



# VIPA System 200V



## IM | Handbuch

HB97D\_IM | Rev. 12/33

August 2012

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

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

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

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

VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH

Ohmstraße 4, D-91074 Herzogenaurach, Germany

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

Fax.: +49 9132 744 1864

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

<http://www.vipa.de>

## **Hinweis**

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

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

## **CE-Konformität**

Hiermit erklärt VIPA GmbH, dass die Produkte und Systeme mit den grundlegenden Anforderungen und den anderen relevanten Vorschriften der folgenden Richtlinien übereinstimmen:

- 2004/108/EG Elektromagnetische Verträglichkeit
- 2006/95/EG Niederspannungsrichtlinie

Die Übereinstimmung ist durch CE-Zeichen gekennzeichnet.

## **Informationen zur Konformitätserklärung**

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

## **Warenzeichen**

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

SPEED7 ist ein eingetragenes Warenzeichen der profichip GmbH.

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

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

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

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

## **Dokument-Support**

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefax: +49 9132 744 1204

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

## **Technischer Support**

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

VIPA GmbH, Ohmstraße 4, 91074 Herzogenaurach, Germany

Telefon: +49 9132 744 1150 (Hotline)

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

## Über dieses Handbuch

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

### Überblick

#### **Teil 1: Grundlagen**

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: 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.

#### **Teil 3: 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 4: 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 5: 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 6: 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 7: 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 8: 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 9: Buserweiterung**

In diesem Kapitel wird die Buserweiterung IM 260 und IM 261 beschrieben, die das Aufteilen einer System 200V Zeile auf bis zu 4 Zeilen ermöglicht.

## Inhaltsverzeichnis

<b>Benutzerhinweise</b> .....	<b>1</b>
<b>Sicherheitshinweise</b> .....	<b>2</b>
<b>Teil 1 Grundlagen</b> .....	<b>1-1</b>
Sicherheitshinweise für den Benutzer .....	1-2
Übersicht.....	1-3
Komponenten.....	1-4
Allgemeine Beschreibung System 200V.....	1-5
<b>Teil 2 Montage und Aufbaurichtlinien</b> .....	<b>2-1</b>
Übersicht.....	2-2
Montage .....	2-5
Verdrahtung .....	2-8
Einbaumaße.....	2-10
Aufbaurichtlinien.....	2-12
<b>Teil 3 Profibus-DP</b> .....	<b>3-1</b>
Systemübersicht.....	3-2
Systemübersicht.....	3-2
Grundlagen .....	3-5
IM 208DP - Master - Aufbau.....	3-13
IM 208DP - Master - Einsatz an einer CPU 21x.....	3-17
IM 208DP - Master - Projektierung .....	3-18
IM 208DP - Master - Slave-Betrieb.....	3-28
IM 208DP - Master - Urlöschen .....	3-32
IM 208DP - Master - Firmwareupdate .....	3-33
IM 253-1DPx0 - DP-V0-Slave - Aufbau .....	3-35
IM 253-2DP20 - DP-V0-Slave mit DO 24xDC 24V - Aufbau .....	3-38
IM 253-2DP50 - DP-V0-Slave (redundant) - Aufbau.....	3-42
IM 253-xDPx0 - DP-V0-Slave - Blockschaltbild .....	3-45
IM 253-xDPx0 - DP-V0-Slave - Projektierung.....	3-46
IM 253-xDPx0 - DP-V0-Slave - Parameter .....	3-48
IM 253-xDPx0 - DP-V0-Slave - Diagnosefunktionen .....	3-49
IM 253-xDPx1 - DP-V1-Slave - Aufbau.....	3-56
IM 253-xDPx1 - DP-V1-Slave - Blockschaltbild .....	3-60
IM 253-xDPx1 - DP-V1-Slave - Projektierung.....	3-61
IM 253-xDPx1 - DP-V1-Slave - Parameter .....	3-63
IM 253-xDPx1 - DP-V1-Slave - Diagnosefunktionen .....	3-68
IM 253-xDPx1 - DP-V1-Slave - Firmware-Update.....	3-76
IM 253-xDPx1 - DP-V1-Slave - I&M-Daten .....	3-77
Aufbaurichtlinien.....	3-79
Inbetriebnahme .....	3-89
Einsatz der Diagnose-LEDs .....	3-90
Beispiele zur Profibus-Kommunikation .....	3-91
Technische Daten .....	3-99

<b>Teil 4</b>	<b>Interbus .....</b>	<b>4-1</b>
	Systemübersicht.....	4-2
	Grundlagen .....	4-3
	IM 253IBS - Interbus-Koppler - Aufbau.....	4-7
	Anschluss an Interbus .....	4-10
	Einsatz im Interbus.....	4-11
	Inbetriebnahme .....	4-15
	Technische Daten .....	4-18
<b>Teil 5</b>	<b>CANopen .....</b>	<b>5-1</b>
	Systemübersicht.....	5-2
	Grundlagen .....	5-4
	IM 208CAN - CANopen-Master - Aufbau.....	5-6
	IM 208CAN - CANopen-Master - Projektierung .....	5-8
	IM 208CAN - CANopen-Master - Firmwareupdate .....	5-15
	IM 208CAN - CANopen-Master - Betriebsarten .....	5-16
	IM 208CAN - CANopen-Master - Prozessabbild .....	5-17
	IM 208CAN - CANopen-Master - Telegrammaufbau .....	5-19
	IM 208CAN - CANopen-Master - Objekt-Verzeichnis .....	5-24
	IM 253CAN - CANopen-Slave - Aufbau.....	5-38
	IM 253CAN - CANopen-Slave DO 24xDC 24V - Aufbau .....	5-43
	IM 253CAN - CANopen-Slave - Schnelleinstieg .....	5-47
	IM 253CAN - CANopen-Slave - Baudrate und Modul-ID .....	5-51
	IM 253CAN - CANopen-Slave - Telegrammaufbau .....	5-52
	IM 253CAN - CANopen-Slave - PDO .....	5-54
	IM 253CAN - CANopen-Slave - SDO .....	5-58
	IM 253CAN - CANopen-Slave - Objekt-Verzeichnis .....	5-60
	IM 253CAN - CANopen-Slave - Emergency Object.....	5-101
	IM 253CAN - CANopen-Slave - NMT - Netzwerk Management.....	5-102
	Technische Daten .....	5-104
<b>Teil 6</b>	<b>DeviceNet.....</b>	<b>6-1</b>
	Systemübersicht.....	6-2
	Grundlagen .....	6-3
	IM 253DN - DeviceNet-Koppler - Aufbau.....	6-5
	Projektierung unter Einsatz des DeviceNet-Managers .....	6-8
	Einstellung von Baudrate und DeviceNet-Adresse .....	6-9
	Test am DeviceNet-Bus .....	6-10
	Module im DeviceNet-Manager parametrieren .....	6-11
	I/O-Adressierung des DeviceNet-Scanners .....	6-16
	Diagnose.....	6-17
	Technische Daten .....	6-22
<b>Teil 7</b>	<b>SERCOS - Ersatzteil .....</b>	<b>7-1</b>
	Systemübersicht.....	7-2
	Grundlagen .....	7-3
	IM 253Sercos - SERCOS-Koppler - Aufbau .....	7-5
	Grundparametrierung über Adresseinsteller.....	7-8
	SERCOS Identifier .....	7-10
	Beispiel zur automatischen ID-Zuweisung.....	7-13
	Technische Daten .....	7-22

<b>Teil 8 Ethernet-Koppler .....</b>	<b>8-1</b>
Systemübersicht.....	8-2
Grundlagen Ethernet.....	8-3
Planung eines Netzwerks.....	8-7
IM 253NET - Ethernet-Koppler - Aufbau.....	8-9
Zugriffsmöglichkeiten auf den Ethernet-Koppler.....	8-11
Prinzip der automatischen Adressierung .....	8-14
Projektierung unter WinNCS .....	8-15
Diagnose und Test mittels Internet Browser.....	8-16
ModbusTCP .....	8-20
Modbus-Funktionscodes .....	8-21
Siemens S5 Header Protokoll .....	8-26
Prinzip der Alarmbearbeitung .....	8-28
Programmierbeispiel .....	8-32
Technische Daten .....	8-33
<b>Teil 9 Buserweiterung IM 260 - IM 261 .....</b>	<b>9-1</b>
Einsatzbereich.....	9-2
Verkabelung.....	9-3
Statusanzeigen .....	9-4
Technische Daten .....	9-5
<b>Anhang.....</b>	<b>A-1</b>
Index.....	A-1



## Benutzerhinweise

**Zielsetzung und Inhalt** Dieses Handbuch beschreibt 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 in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.

