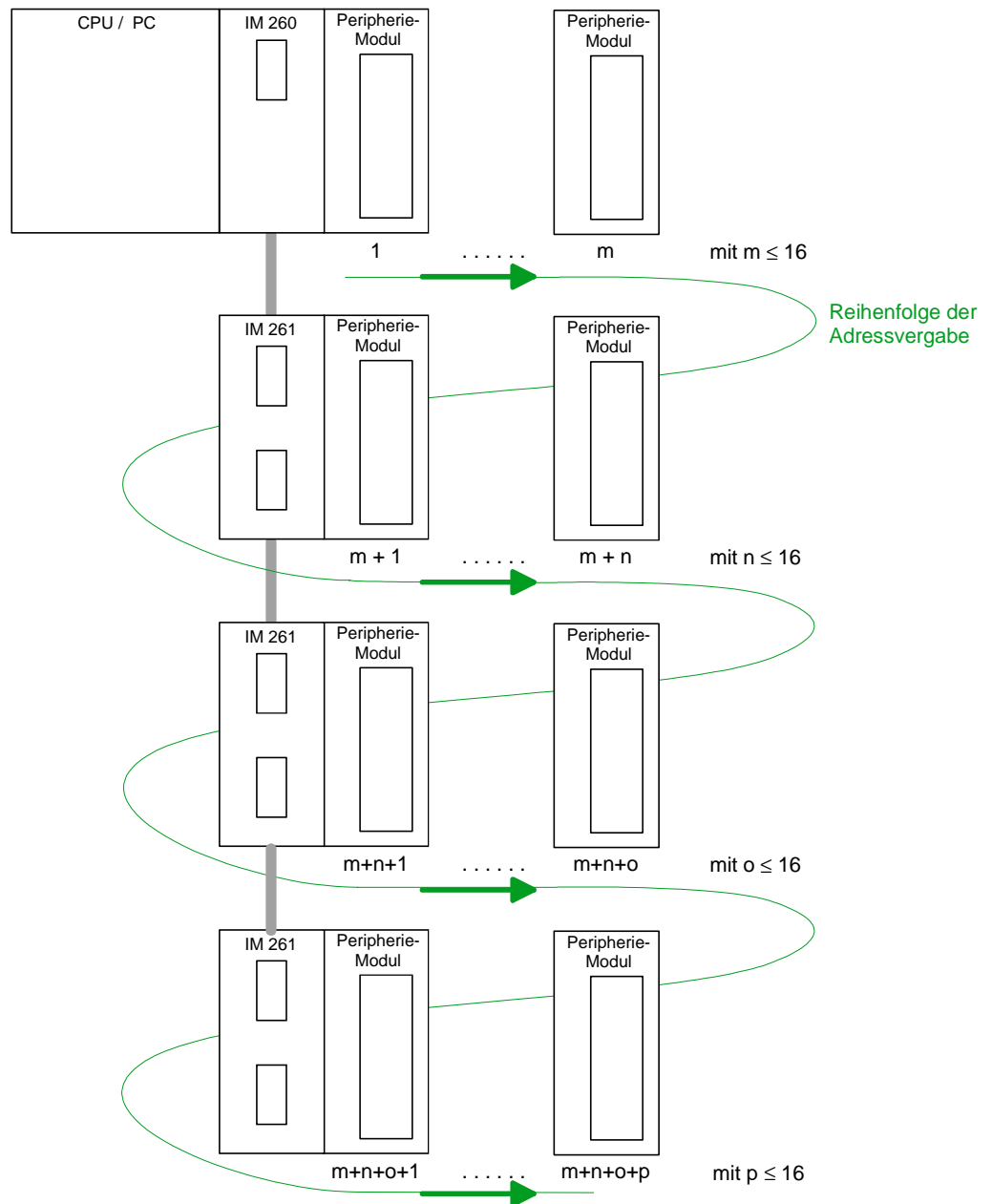
### Bitte beachten!

Für den Einsatz der Buserweiterung gibt es gewisse Regeln, die zu beachten sind:

- Die Buserweiterung darf nur bei Verwendung des PC 288 (VIPA 288-2BL10) oder einer CPU (auch Kombi-CPU's) verwendet werden. In dezentralen Systemen wie z.B. hinter einem Profibus-DP-Slave darf das System nicht eingesetzt werden!
- Es dürfen maximal 4 Zeilen aufgebaut werden.
- Jede Zeile darf maximal 16 Peripheriemodule beinhalten.
- Die Summe von max. 32 Peripheriemodulen insgesamt darf nicht überschritten werden.
- In kritischem Umfeld sollte die Gesamtkabellänge von max. 2m nicht überschritten werden.
- In einer Zeile dürfen über den Rückwandbus max. 1,5A aufgenommen werden, in der Summe 4A.
- Neben der Basisanschlusung IM 260 muss mindestens ein Peripheriemodul stecken!

**Aufbau**

In der nachfolgenden Abbildung ist der Aufbau einer Buserweiterung unter Beachtung der Aufbauregeln aufgeführt:



Es gilt:  $m + n + o + p \leq 32$

**Hinweis!**

Die Buserweiterung darf nur bei Verwendung des PC 288 (VIPA 288-2BL10) oder einer CPU (auch Kombi-CPU) verwendet werden!

Ab folgenden Firmwareständen wird die Buserweiterung unterstützt:

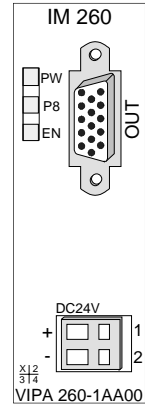
CPU befehlskompatibel zu STEP<sup>®</sup> 5 von Siemens: ab Version 2.07

CPU befehlskompatibel zu STEP<sup>®</sup> 7 von Siemens: ab Version 1.0

CPU für IEC 61131-3: ab Version 1.0

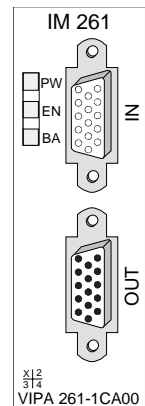
**Statusanzeige  
Basisanschlutung  
IM 260**

LED	Farbe	Beschreibung
PW	gelb	Versorgungsspannung liegt an
P8	gelb	Versorgungsspannung für nachfolgende Zeilen aktiv
EN	gelb	Rückwandbus-Kommunikation aktiv



**Statusanzeige  
Zeilenanschlutung  
IM 261**

LED	Farbe	Beschreibung
PW	gelb	Versorgungsspannung über IM 260 liegt an
EN	gelb	Rückwandbus-Kommunikation aktiv
BA	rot	Befehlsausgabesperre (BASP) aktiv



**Bestelldaten  
Kabel**

Typ	Bestellnummer	Beschreibung
Kabel 0,5m	VIPA 260-1XY05	Verbindungskabel mit 0,5m Länge
Kabel 1m	VIPA 260-1XY10	Verbindungskabel mit 1m Länge
Kabel 1,5m	VIPA 260-1XY15	Verbindungskabel mit 1,5m Länge
Kabel 2m	VIPA 260-1XY20	Verbindungskabel mit 2m Länge
Kabel 2,5m	VIPA 260-1XY25	Verbindungskabel mit 2,5m Länge

**Technische Daten**

Elektrische Daten	VIPA 260-1AA00	VIPA 261-1CA00
Spannungsversorgung	über Front DC24V	-
Stromaufnahme	1,9A	-
Stromaufnahme Rückwandbus	30mA	-
Stromversorgung Rückwandbus an IM 261	-	max. 1,5A pro Zeile (Summe max. 4A)
max. Kabellänge zw. 1. und letzter Zeile	2,5m	
Maße und Gewicht		
Abmessungen (BxHxT) in mm	25,4x76x76	25,4x76x76
Gewicht	80g	50g



## 4fach Mini-Switch CM 240

### Bestelldaten

4fach Mini-Switch CM 240

VIPA 240-1DA10

**Achtung: der 4-fach Mini-Switch hatte ehemals die Best.-Nr. 243-1DA10!**

### Übersicht

Der 4fach Mini-Switch ergänzt das System 200V im Bereich der Netzwerktechnologie.

Auto-Negotiation, Speed-Auto-Sensing und das Auto MDI/MDIX-Crossover für jeden Port machen das Modul "plug & play"-fähig.

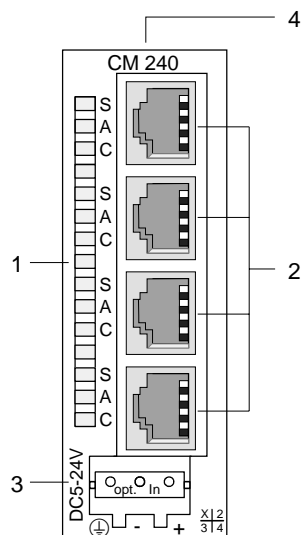
Das Modul wird über den Rückwandbus mit der erforderlichen Betriebsspannung versorgt. Alternativ können Sie das Modul auch über die Front versorgen.

Die Statusanzeige der 4 Ports erfolgt über LEDs auf der Frontseite.

### Eigenschaften

- 4 Ports für 10 bzw. 100MBit/s,
- "plug and play"-fähig durch Auto-MDI/MDIX-crossover für 100BASE-TX und 10BASE-T,
- Auto-Negotiation und Speed-Auto-Sensing
- je Port automatische Umschaltung zwischen 10 und 100MBit/s bzw. Halb- und Vollduplexbetrieb
- LED-Anzeige für Activity, Speed und Collision
- Unterstützt IEEE 802.3, IEEE 802.3u und IEEE 802.3x
- Extra hohe Performance bis zu 150m bei UTP (ungeschirmtes Twisted Pair Kabel)
- Back-pressure-based flow control bei Halbduplexbetrieb
- Pause-frame-based flow control bei Vollduplexbetrieb
- Store-and-forward switching mode
- Shared memory based switch

### Frontansicht CM 240



- [1] LED Statusanzeige
- [2] Twisted Pair Ports für Ethernet
- [3] Spannungsversorgung extern
- [4] DIL-Schalter für Security-Funktion

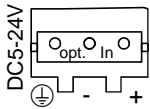
**Komponenten**

**LEDs**

Für jede Twisted Pair Buchse befinden sich je 3 LEDs auf der Frontseite. Die LEDs haben folgende Funktionen:

Name	Farbe	Funktion	Beschreibung
S	grün	Speed	an: 100MBit, aus: 10MBit
A	gelb	Activity	an: physikalisch verbunden, aus: keine physikalische Verbindung blinkt: zeigt Busaktivität an
C	gelb	Collision	an: Vollduplexbetrieb aktiv, aus: Halbduplexbetrieb aktiv blinkt: Collision detected

**Spannungsversorgung**



Die Spannungsversorgung erfolgt über den Rückwandbus des System 200V. Sie können den Mini-Switch auch als Stand-Alone-Gerät betreiben. Hierzu ist dieser extern mit DC 5...24V zu versorgen.

Der Stecker zum Anschluss einer externen Spannungsversorgung befindet sich unter einer Abdeckung, die herauszubringen ist.

Zum Anschluss einer externen Spannungsversorgung erhalten Sie von VIPA unter der Best.-Nr. VIPA 970-0CM00 eine Anschluss-Buchse.



**Achtung!**

Die Spannungsversorgung hat entweder über den Rückwandbus oder von extern zu erfolgen.

**Eine gleichzeitige Einspeisung ist unbedingt zu vermeiden!**

**Twisted Pair Ports**

Die Twisted Pair Buchsen dienen zum Aufbau eines Twisted Pair Netzwerks in Sterntopologie. Hierbei können Sie bis zu 4 Ethernet-Komponenten anschließen wobei 1 Anschluss als Uplink-Port an das weiterführende Netzwerk anzukoppeln ist.  
 Der Uplink-Port wird automatisch erkannt, sofern Sie nicht die Security-Funktion via DIL-Schalter aktiviert haben.

**DIL-Schalter für Security-Funktion:**



Die Aktivierung der Security-Funktion verhindert die Querkommunikation der Teilnehmer der unteren drei Twisted Pair Buchsen, Voraussetzung ist in diesem Fall der Uplink über den obersten Twisted Pair Port.  
 Zur Aktivierung der Security-Funktion befindet an der Oberseite des Moduls auf der Platine ein DIL-Schalter, der folgende Belegung hat:  
 Schalter 1 an: Security-Funktion deaktiviert (Default)  
 aus: Security-Funktion aktiv  
 Schalter 2 zurzeit keine Funktion

**Technische Daten**

Elektrische Daten	VIPA 240-1DA10
Anzahl der Ports	4
Stromaufnahme über Rückwandbus	460mA
Spannungsversorgung (intern)	DC 5V, über Rückwandbus
Spannungsversorgung (extern)	optional Anschluss-Buchse VIPA 970-0CM00 DC 5...24V
Statusanzeige	über LEDs auf der Frontseite
Schnittstelle	
	RJ45 Twisted Pair, UTP, S/FTP
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x76
Gewicht	50g

# Klemmen-Modul CM 201

## 2 x 11 Pole

Das Klemmen-Modul hat die Best.-Nr.: VIPA 201-1AAxx.

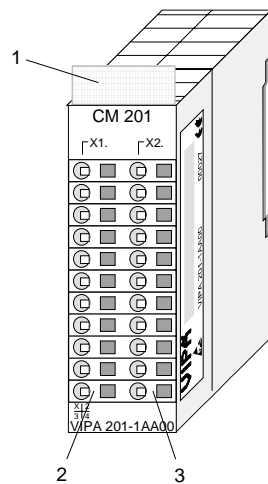
Das Modul ist ein Ergänzungs-Modul für 2- oder 3-Draht Installation. Es hat keine Verbindung zum Systembus.

## Eigenschaften

- 2 getrennte Reihen mit je 11 elektrisch verbundenen Klemmen.
- Keine Verbindung zum Systembus.
- Maximaler Klemmenstrom 10A.

## Aufbau und Prinzipschaltbild

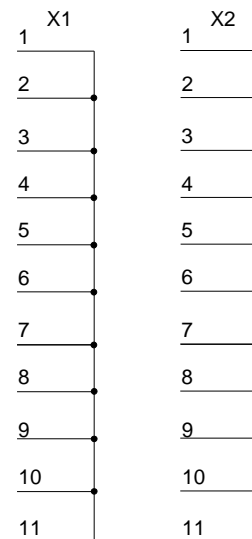
Aufbau



Beschreibung

- [1] Beschriftungsstreifen
- [2] 1. Klemmleiste
- [3] 2. Klemmleiste

Prinzipschaltbild



## Technische Daten

Elektrische Daten	VIPA 201-1AA00	VIPA 201-1AA10	VIPA 201-1AA20
Anzahl Reihen	2	2	2
Anzahl Klemmen je Reihe	11	11	11
Maximaler Klemmenstrom	10A	10A	10A
Klemmenfarbe	grau/grau	grün-gelb/grau	rot/blau
Maße und Gewicht			
Abmessungen (BxHxT) in mm	25,4 x 76 x 76	25,4 x 76 x 76	25,4 x 76 x 76
Gewicht	50g	50g	50g

**2 x 5 Pole**  
**2 x 6 Pole**

Das Klemmen-Modul hat die Best.-Nr.: VIPA 201-1AA40.