**Orientierung im Dokument** Als Orientierungshilfe stehen im Handbuch zur Verfügung:

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

## 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 Grundlagen

### 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.

Nachfolgend sind beschrieben:

- Vorstellung des System 200V
- Allgemeine Beschreibung, wie Maße, Montage, Betriebssicherheit und Umgebungsbedingungen

### Inhalt

Thema	Seite
<b>Teil 1 Grundlagen</b> .....	<b>1-1</b>
Sicherheitshinweise für den Benutzer .....	1-2
Übersicht .....	1-3
Komponenten .....	1-4
Allgemeine Beschreibung System 200V .....	1-5

## 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!

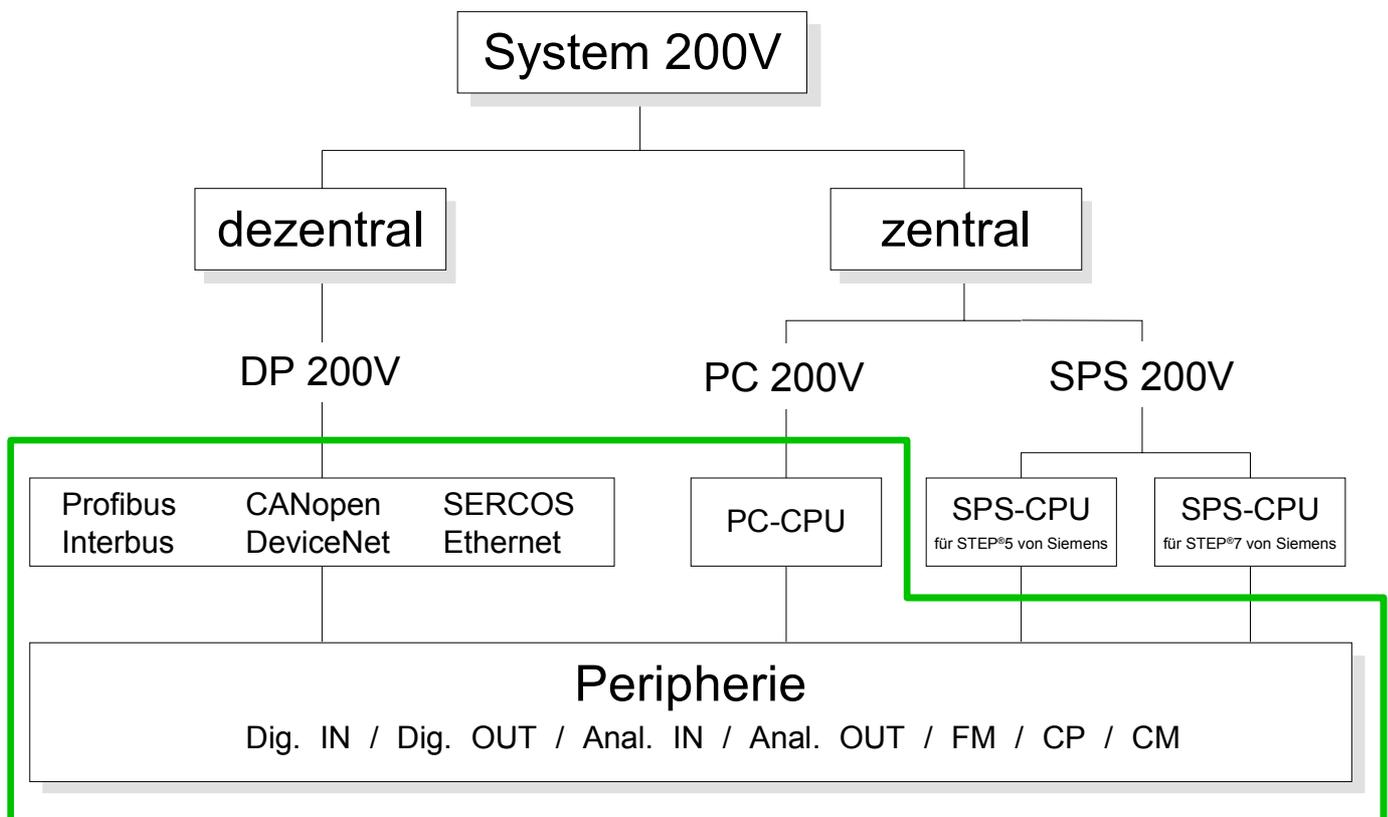
Bei Arbeiten mit und an elektrostatisch gefährdeten Baugruppen ist auf ausreichende Erdung des Menschen und der Arbeitsmittel zu achten.

## 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 können Sie mit dem VIPA Projektierool WinNCS bzw. mit dem SIMATIC Manager projektieren.
- Die Anbindung an weitere Feldbusgeräte ermöglichen Slaves für Interbus, CANopen, DeviceNet, SERCOS und Ethernet.
- 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.
- Einbindung über GSD-Datei** 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 Siemens SIMATIC Manager jederzeit gegeben ist.
- Für jede Systemfamilie erhalten Sie eine GSD-Datei. Aktuelle GSD-Dateien finden Sie unter [ftp.vipa.de/support](http://ftp.vipa.de/support).

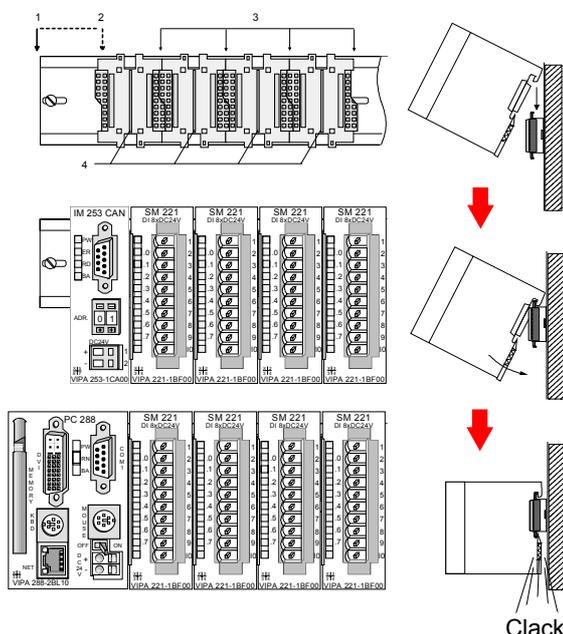
## Allgemeine Beschreibung System 200V

### Aufbau/Maße

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

### Montage

Bitte beachten Sie, dass Sie Kopfmodule, 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

Sie können maximal 32 Module stecken, hierbei ist zu beachten, dass der **Summenstrom** von **3,5A** am Rückwandbus nicht überschritten wird!