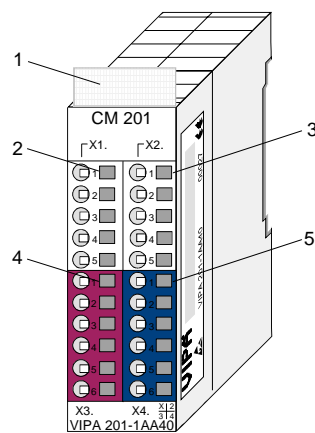
Das Modul ist ein Erganzungs-Modul fur 2- oder 3-Draht Installation. Es hat keine Verbindung zum Systembus.

**Eigenschaften**

- 4 getrennte Reihen mit 2 x 5 und 2 x 6 elektrisch verbundenen Klemmen.
- Keine Verbindung zum Systembus.
- Maximaler Klemmenstrom 10A.

**Aufbau und  
Prinzipschaltbild**

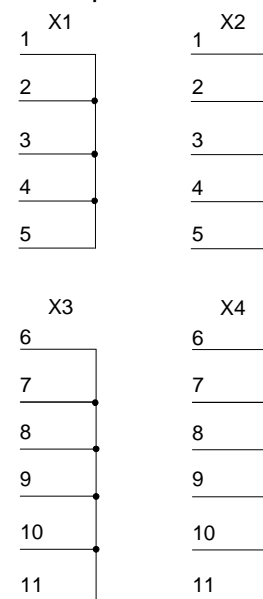
**Aufbau**



**Beschreibung**

- [1] Beschriftungsstreifen  
[2] 1. Klemmleiste  
[3] 2. Klemmleiste  
[4] 3. Klemmleiste  
[5] 4. Klemmleiste

**Prinzipschaltbild**



**Technische Daten**

Elektrische Daten	VIPA 201-1AA40
Anzahl Reihen	2
Anzahl Klemmen je Reihe	5 + 6
Maximaler Klemmenstrom	10A
Klemmenfarbe	grau/rot/blau
Mae und Gewicht	
Abmessungen (BxHxT) in mm	25,4 x 76 x 76
Gewicht	50g



## Teil 20 Montage und Aufbaurichtlinien

### Überblick

In diesem Kapitel finden Sie alle Informationen, die für den Aufbau und die Verdrahtung einer Steuerung aus den Komponenten des Systems 200V erforderlich sind.

Nachfolgend sind beschrieben:

- Allgemeine Übersicht der Komponenten
- Schritte der Montage und Verdrahtung
- Tabelle zur Montage unter Berücksichtigung der Stromaufnahme
- EMV-Richtlinien zum Aufbau eines System 200V

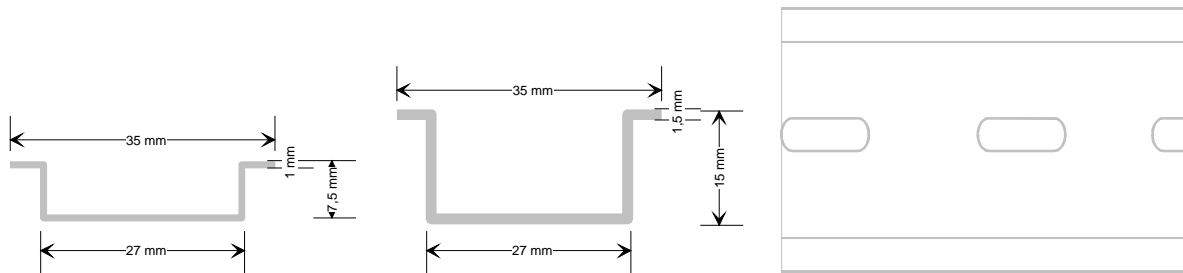
### Inhalt

Thema	Seite
<b>Teil 20 Montage und Aufbaurichtlinien .....</b>	<b>20-1</b>
Übersicht .....	20-2
Montage.....	20-4
Verdrahtung.....	20-9
Einbaumaße .....	20-11
Maschinelle Beschriftung .....	20-12
Aufbaurichtlinien .....	20-13

# Übersicht

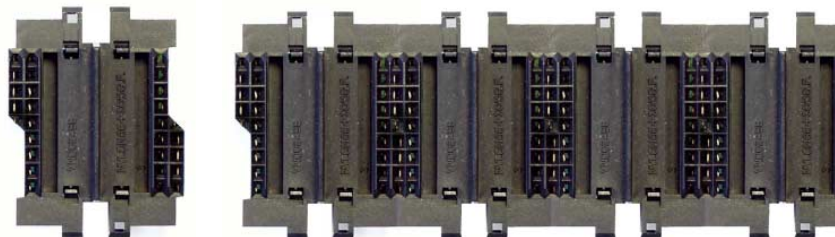
**Allgemein** Die einzelnen Module werden direkt auf eine Tragschiene montiert und über Rückwandbusverbinder, die vorher in die Profilschiene eingelegt werden, gekoppelt.

**Tragschienen** Für die Montage können Sie folgende 35mm-Normprofilschiene verwenden:



**Busverbinder** Für die Kommunikation der Module untereinander wird beim System 200V ein Rückwandbusverbinder 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 ist isoliert und ist in die Tragschiene einzulegen, bis dieser sicher einrastet, so dass die Bus-Anschlüsse aus der Tragschiene herauschauen.

**Bestelldaten**

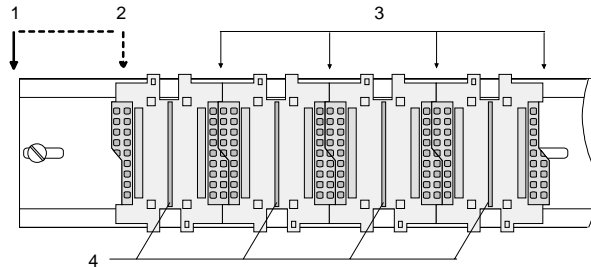
Typ	Bestellnummer	Beschreibung
Busverbinder	VIPA 290-0AA10	Busverbinder 1fach
Busverbinder	VIPA 290-0AA20	Busverbinder 2fach
Busverbinder	VIPA 290-0AA40	Busverbinder 4fach
Busverbinder	VIPA 290-0AA80	Busverbinder 8fach
Tragschiene	VIPA 290-0AF30	35x15mm, 530mm lang 1,5mm dick, gelocht



**Montage auf Tragschiene**

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

Die einzelnen Modulsteckplätze sind durch Führungsleisten abgegrenzt.

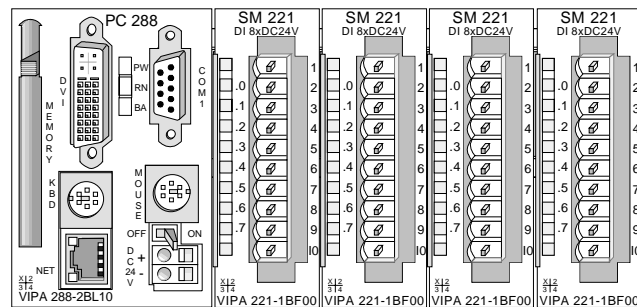
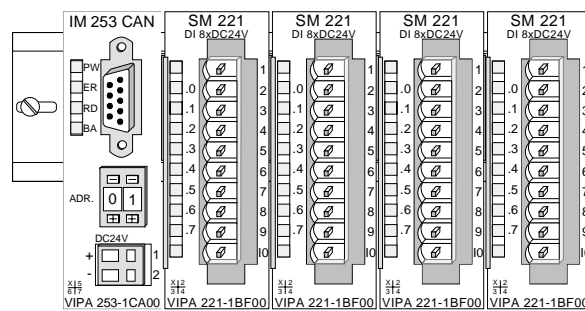


- [1] Kopfmodul wie PC, CPU, Bus-Koppler, wenn doppelt breit
- [2] Kopfmodul (einfach breit)
- [3] Peripheriemodule
- [4] Führungsleisten

**Hinweis**

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

Weiter unten finden Sie eine Übersicht der Stromaufnahmen.

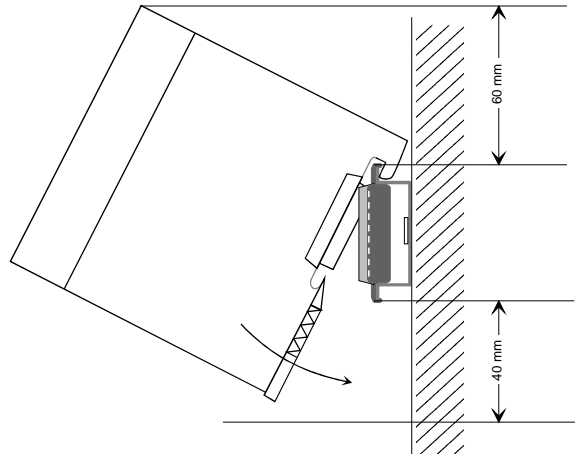


## Montage

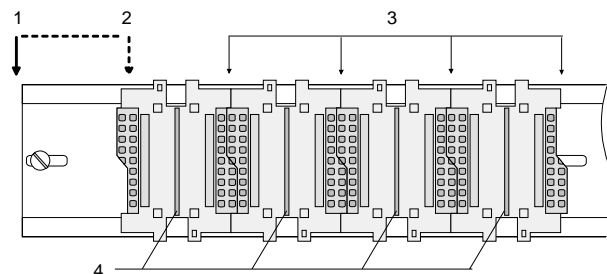


### Bitte bei der Montage beachten!

- Schalten Sie die Stromversorgung aus, bevor Sie Module stecken bzw. abziehen!
- Bitte beachten Sie, dass sie oberhalb der Busschiene einen Modul-Montageabstand von mindestens 60mm und unterhalb 40mm einhalten.



- Eine Zeile ist immer von links nach rechts aufzubauen und hat immer mit einem Kopfmodul (PC, CPU, Buskoppler) zu beginnen.



- [1] Kopfmodul wie PC, CPU, Buskoppler, wenn doppelt breit
- [2] Kopfmodul (einfach breit)
- [3] Peripheriemodule
- [4] Führungsleisten

- Module müssen immer direkt nebeneinander gesteckt sein. Lücken zwischen den Modulen 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.

**Montage unter Berücksichtigung der Stromaufnahme**

- Verwenden Sie möglichst lange Busverbinder
- Ordnen Sie Module mit hohem Stromverbrauch rechts neben Ihrem Kopfmodul an. Die nachfolgende Tabelle soll Ihnen hierbei behilflich sein:

*Stromaufnahme der Peripherie-Module*

Bestell-Nr.	Produktbezeichnung	Stromaufnahme
<b>Digitale Eingabe-Module SM 221</b>		
VIPA 221-1BF00	DI 8xDC24V	20mA
VIPA 221-1BF10	DI 8xDC24V (0,2ms)	20mA
VIPA 221-1BF20	DI 8xDC24V Alarm	140mA
VIPA 221-1BF50	DI 8xDC24V NPN	20mA
VIPA 221-1FD00	DI 4xAC/DC 90...230V	80mA
VIPA 221-1FF20	DI 8xAC/DC 60...230V	80mA
VIPA 221-1FF30	DI 8xAC/DC 24...60V	80mA
VIPA 221-1FF40	DI 8xAC 230V, 20mA	80mA
VIPA 221-1FF50	DI 8xAC/DC 180...265V	30mA
VIPA 221-1BH00	DI 16xDC 24V, UB4x	20mA
VIPA 221-1BH10	DI 16xDC 24V	30mA
VIPA 221-1BH20	DI 16xDC 24V, 1 Zähler, 100kHz	100mA
VIPA 221-1BH50	DI 16xDC 24V, NPN	20mA
VIPA 221-2BL10	DI 32xDC 24V	50mA
<b>Digitale Ausgabe-Module SM 222</b>		
VIPA 222-1BF00	DO 8xDC 24V, 1A	50mA
VIPA 222-1BF10	DO 8xDC 24V, 2A	50mA
VIPA 222-1BF20	DO 8xDC 24V, 2A, pot. getr.	50mA
VIPA 222-1BH00	DO 16xDC 24V, 0,5A, UB4x	100mA
VIPA 222-1BH10	DO 16xDC 24V, 1A	80mA
VIPA 222-1BH20	DO 16xDC 24V, 2A	100mA
VIPA 222-1BH50	DO 16xDC 24V, 0,5A, NPN	100mA
VIPA 222-2BL10	DO 32xDC 24V, 1A	140mA
VIPA 222-1HF00	DO 8xRelais	270mA
VIPA 222-1HD10	DO 4xRelais, pot. getrennt	150mA
VIPA 222-1HD20	DO 4xRelais, pot. getrennt, bistabil	40mA
VIPA 222-1FF00	DO 8xRelais, Solid State	140mA
VIPA 222-1FD10	DO 4xRelais, Solid State	100mA

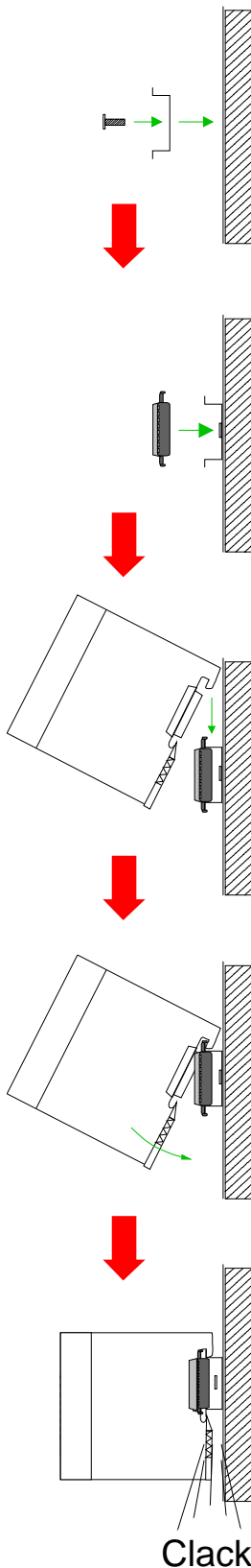
*Fortsetzung ...*

... Fortsetzung

Bestell-Nr.	Produktbezeichnung	Strom- aufnahme
<b>Digitale Ein-/Ausgabemodule SM 223</b>		
VIPA 223-1BF00	DIO 8xDC 24V, 1A	65mA
VIPA 223-2BL10	DI 16xDC24V, DO 16xDC24V, 1A	100mA
<b>Analoge Eingabe-Module SM 231</b>		
VIPA 231-1BD52	AI 4x16Bit, Multiinput	240mA
VIPA 231-1BD60	AI 4x12Bit, 4...20mA	280mA
VIPA 231-1BD70	AI 4x12Bit, $\pm 10V$	300mA
VIPA 231-1BF00	AI 8x16Bit, (2L) 4x16Bit (4L)	280mA
VIPA 231-1FD00	AI 4x16Bit f, U/I	300mA
<b>Analoge Ausgabe-Module SM 232</b>		
VIPA 232-1BD50	AO 4x12Bit	30mA
VIPA 232-1BD60	AO 4x12Bit f	50mA
<b>Analoge Ein-/Ausgabemodule SM 234</b>		
VIPA 234-1BD50	AI/AO 2x12Bit, Multiin-/output	110mA
<b>Sonstige Module</b>		
VIPA 208-1CA01	IM 208CAN - CANopen Master	250mA
VIPA 208-1DP01	IM 208DP für CPU 21x	450mA
VIPA 208-2DP10	IM 208DPO	500mA
VIPA 240-1BA00	CP 240, RS232C	200mA
VIPA 240-1BA10	CP 240, RS232C, Modbus	200mA
VIPA 240-1CA00	CP 240, RS422/485	200mA
VIPA 240-1CA10	CP 240, RS422/485, Modbus	200mA
VIPA 240-1DA10	CM 240, 4port Mini-Switch	460mA
VIPA 250-1BA00	Counter 2x32Bit FM 250	80mA
VIPA 250-1BS00	SSI-Modul FM250S	200mA
VIPA 253-1BA00	MotionControl Stepper	200mA
VIPA 254-1BA00	MotionControl Servo	295mA