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

### 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



## Teil 2 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
- EMV-Richtlinien zum Aufbau eines System 200V

### Inhalt

Thema	Seite
<b>Teil 2 Montage und Aufbaurichtlinien</b> .....	<b>2-1</b>
Übersicht.....	2-2
Montage.....	2-5
Verdrahtung.....	2-8
Einbaumaße.....	2-10
Aufbaurichtlinien.....	2-12

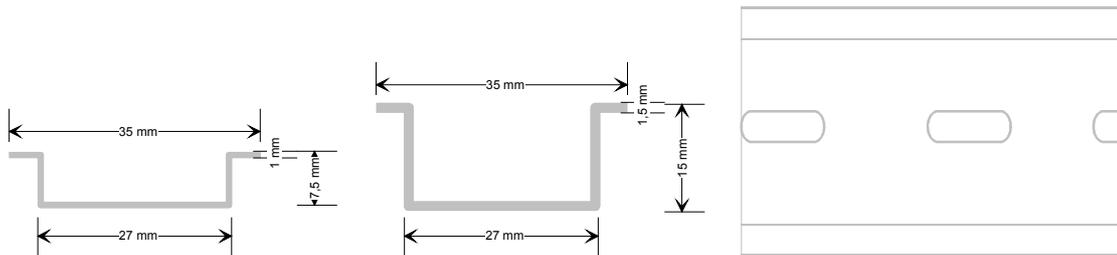
## Übersicht

### Allgemein

Die einzelnen Module werden direkt auf eine Tragschiene montiert und über Rückwandbusverbinder, die vorher in die Profilschiene eingelegt werden, gekoppelt.

### Tragschienen

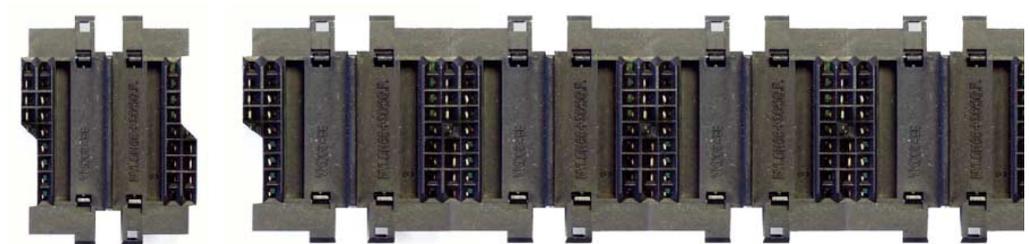
Für die Montage können Sie folgende 35mm-Normprofilschiene verwenden:



### 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:

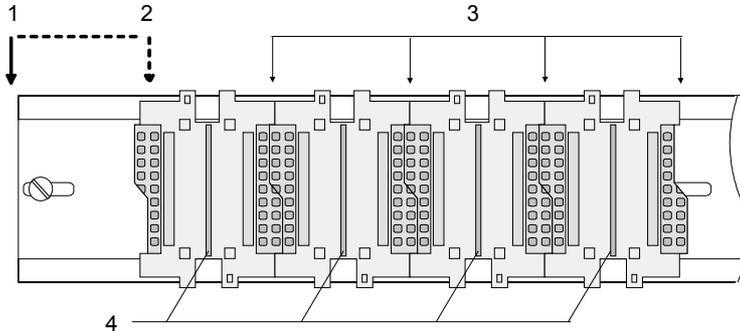


Der Busverbinder wird in die Tragschiene eingelegt, bis dieser sicher einrastet, so dass die Bus-Anschlüsse aus der Tragschiene herauschauen.

**Montage auf Tragschiene**

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

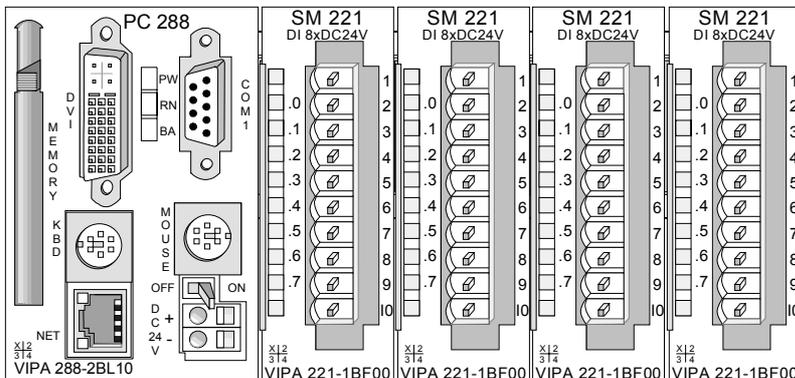
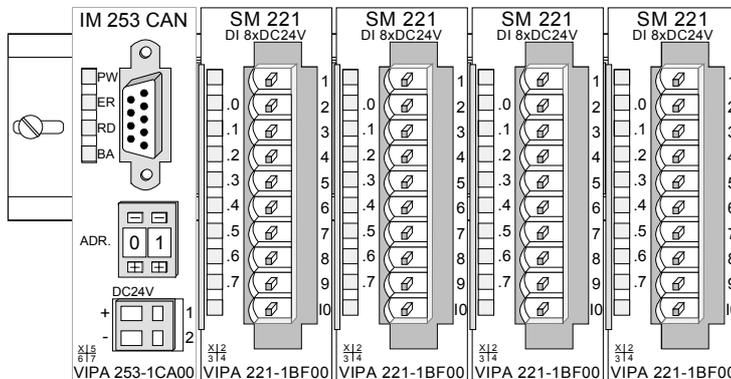
Die einzelnen Modulsteckplätze sind durch Führungsleisten abgegrenzt.



- [1] Kopfmódul, wie PC, CPU, Bus-Koppler, wenn doppelt breit
- [2] Kopfmódul (einfach breit)
- [3] Peripheriemodule
- [4] Führungsleisten

**Hinweis**

Sie können maximal 32 Module stecken. Hierbei ist zu beachten, dass der **Summenstrom** von **3,5A** am Rückwandbus nicht überschritten wird!



**Montage unter Berücksichtigung der Stromaufnahme**

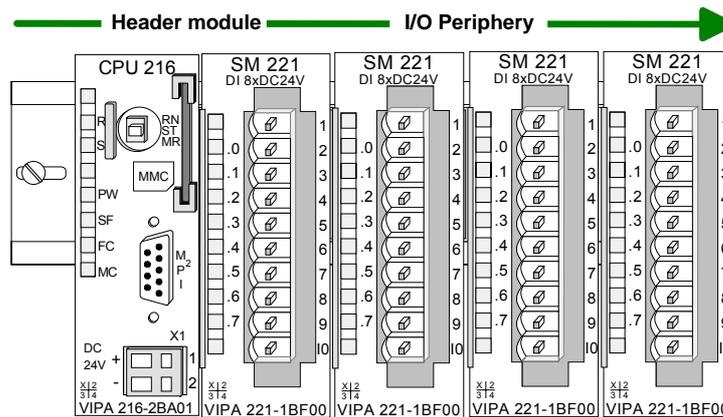
- Verwenden Sie möglichst lange Busverbinder.
- Ordnen Sie Module mit hohem Stromverbrauch direkt rechts neben Ihrem Kopfmódul an. Unter [ftp.vipa.de/manuals/system200v](http://ftp.vipa.de/manuals/system200v) finden Sie alle Stromaufnahmen des System 200V in einer Liste zusammengefasst.