## Montage Vorgehensweise

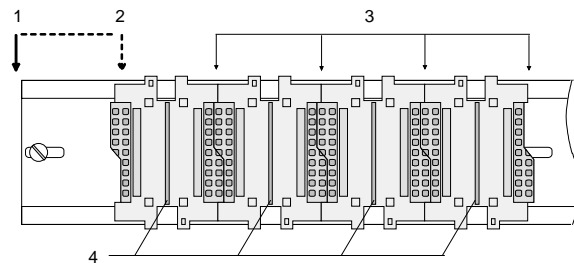
Die nachfolgende Abfolge stellt die Montageschritte in der Seitenansicht dar.



- Montieren Sie die Tragschiene! Bitte beachten Sie, dass sie oberhalb der Schiene einen Modul-Montageabstand von mindestens 60mm und unterhalb 40mm einhalten.

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

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



- [1] Kopfmodul wie PC, CPU, Buskoppler, wenn doppelt breit
- [2] Kopfmodul (einfach breit)
- [3] Peripheriemodule
- [4] Führungsleisten

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

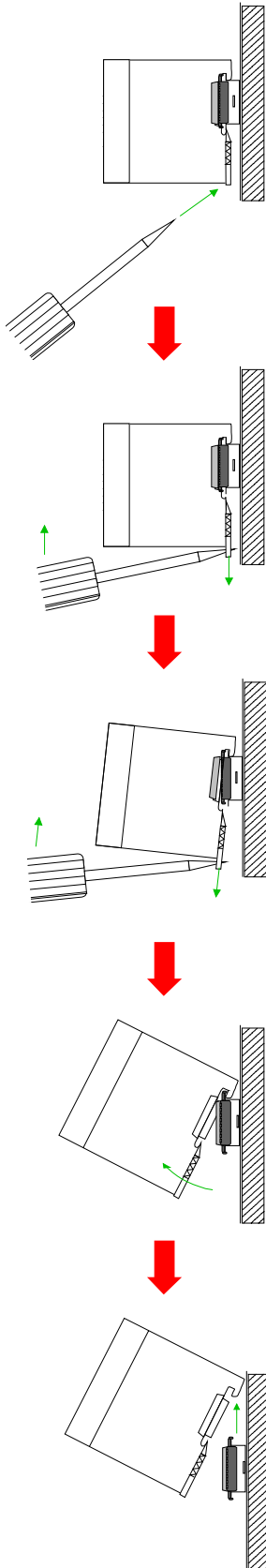


### Achtung!

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

**Demontage  
Vorgehensweise**

Die nachfolgende Abfolge stellt die Schritte zur Demontage in der Seitenansicht dar.



- Zur Demontage befindet sich am Gehäuseunterteil eine gefederter Demontageschlitz.
- Steckern Sie wie gezeigt einen Schraubendreher in den Demontageschlitz

- Durch Druck des Schraubendrehers nach oben wird das Modul entriegelt.

- 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.

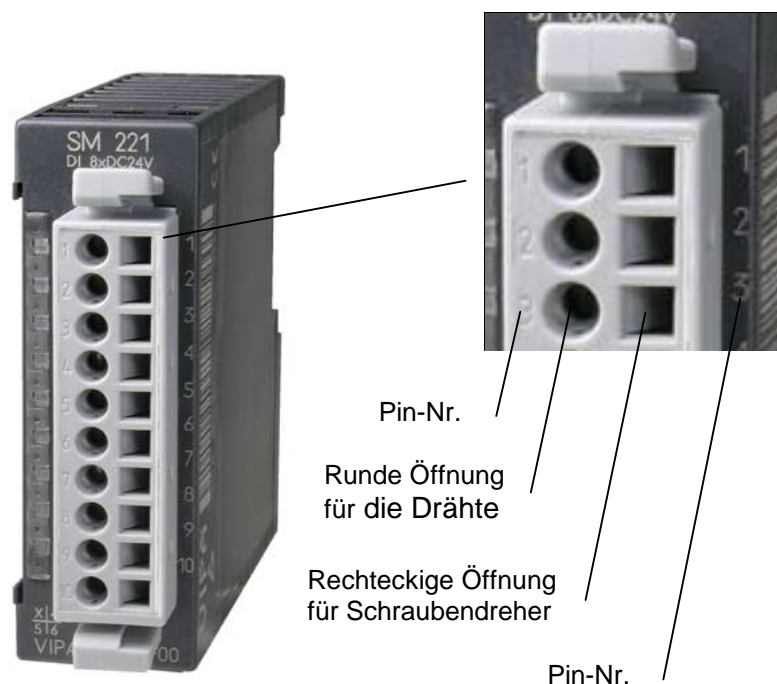
Bei der Verdrahtung werden Steckverbinder mit Federklemmtechnik eingesetzt.

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  $0,08\text{mm}^2$  bis  $2,5\text{mm}^2$  (bis  $1,5\text{mm}^2$  bei 18poligen) anschließen.

Folgende Abbildung zeigt ein Modul mit einem 10poligen Steckverbinder.

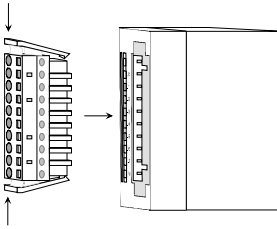


### Hinweis!

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

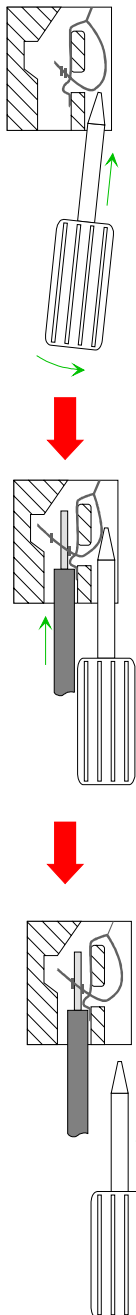
Drücken Sie den Schraubendreher nur in die rechteckigen Öffnungen des Steckverbinders!

## 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.



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



## Einbaumaße

### Übersicht

Hier finden Sie alle wichtigen Maße des System 200V.

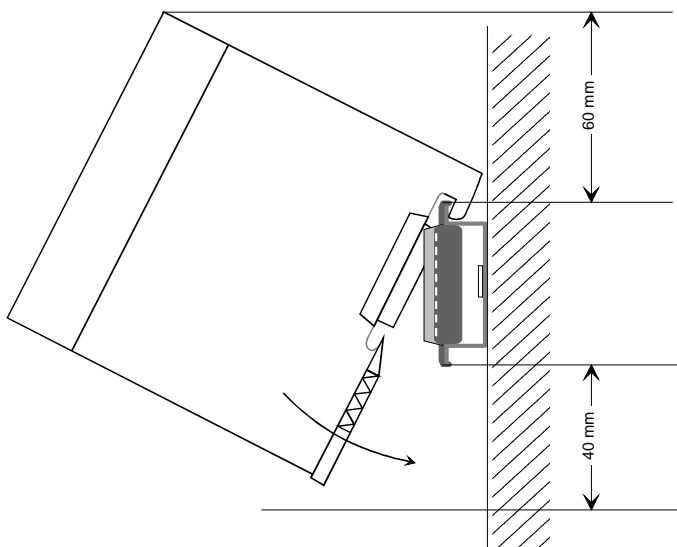
### Maße

#### Grundgehäuse

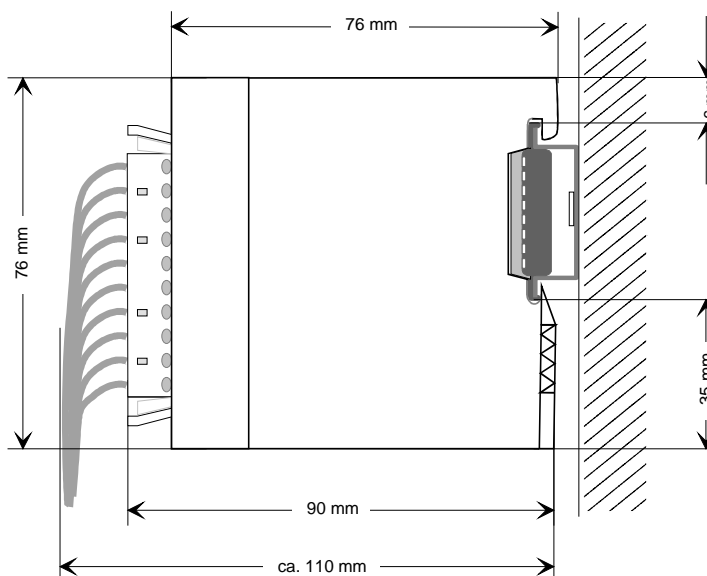
1fach breit (HxBxT) in mm: 76 x 25,4 x 76

2fach breit (HxBxT) in mm: 76 x 50,8 x 76

### Montagemaße



### Maße montiert und verdrahtet



## Maschinelle Beschriftung

**Allgemein** Die System 200V Peripherie-Module besitzen Beschriftungsstreifen, die maschinell beschriftet werden können.

**Beschriftung unter WinNCS** Zur Beschriftung dieser Etiketten können Sie die Etikettendruck-Komponente aus WinNCS verwenden. WinNCS ist ein Projektierwerkzeug von VIPA, dessen Etikettendruck-Komponente speziell auf die Beschriftungsstreifen des System 200V abgestimmt ist.

**Beschriftung unter WinLP** Für den Etiketten-Druck eines S7-Projekts von Siemens ist bei VIPA die Software WinLP erhältlich, die aus einer S7-cfg-Datei von Siemens Beschriftungsstreifen für das System 200V generiert.

### Bestelldaten

Typ	Bestellnummer	Beschreibung
WinNCS	VIPA SW-WinNCS	Parametrier- und Beschriftungssoftware für System 200V unter Win9x/NT einschl. WinLP
Demo-Software	VIPA SW-Tool Demo	Demoversionen aller VIPA-Tools mit voller Beschriftungsfunktion unter WinNCS
Beschriftungskarte	VIPA 292-1XY00	10 Beschriftungskarten mit Deckfolie
Beschriftungsbogen	VIPA 292-1XY10	10 Beschriftungsbögen perforiert à 8 Beschriftungsstreifen

## Aufbaurichtlinien

- Allgemeines** Die Aufbaurichtlinien enthalten Informationen über den störsicheren 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.
  - Beschalten Sie alle Induktivitäten mit Löschgliedern, die nicht von System 200V Modulen angesteuert werden.
  - Benutzen Sie zur Beleuchtung von Schränken Glühlampen und vermeiden Sie Leuchtstofflampen.
- 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



# Anhang

## A Index

- 2  
 20mA-Schnittstelle ..... 9-11; 9-15  
 2-Draht-Anschluss ..... 16-4
- 3  
 3964(R) ..... 9-5; 9-24  
   mit RK512 ..... 9-6; 9-24
- 4  
 4-Draht-Anschluss ..... 16-4
- A**  
 Abisolierlängen ..... 2-56  
 Aderquerschnitt ..... 1-7  
 Adresseinsteller  
   CAN-Bus ..... 4-17  
   DeviceNet ..... 5-6  
   Profibus-DP ..... 2-35  
   SERCOS ..... 6-6  
 Aktoren anschließen ..... 17-3  
 Alarmeingang ..... 13-8  
 Analoge Ausgabe ..... 17-1  
 Analoge Ein-/Ausgabe ..... 18-1  
 Analoge Eingabe ..... 16-1  
 Analogsignalleitungen ..... 16-4  
 Anwenderquittung ..... 9-29  
 Anwendungsschicht ..... 1-10  
 ASCII ..... 9-4; 9-22  
 Aufbaurichtlinien ..... 20-13  
 Auto Reload ..... 10-13
- B**  
 Basic CMOS Configuration ..... 8-15  
 BASP-Signal ..... 10-7  
 Baudrate  
   CAN-Bus ..... 4-17  
   CP ..... 9-26  
   DeviceNet ..... 5-9  
   SERCOS ..... 6-8  
   SSI-Interface ..... 10-6  
 BCC-Byte ..... 9-7  
 Betriebszustände  
   Interbus ..... 3-4  
   Profibus-Master ..... 2-12  
 Bitübertragungsschicht ..... 1-9  
 Bootlaufwerk ..... 8-18  
 Buserweiterung ..... 19-4  
 Busverbinder ..... 20-2  
 BWZ ..... 9-29
- C**  
 CAN-Bus ..... 4-1  
   Adresseinsteller ..... 4-17  
   Baudrate ..... 4-17  
   Buszugriff ..... 4-4  
   CANopen ..... 4-3  
   Fehlermeldung ..... 4-10  
   Fehlermeldungen ..... 4-6  
   Funktionscodes ..... 4-19  
   Kommunikationsarten ..... 4-22  
   Objektverzeichnis ..... 4-26  
   Schnelleinstieg ..... 4-13  
   Slave ..... 4-5  
   Slave mit DO 24xDC 24V ..... 4-9  
   Telegrammaufbau ..... 4-18  
   Verkabelung ..... 4-7; 4-12  
 CAN-Identifizierung ..... 4-13; 4-21  
 CLK ..... 10-13  
 COB-ID ..... 4-21  
 Com1 Mode ..... 8-19  
 CompactFlash ..... 8-5; 8-9; 8-15  
 Compare Load ..... 10-13  
 Controlbyte ..... 10-11  
 CP ..... 9-1  
   Diagnose ..... 9-13  
   Hantierungsbausteine ..... 9-45  
     für CPU 21x ..... 9-61  
     für CPU 24x ..... 9-46  
   Kommunikation ..... 9-45  
   mit 20mA/RS232C ..... 9-11  
   mit RS422/RS485 ..... 9-16  
   Parameterbeschreibung ..... 9-26  
   Parametrierung ..... 9-22  
   Protokolle ..... 9-3  
   Softwarehandshake ..... 9-31  
   Verkabelung ..... 9-14; 9-19