**Aufbau waagrecht bzw. senkrecht**

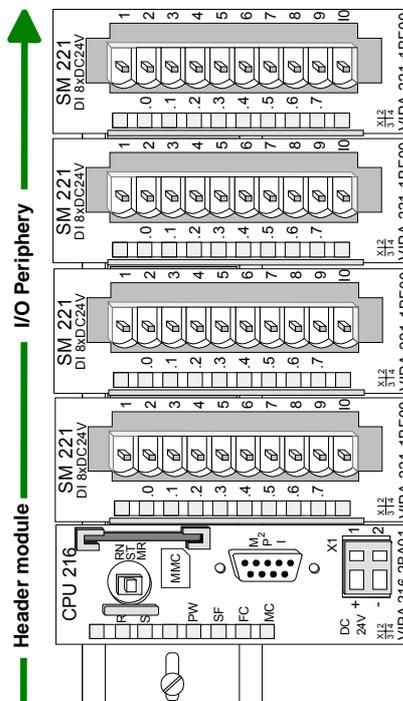
Sie haben die Möglichkeit das System 200V waagrecht oder senkrecht aufzubauen. Beachten Sie bitte die hierbei zulässigen Umgebungstemperaturen:

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

Der waagrechte Aufbau beginnt immer links mit einem Kopfmodul (CPU, Buskoppler, PC); rechts daneben sind die Peripherie-Module zu stecken. Es dürfen maximal 32 Peripherie-Module gesteckt werden.



Der senkrechte Aufbau erfolgt gegen den Uhrzeigersinn um 90° gedreht.

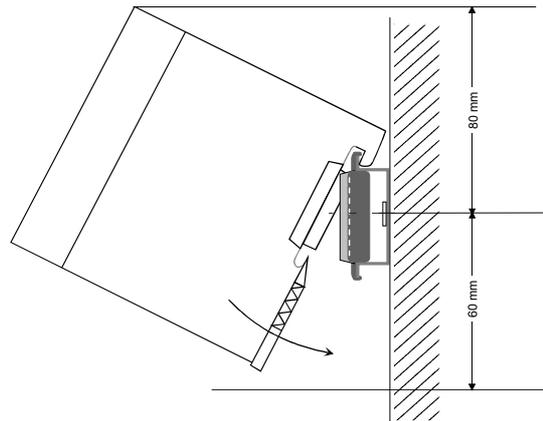


## Montage

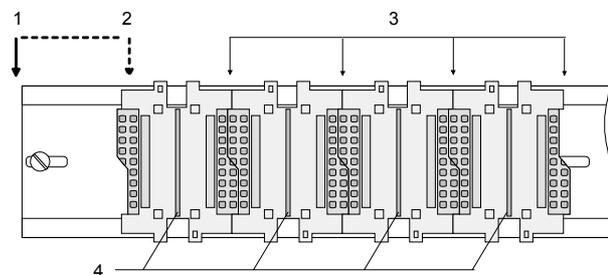


### Bitte bei der Montage beachten!

- Schalten Sie die Stromversorgung aus, bevor Sie Module stecken bzw. abziehen!
- Bitte beachten Sie, dass Sie ab der Mitte der Busschiene nach oben einen Modul-Montageabstand von mindestens 80mm und nach unten von 60mm einhalten.



- Eine Zeile wird immer von links nach rechts aufgebaut und beginnt immer mit einem Kopfmodul (PC, CPU, Buskoppler).



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

- Module müssen immer direkt nebeneinander gesteckt werden. 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.

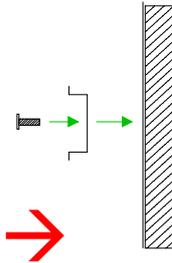


### Hinweis!

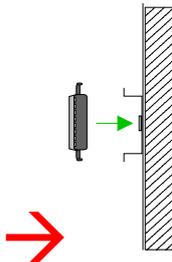
Am Rückwandbus dürfen sich maximal 32 Module befinden. Hierbei ist zu beachten, dass der **Summenstrom** von **3,5A** am Rückwandbus nicht überschritten wird!

**Montage  
Vorgehensweise**

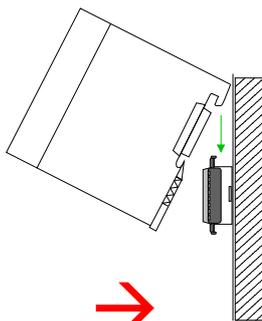
Die nachfolgende Abfolge stellt die Montageschritte in der Seitenansicht dar.



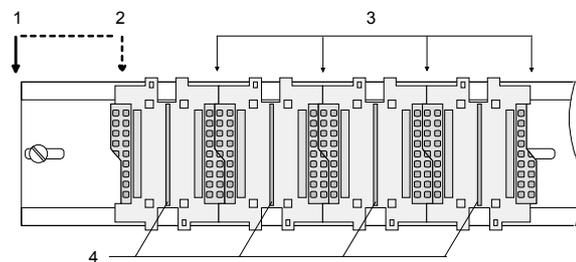
- Montieren Sie die Tragschiene! Bitte beachten Sie, dass Sie ab der Mitte der Busschiene nach oben einen Modul-Montageabstand von mindestens 80mm und nach unten von 60mm einhalten.



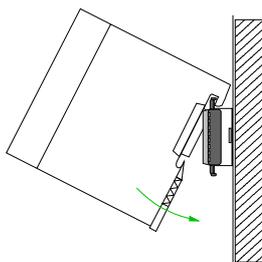
- Drücken Sie den Busverbinder in die Tragschiene, bis dieser sicher einrastet, so dass die Bus-Anschlüsse aus der Tragschiene herauschauen. Sie haben nun die Grundlage zur Montage Ihrer Module.



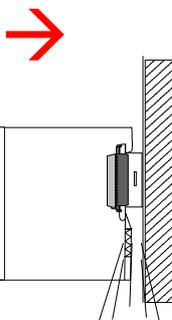
- Beginnen Sie ganz links mit dem Kopfmodul, wie CPU, PC oder Buskoppler und stecken Sie rechts daneben Ihre Peripheriemodule.



- [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.



Clack

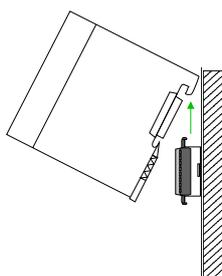
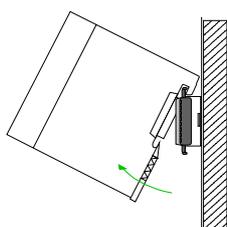
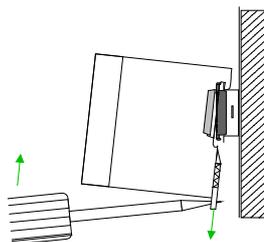
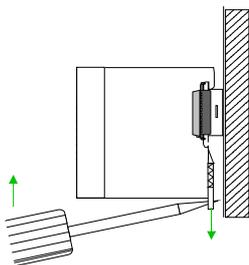
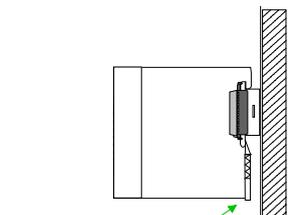


**Achtung!**

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.
- Stecken 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.

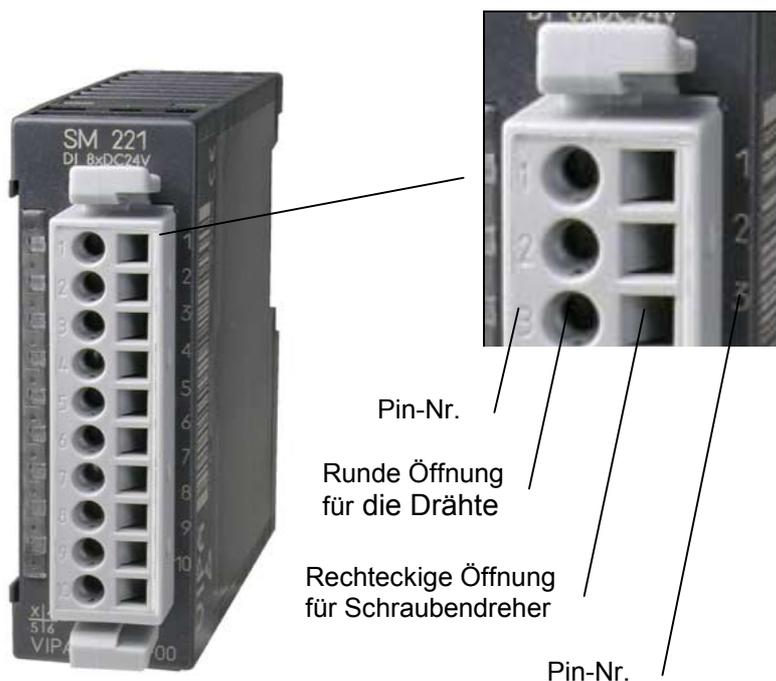
Bei der Verdrahtung werden Steckverbinder mit Federklemmtechnik eingesetzt.