**D**

Darstellungsschicht.....	1-10
Datenbits.....	9-27
Datenkonsistenz	
Interbus.....	3-14
Profibus-DP .....	2-8
DBL.....	9-29
Demontage .....	20-8
Device Details.....	5-10
DeviceNet .....	5-1
Adresse.....	5-9
Adressierung.....	5-4
Anschluss .....	5-6
Baudrate .....	5-9
Beispiel .....	5-14
Buszugriff.....	5-4
Diagnose.....	5-17
-Manager .....	5-8
Projektierung.....	5-8
-Scanner .....	5-16
Schnittstelle zu Profibus .....	5-22
Test.....	5-10
Dezentrales System.....	1-5
DI16/Zähler	
Betriebsarten .....	13-31; 13-32
Ein- Ausgangsdaten .....	13-29
Übersicht.....	13-27
Diagnosefunktionen	
AI 2/AO 2x12Bit .....	18-15
AI 4x16Bit .....	16-15
AI 4x16Bit f .....	16-28
AI 8x16Bit .....	16-39
AO 4x12Bit.....	17-10
AO 4x12Bit f.....	17-18
CP.....	9-13
DeviceNet .....	5-17
Ethernet-Koppler.....	7-16
Profibus-DP .....	2-47
Diagnosekabel .....	9-13
Digitale Ausgabe.....	14-1
Digitale Ein-/Ausgabe .....	15-1
Digitale Eingabe .....	13-1
DIR.....	10-13
DLE-Zeichen.....	9-7
DP-Zyklus .....	2-7
DVI-Buchse.....	8-7

**E**

EasyConn .....	2-56
EDS-Datei.....	5-4
Einbaumaße .....	20-11
Einleitung.....	1-1
Emergency Objekt.....	4-15; 4-67
EMV .....	20-13
Grundregeln.....	20-14
Encoder .....	11-26
Endekennung .....	9-28
Ethernet.....	7-3
Ethernet-Koppler .....	7-1
Adressierung .....	7-14
Aufbau .....	7-9
Auslieferungszustand .....	7-9
Diagnose .....	7-16
Ethernet-Anschluss .....	7-10
Funktionscodes .....	7-21
GSD einbinden .....	7-15
HTTP-Web-Server .....	7-12; 7-16
Modbus-Bereiche 0x...4x.....	7-21
ModbusTCP .....	7-6; 7-20
Netzwerkplanung.....	7-7
OPC-Server .....	7-12
ORG-Format.....	7-25
Projektierung .....	7-11; 7-15
Protokolle.....	7-4
Siemens S5 Header.....	7-6; 7-13; 7-25
Socketprogrammierung.....	7-27
SPS-Header .....	7-26
Technische Daten .....	7-28
Zugriffsmöglichkeiten .....	7-11

**F**

Fehlermeldungen.....	4-67
AI 2/AO 2x12Bit .....	18-15
AI 4x16Bit .....	16-15
AI 4x16Bit f .....	16-28
AI 8x16Bit .....	16-39
AO 4x12Bit .....	17-10
AO 4x12Bit f .....	17-18
CAN-Bus.....	4-6; 4-10; 4-15
CP.....	9-48
DeviceNet .....	5-17
Interbus.....	3-16
Profibus-DP .....	2-50
FETCH (FB20) .....	9-51



- FETCH\_RK512 (FC 2) ..... 9-66  
 Flash Memory Card ..... 2-24  
 Flusskontrolle..... 9-27  
 Fref ..... 10-13  
 Frequenzberechnung..... 10-28  
 Funktions-Nr.  
   AI 2/AO 2 ..... 18-6  
   AI 4x16Bit ..... 16-7  
   AI 4x16Bit f ..... 16-24  
   AI 8x16Bit ..... 16-35  
   AO 4x12Bit..... 17-5  
   AO 4x12Bit f..... 17-14
- G**
- Gate ..... 10-13  
 Gerätebezogene Diagnose..... 2-48  
 Gray-Code ..... 10-7  
 Green Cable  
   Hinweise ..... 1-3  
 Group 2 only Device ..... 5-5  
 GSD-Datei..... 2-9
- H**
- Heartbeat ..... 4-15; 4-69  
 Hold-Eingang ..... 10-3  
 Hold-Funktion ..... 10-7  
 Hub ..... 7-3  
 Hysterese..... 13-18
- I**
- ID-Code..... 3-13  
 ID-Länge ..... 3-13  
 ID-Register..... 3-3  
 Inbetriebnahme  
   DeviceNet ..... 5-8  
   Interbus..... 3-15  
   Profibus..... 2-64  
 Interbus ..... 3-1  
   Anschluss ..... 3-10  
   Betriebsarten ..... 3-4  
   Datenkonsistenz ..... 3-14  
   Datenübertragung..... 3-5  
   Inbetriebnahme..... 3-15  
   Koppler IM 253IBS..... 3-7  
   Master ..... 3-3  
   Master konfigurieren ..... 3-17  
   Prozessdatenzuordnung..... 3-11
- K**
- Klemmen-Module ..... 19-10  
 Kommunikationsebenen..... 1-11  
 Kommunikationsprozessor ..... 9-1  
 Komponenten System 200V ..... 1-5
- L**
- Leitungsabschluss ..... 2-56  
 Lichtwellenleiter ..... 2-9  
 Low-Side..... 14-16  
 LWL ..... 2-57  
   Schnittstelle ..... 2-24
- M**
- Maschinelle Beschriftung..... 20-12  
 Messdaten erfassen  
   AI 4x16Bit ..... 16-13  
   AI 4x16Bit f ..... 16-26  
   AI 8x16Bit ..... 16-36  
 Messgate ..... 10-13  
 Messwertgeber anschließen..... 16-4  
 min\_slave\_interval..... 2-8  
 m-lesend..... 13-10; 13-36  
 MMC ..... 2-12  
 Modbus ..... 9-1  
   Beispiel ..... 9-39  
   Bibliothek einbinden ..... 9-33  
   Einsatz..... 9-32  
   Funktionscodes .... 7-21; 7-22; 9-36  
   Grundlagen..... 9-10  
   GSD einbinden ..... 9-33  
   Hantierungsbausteine..... 9-45  
     für CPU 21x ..... 9-61  
     für CPU 24x ..... 9-46  
   Parameter..... 9-25  
 Modul-ID ..... 4-17  
 Montage..... 20-1; 20-4  
 Montagemaße ..... 20-11  
 Mosfet-Ausgänge ..... 14-16  
 MotionControl Servo ..... 11-23  
   Anschluss eines Drehgebers 11-26  
   Antrieb ..... 11-26  
   Betriebsarten ..... 11-11; 11-31  
   Datenübergabe an CPU ..... 11-37  
   Datenübergabe an FM 254... 11-30  
   Feldkennung..... 11-30  
   Komponenten ..... 11-24  
   Parameter..... 11-28

Parametrierung .....	11-29	BIOS-Setup .....	8-13
Verdrahtung .....	11-27	Projektierung .....	8-3
MotionControl Stepper .....	11-3	Prozessabbild .....	8-10
Ausgänge .....	11-6	Register .....	8-21
Datenübergabe an CPU .....	11-15	PDO .....	4-20
Datenübergabe an FM 253 .....	11-8	Linking .....	4-22
Eingänge .....	11-6	Übertragungsart .....	4-23
Komponenten .....	11-4	Periodendauerberechnung .....	10-31
Parametrierung .....	11-9	Peripheriemodule .....	1-5
Standardfunktionsbausteine .....	11-21	Poll only Device .....	5-5
Verdrahtung .....	11-7	Priorität .....	9-29
MotionControl-Module .....	11-1	Profibus-DP .....	2-1
Multi-In-/Output .....	18-4	Adressierung .....	2-9
<b>N</b>		Aufbau Richtlinien .....	2-54
Netzwerkschicht .....	1-9	Beispiele .....	2-66
NMT .....	4-68	Datenübertragung .....	2-7
Node Guarding .....	4-14; 4-69	Fehlermeldungen .....	2-50
Normdiagnose-Daten .....	2-48	GSD .....	2-9
NPN .....	14-16	Inbetriebnahme .....	2-64
<b>O</b>		Master .....	2-5
Option-Byte		Master mit LWL .....	2-23
AI 2/AO 2 .....	18-14	Betriebszustände .....	2-25
AI 4x16Bit .....	16-14	Einsatz an CPU 21x .....	2-26
AI 8x16Bit .....	16-38	Flash Memory Card .....	2-24
Optoelektrischer Profibus .....	2-63	Projektierung .....	2-27
<b>P</b>		Master mit RS485 .....	2-10
Parametrierung		Betriebszustände .....	2-12
AI 2/AO 2 .....	18-13	Einsatz an CPU 21x .....	2-13
AI 2/AO 2 .....	18-5	Projektierung .....	2-14
AI 4x16Bit .....	16-6; 16-13	Urlöschen .....	2-22
AI 4x16Bit f .....	16-26	Multi Master System .....	2-61
AI 8x16Bit .....	16-37	Netze .....	2-60
AO 4x12Bit .....	17-9	Optischer Profibus .....	2-62
AO 4x12Bit f .....	17-17	Redundanzstatus .....	2-53
CP .....	9-22	Schnittstelle zu DeviceNet .....	5-22
DeviceNet .....	5-11	Slave .....	2-5; 2-33
MotionControl Servo .....	11-29	Blockschaltbild .....	2-43
MotionControl Stepper .....	11-9	Datenverkehr .....	2-6
Profibus DP-Slave .....	2-46	Diagnosefunktionen .....	2-47
SERCOS .....	6-8	GSD einbinden .....	2-44
SSI-Interface .....	10-6	mit 24xDO .....	2-39
Zähler-Modul FM 250 .....	10-11	Parameter .....	2-46
Parity .....	9-27	Projektierung .....	2-44
Passivbetrieb .....	9-6	IM 308-B von Siemens ..	2-45
PC 288 - CPU .....	8-1	S7-400 von Siemens .....	2-45
		redundant .....	2-36
		Stecker .....	2-56
		Token-Passing-Verfahren .....	2-6

- Übertragungsprotokoll ..... 2-6
- Protokolle ..... 9-4
- Prozeduren ..... 9-5
- Prozessabbild
  - DeviceNet ..... 5-16
  - Interbus ..... 3-11
  - PC 288 - CPU ..... 8-10
  - SSI-Modul ..... 10-8
  - Zähler-Modul ..... 10-11
- PS2-Buchsen ..... 8-6
- Pulse ..... 10-13
- Q**
- QVZ ..... 9-29
- R**
- RECEIVE (FB4) ..... 9-49
- RECEIVE\_ASCII\_3964 (FC 1) ..... 9-64
- Receivebuffer ..... 9-28
- Referenzfrequenz ..... 10-13
- Register PC 288 - CPU ..... 8-21
- Relais-Ausgabe ..... 14-20; 14-22
- Relais-Ausgabe bistabil ..... 14-24
- RES ..... 10-13
- RS232/RS422-Umschaltung ..... 8-22
- RS232C-Schnittstelle ..... 9-11; 9-14
- RS422-Schnittstelle ..... 9-16; 9-19
- RS485-Schnittstelle ..... 9-16; 9-20
  - Profibus ..... 2-11
- Rückwandbusverbinder ..... 20-2
- Ruhepegel ..... 9-21
- S**
- S/R\_ALL (FB23) ..... 9-55
- S/R\_ALL\_RK512 (FC 4) ..... 9-72
- S5-Format von Siemens 16-18; 16-21
- S7-Format von Siemens 16-12; 16-25
- Schichtenmodell ..... 1-8
- Schieberegister ..... 3-3
- Schirmung von Leitungen ..... 20-15
- SDO ..... 4-24
  - Error-Codes ..... 4-25
- Segmentlänge unter Profibus ..... 2-54
- SEND (FB22) ..... 9-53
- SEND (FB3) ..... 9-47
- SEND\_ASCII\_STX\_3964 (FC 0) ..... 9-62
- SEND\_RK512 (FC 3) ..... 9-69
- SERCOS ..... 6-1
- Adressierung ..... 6-4
- Anschluss ..... 6-6
- Baudrate ..... 6-8
- set\_address\_table ..... 8-11
- Sicherungsschicht ..... 1-9
- SIP-Tool ..... 2-20
- Sitzungsschicht ..... 1-10
- Softwarehandshake ..... 9-31
- Solid State ..... 14-26; 14-28
- Spannungsversorgungen ..... 12-1
  - Montage ..... 12-8
  - PS 207/2, 2A ..... 12-4
  - PS 207/2CM, 2A ..... 12-6
  - Sicherheitshinweise ..... 12-2
  - Verdrahtung ..... 12-9
- SSI-Interface FM 250S ..... 10-2; 10-3
  - Codierung ..... 10-7
  - Ein-/Ausgabedaten ..... 10-8
  - Leitungslängen ..... 10-5
  - Parametrierung ..... 10-6
- Startkennungen ..... 9-28
- Statusbyte ..... 10-11
- Stopbits ..... 9-27
- Störeinträge ..... 20-13
- Stromaufnahmen ..... 20-5
- STX/ETX ..... 9-4; 9-23
- STX-Wiederholungen ..... 9-29
- Switch ..... 7-3; 19-7
  - DIL-Schalter ..... 19-9
- SYNCHRON\_RESET (FC 9) ..... 9-74
- SYNCRON (FB25) ..... 9-57
- Systemerweiterungen ..... 19-1
- T**
- Telegrammaufbau
  - CAN-Bus ..... 4-18
- Time-out-Zeiten ..... 9-6
- TMO ..... 9-28
- Tragschienen ..... 20-2
- Transportschicht ..... 1-9
- Twisted Pair ..... 7-3
  - Einschränkungen ..... 7-8
- Twisted-Pair-Anschluss ..... 8-6
- U**
- UB4x ..... 13-23; 14-11
- Übersicht System 200V ..... 1-4
- Umgebungsbedingungen ..... 1-7

**V**

vbus\_api.c..... 8-10  
 vbus\_api.h..... 8-10  
 vbus\_read\_pword ..... 8-11  
 vbus\_set\_param ..... 8-12  
 vbus\_write\_pword..... 8-11  
 V-Bus-Zyklus..... 2-7  
 Verdrahtung ..... 20-9  
 Versionsdaten ..... 8-19  
 VIPA Configuration ..... 8-18

**W**

Watchdog..... 8-19  
 Wide-Range-Input..... 12-3  
 WinNCS  
   Ethernet-Koppler..... 7-15  
   unter Profibus-DP ..... 2-21; 2-32

**Z**

Zählermodi..... 10-14; 10-50  
   (0) 32 Bit Zähler..... 10-14; 10-50  
   (1) Encoder 1 Flanke ..... 10-16  
   (12,13) 32 Bit Zähler ..... 10-25  
   (14,15) 32 Bit Zähler ..... 10-26  
   (16) Frequenzmessung ..... 10-28  
   (17) Periodendauermessung 10-30

(18) Frequenzmessung ..... 10-32  
 (19) Periodendauermessung 10-34  
 (20) Pulsmessung ..... 10-36  
 (21) Pulsmessung ..... 10-38  
 (22) Pulsmessung ..... 10-40  
 (23) One Shot..... 10-42  
 (24) One Shot..... 10-44  
 (25) One Shot..... 10-46  
 (26) One Shot..... 10-48  
 (28) Encoder 1 Flanke..... 10-52  
 (29) Encoder 2 Flanken..... 10-54  
 (3) Encoder 2 Flanken..... 10-18  
 (30) Encoder 4 Flanken..... 10-56  
 (5) Encoder 4 Flanken..... 10-20  
 (6) Pulsmessung ..... 10-22  
 (8...11) Zähler, 2 Eingänge... 10-24  
 Zähler-Modul FM 250 ..... 10-9  
   Ein-/Ausgabedaten ..... 10-11  
   remanent ..... 10-11  
   Zählermodi ..... 10-12  
 Zähler-Module ..... 10-1  
 Zeichenrahmen..... 9-3  
 Zentrales System..... 1-5  
 ZNA ..... 9-28  
 ZVZ..... 9-28