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 0,08mm<sup>2</sup> bis 2,5mm<sup>2</sup> (bis 1,5mm<sup>2</sup> bei 18poligen) anschließen.

Folgende Abbildung zeigt ein Modul mit einem 10poligen Steckverbinder.



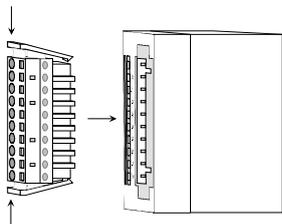
### Hinweis!

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

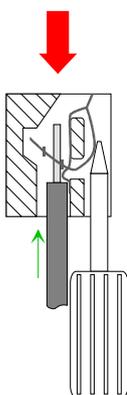


- Stecken Sie den Steckverbinder auf das Modul bis dieser hörbar einrastet. Drücken Sie hierzu während des Steckens, wie gezeigt, die beiden Verriegelungsklinken zusammen. Der Steckverbinder ist nun in einer festen Position und kann leicht verdrahtet werden.

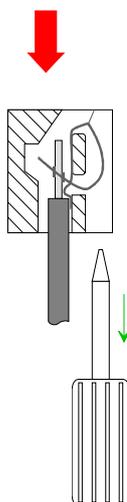
Die nachfolgende Abfolge stellt die Schritte der Verdrahtung in der Draufsicht dar.



- 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.



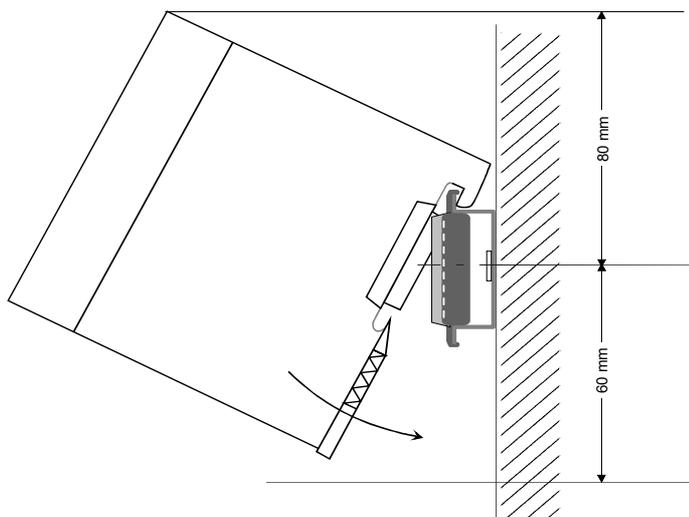
Verdrahten Sie zuerst die Versorgungsleitungen (Spannungsversorgung) und dann die Signalleitungen (Ein- und Ausgänge)!

# Einbaumaße

**Übersicht** Hier finden Sie alle wichtigen Maße des System 200V.

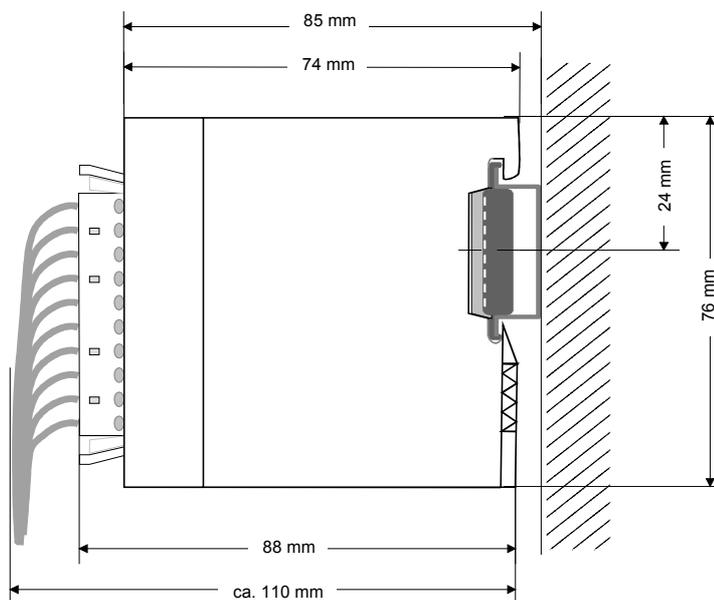
**Maße Grundgehäuse**  
 1fach breit (HxBxT) in mm: 76 x 25,4 x 74  
 2fach breit (HxBxT) in mm: 76 x 50,8 x 74

## Montagemaße

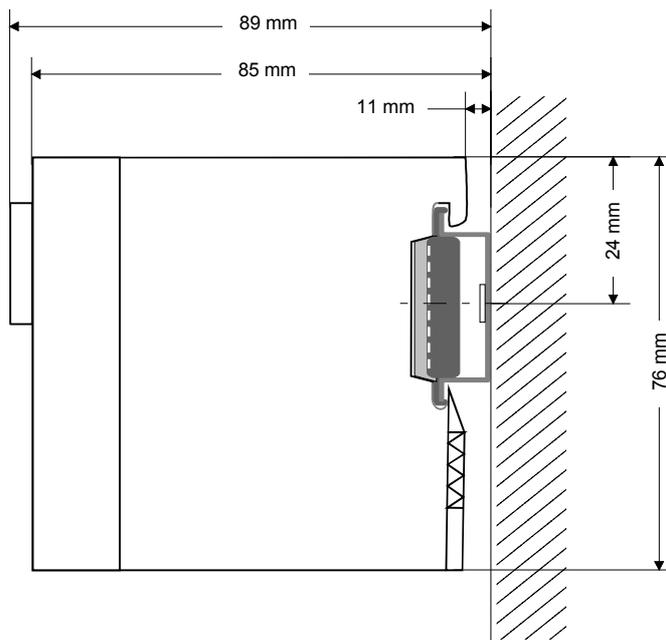


## Maße montiert und verdrahtet

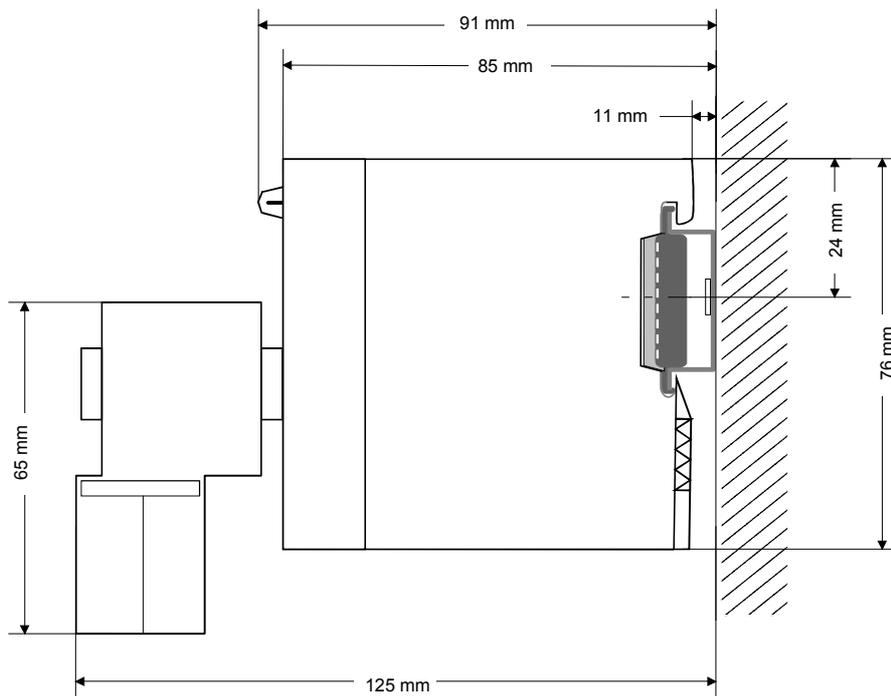
Ein- / Ausgabe-  
module



Funktionsmodule



CPUs (hier mit VIPA EasyConn)



## Aufbaurichtlinien

<b>Allgemeines</b>	<p>Die Aufbaurichtlinien enthalten Informationen über den störsicheren Aufbau des System 200V. Es wird beschrieben, wie Störungen in Ihre Steuerung gelangen können, wie die elektromagnetische Verträglichkeit (EMV) sicher gestellt werden kann und wie bei der Schirmung vorzugehen ist.</p>
<b>Was bedeutet EMV?</b>	<p>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.</p> <p>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.</p>
<b>Mögliche Störeinträge</b>	<p>Elektromagnetische Störungen können sich auf unterschiedlichen Pfaden in Ihre Steuerung einkoppeln:</p> <ul style="list-style-type: none"><li>• Felder</li><li>• E/A-Signalleitungen</li><li>• Bussystem</li><li>• Stromversorgung</li><li>• Schutzleitung</li></ul> <p>Je nach Ausbreitungsmedium (leitungsgebunden oder -ungebunden) und Entfernung zur Störquelle gelangen Störungen über unterschiedliche Kopplungsmechanismen in Ihre Steuerung.</p> <p>Man unterscheidet:</p> <ul style="list-style-type: none"><li>• galvanische Kopplung</li><li>• kapazitive Kopplung</li><li>• induktive Kopplung</li><li>• Strahlungskopplung</li></ul>

**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 Löschielern, die 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

## Teil 3 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-V0 / DP-V1
- Aufbau und Projektierung der Profibus-Master IM 208DP und Profibus-Slaves IM 253DP
- Projektierbeispiele und Technische Daten

### Inhalt

Thema	Seite
<b>Teil 3 Profibus-DP</b> .....	<b>3-1</b>
Systemübersicht.....	3-2
Grundlagen .....	3-5
IM 208DP - Master - Aufbau.....	3-13
IM 208DP - Master - Einsatz an einer CPU 21x.....	3-17
IM 208DP - Master - Projektierung .....	3-18
IM 208DP - Master - Slave-Betrieb.....	3-28
IM 208DP - Master - Urlöschen .....	3-32
IM 208DP - Master - Firmwareupdate .....	3-33
IM 253-1DPx0 - DP-V0-Slave - Aufbau .....	3-35
IM 253-2DP20 - DP-V0-Slave mit DO 24xDC 24V - Aufbau .....	3-38
IM 253-2DP50 - DP-V0-Slave (redundant) - Aufbau.....	3-42
IM 253-xDPx0 - DP-V0-Slave - Blockschaltbild .....	3-45
IM 253-xDPx0 - DP-V0-Slave - Projektierung.....	3-46
IM 253-xDPx0 - DP-V0-Slave - Parameter .....	3-48
IM 253-xDPx0 - DP-V0-Slave - Diagnosefunktionen .....	3-49
IM 253-xDPx1 - DP-V1-Slave - Aufbau.....	3-56
IM 253-xDPx1 - DP-V1-Slave - Blockschaltbild .....	3-60
IM 253-xDPx1 - DP-V1-Slave - Projektierung.....	3-61
IM 253-xDPx1 - DP-V1-Slave - Parameter .....	3-63
IM 253-xDPx1 - DP-V1-Slave - Diagnosefunktionen .....	3-68
IM 253-xDPx1 - DP-V1-Slave - Firmware-Update.....	3-76
IM 253-xDPx1 - DP-V1-Slave - I&M-Daten.....	3-77
Aufbau Richtlinien.....	3-79
Inbetriebnahme .....	3-89
Einsatz der Diagnose-LEDs .....	3-90
Beispiele zur Profibus-Kommunikation .....	3-91
Technische Daten .....	3-99

## Systemübersicht

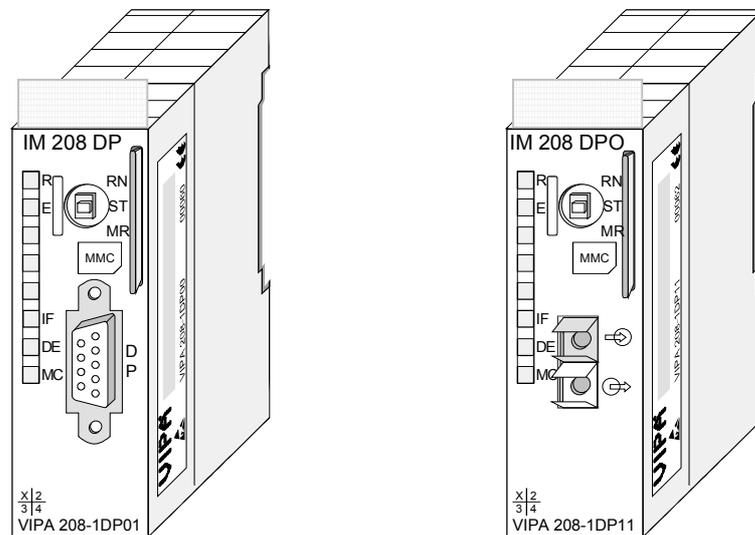
### System 200V Profibus-DP Module

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

- Profibus-DP-Master
- Profibus-DP-Slave mit DP-V0 / DP-V1
- Profibus-DP-Slave Kombimodule
- CPU 21x DP - CPU 21x für S7 von Siemens mit integriertem Profibus-DP-Slave (siehe Handbuch HB97\_CPU)
- 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 bzw. im Siemens SIMATIC Manager
- Projektierdaten werden im internen Flash-ROM abgelegt oder auf MMC gespeichert.

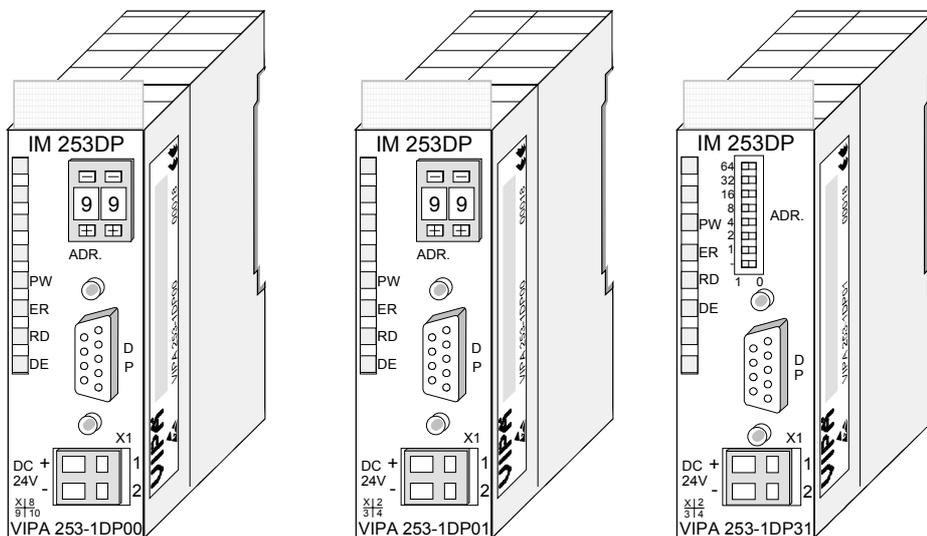


Bestelldaten  
DP-Master

Typ	Bestellnummer	Beschreibung	Seite
IM 208DP	VIPA 208-1DP01	Profibus-DP-Master mit RS485	3-13
IM 208DPO	VIPA 208-1DP11	Profibus-DP-Master mit LWL	

**Profibus-DP-Slaves**

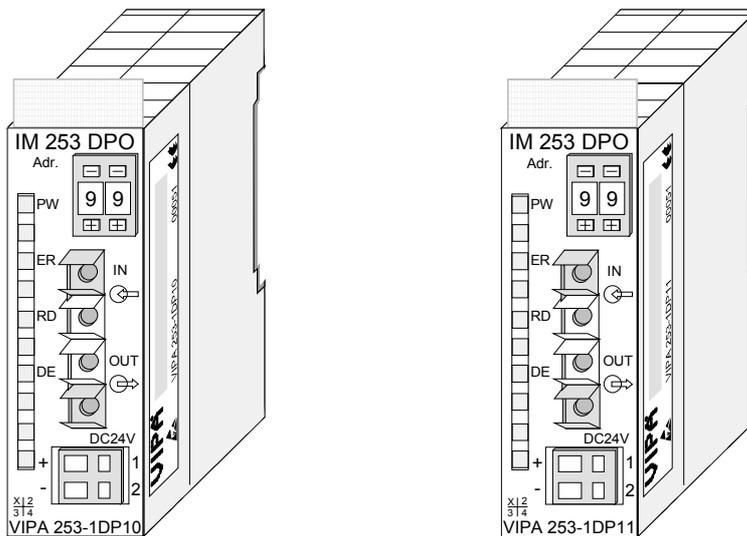
- Ausführung mit RS485-Schnittstelle oder Lichtwellenleiter-Anschluss
- Ausführung mit DP-V1-Interface
- Online Diagnoseprotokoll



Bestelldaten

Typ	Bestellnummer	Beschreibung	Seite
IM 253DP	VIPA 253-1DP00	Profibus-DP-V0-Slave	3-35
IM 253DP	VIPA 253-1DP01	Profibus-DP-V0/V1-Slave	3-56
IM 253DP	VIPA 253-1DP31	Profibus-DP-V0/V1-Slave - ECO	3-56

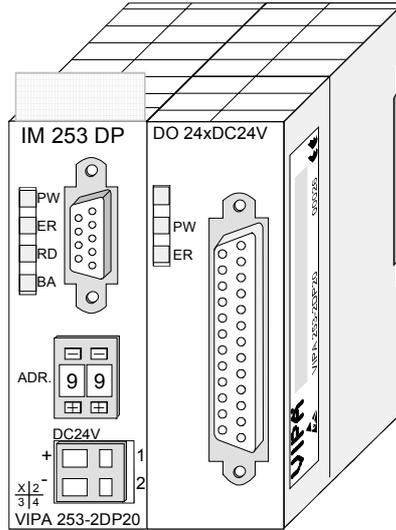
**Profibus-DP-Slaves  
LWL-Interface**



Bestelldaten

Typ	Bestellnummer	Beschreibung	Seite
IM 253DPO	VIPA 253-1DP10	Profibus-DP-V0-Slave mit LWL-Interface	3-35
IM 253DPO	VIPA 253-1DP11	Profibus-DP-V0/V1-Slave mit LWL-Interface	3-56

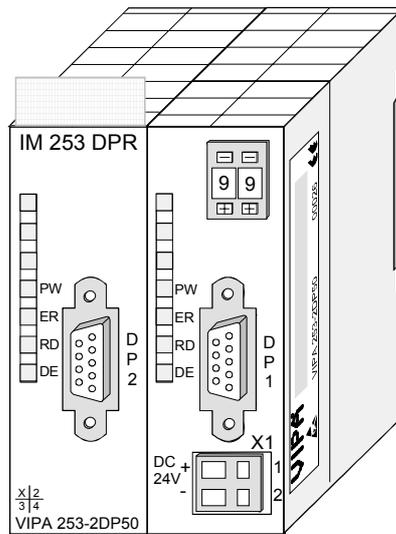
**Profibus-DP-Slave  
(Kombimodule)**



Bestelldaten

Typ	Bestellnummer	Beschreibung	Seite
IM 253DP DO 24xDC24V	VIPA 253-2DP20	Profibus-DP-V0-Slave mit 24fach DO	3-38

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



Bestelldaten

Typ	Bestellnummer	Beschreibung	Seite
IM 253DPR	VIPA 253-2DP50	Profibus-DP-V0-Slave 2 Kanal redundant	3-42

## 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. Seit 1999 ist PROFIBUS zusammen mit weiteren Feldbus-Systemen in der **IEC 61158** standardisiert. Die *IEC 61158* trägt den Titel "Digital data communication for measurement and control - Fieldbus for use in industrial control systems".

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

### Profibus DP-V0

Profibus-DP-V0 (Decentralized Peripherals) stellt die Grundfunktionalitäten von DP zur Verfügung. Dazu gehören der zyklische Datenaustausch sowie die stations-, modul- und kanalspezifische Diagnose.

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. DP steht für einfachen, schnellen, zyklischen Prozessdatenaustausch zwischen einem Busmaster und den zugeordneten Slave-Geräten.

### Profibus DP-V1

Die mit DP-V0 bezeichnete Funktionsstufe wurde um einen azyklischen Datenaustausch zwischen Master und Slave in der Stufe DP-V1 erweitert.

*DP-V1* enthält Ergänzungen mit Ausrichtung auf die Prozessautomatisierung, vor allem den azyklischen Datenverkehr für Parametrierung, Bedienung, Beobachtung und Alarmbearbeitung intelligenter Feldgeräte, parallel zum zyklischen Nutzdatenverkehr. Das erlaubt den Online-Zugriff auf Busteilnehmer über Engineering Tools. Weiterhin enthält DP-V1 Alarme. Dazu gehören unter anderem der Statusalarm, Update-Alarm und ein herstellerspezifischer Alarm.

Wenn Sie die DP-V1-Funktionalität verwenden möchten, ist darauf zu achten, dass Ihr DP-Master ebenfalls DP-V1 unterstützt. Näheres hierzu finden Sie in der Dokumentation zu Ihrem DP-Master.

**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.

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.

**Master Klasse 1  
MSAC\_C1**

Beim Master der Klasse 1 handelt es sich um eine zentrale Steuerung, die in einem festgelegten Nachrichtenzyklus Informationen mit den dezentralen Stationen (Slaves) zyklisch austauscht. Typische MSAC\_C1-Geräte sind Steuerungen (SPS) oder PCs. MSAC\_C1-Geräte verfügen über einen aktiven Buszugriff, mit welchem sie zu festen Zeitpunkten die Messdaten (Eingänge) der Feldgeräte lesen und die Sollwerte (Ausgänge) der Aktuatoren schreiben können.

**Master Klasse 2  
MSAC\_C2**

MSAC\_C2 werden zur Wartung und Diagnose eingesetzt. Hier können angebundenen Geräte konfiguriert, Messwerte und Parameter ausgewertet sowie Gerätezustände abgefragt werden. MSAC\_C2-Geräte müssen nicht permanent am Bussystem angeschlossen sein. Auch verfügen diese über einen aktiven Buszugriff.

Typische MSAC\_C2-Geräte sind Engineering-, Projektierungs- oder Bediengeräte.

<b>Kommunikation</b>	Das Busübertragungsprotokoll bietet zwei Verfahren für den Buszugriff:
<b>Master mit Master</b>	<p>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.</p> <p>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.</p>
<b>Master-Slave-Verfahren</b>	<p>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.</p> <p>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.</p> <p>Zusätzlich zum Nutzdatentransfer den der Master selbständig durchführt, können Sie neue Parametrierdaten an einen Bus-Koppler schicken.</p> <p>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.</p>
<b>Datenkonsistenz</b>	<p>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.</p> <p>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.</p> <p>Für Daten, deren Länge ein Byte überschreiten, wie z.B. bei Analogwerten muss die Datenkonsistenz erweitert werden.</p> <p>VIPA Profibus-DP-Master garantieren ab Firmware-Version V3.00 eine Konsistenz über die erforderliche Länge.</p>

**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**

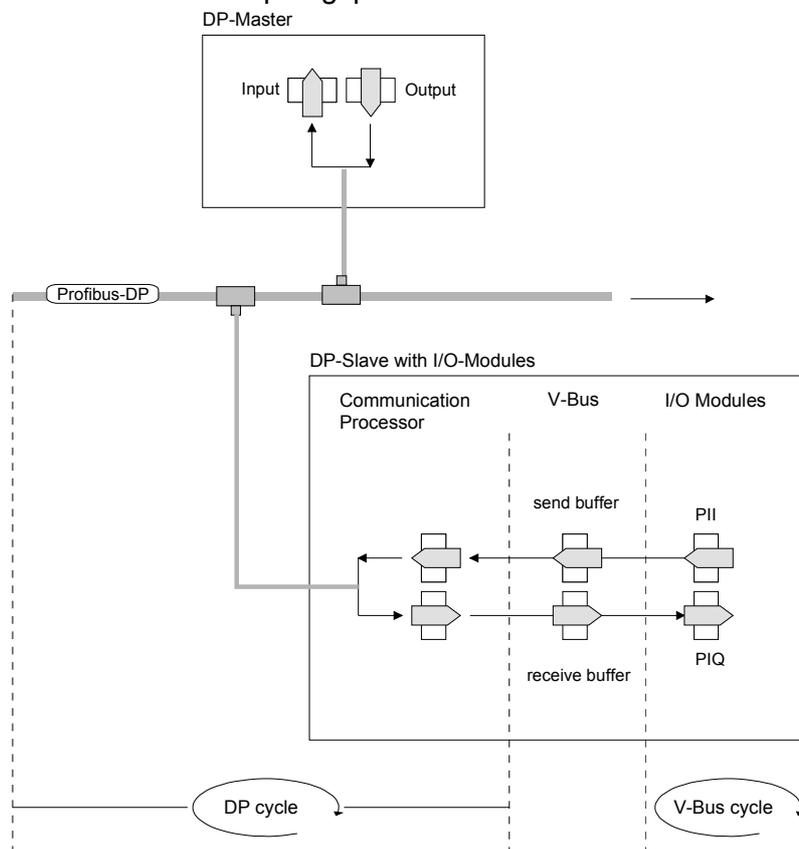
Die umfangreichen Diagnosefunktionen unter Profibus-DP ermöglichen eine schnelle Fehlerlokalisierung. Die Diagnosedaten werden über den Bus übertragen und beim Master zusammengefasst.

Als weitere Funktion wurde bei DP-V1 die gerätebezogene Diagnose verfeinert und in die Kategorien Alarme und Statusmeldungen aufgliedert.

**Funktionsweise der zyklischen Datenübertragung (DP-V0)**

DP-V0 stellt die Grundfunktionalitäten von DP zur Verfügung. Dazu gehören der zyklische Datenaustausch sowie die stations-, modul- und kanalspezifische Diagnose.

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



PII: Prozessabbild der Eingänge  
 PIQ: 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 Empfangspuffer (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 (VIPA\_0550.gsd) 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.

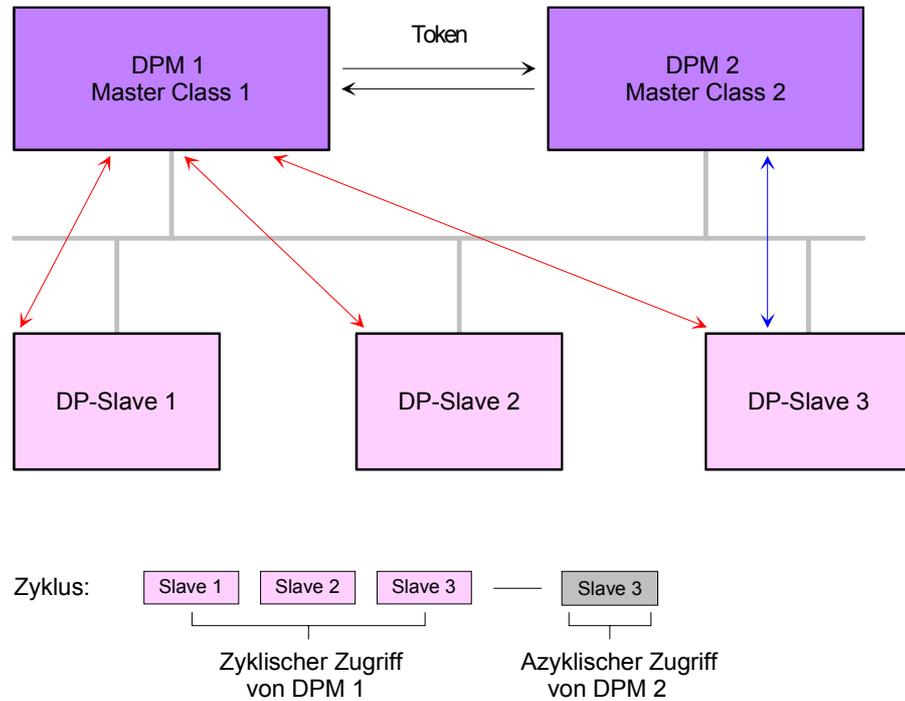
**Hinweis!**

Ab Ausgabestand 6 erlischt bei einem DP-V0-Slave die RUN-LED, sobald der V-Bus-Zyklus länger dauert als der DP-Zyklus. Diese Funktion ist bei Einsatz eines DP-V1-Slaves als DP-V0 deaktiviert.

**Funktionsweise der azyklischen Datenübertragung (DP-V1)**

Der Schwerpunkt der Leistungsstufe von DP-V1 liegt auf dem hier zusätzlich verfügbaren azyklischen Datenverkehr. Dieser bildet die Voraussetzung für Parametrierung und Kalibrierung von Feldgeräten über den Bus während des laufenden Betriebes und für die Einführung bestehender Alarmmeldungen.

Die Übertragung der azyklischen Daten erfolgt parallel zum zyklischen Datenverkehr, allerdings mit niedrigerer Priorität.



In der oben gezeigten Abbildung besitzt der DPM 1 (Master Class 1) die Sendeberechtigung (den Token) und korrespondiert per Aufforderung und Antwort mit Slave 1, danach mit Slave 2 usw. in fester Reihenfolge bis zum letzten Slave der aktuellen Liste (MSO-Kanal); danach übergibt er den Token an den DPM 2 (Master Class 2). Dieser kann in der noch verfügbaren Restzeit ("Lücke") des programmierten Zyklus eine azyklische Verbindung zu einem beliebigen Slave (z.B. Slave 3) zum Austausch von Datensätzen aufnehmen (MS2-Kanal); am Ende der laufenden Zykluszeit gibt er den Token an den DPM 1 zurück.

Der azyklische Austausch von Datensätzen kann sich über mehrere Zyklen bzw. deren "Lücken" hinziehen. Am Ende nutzt der DPM 2 wiederum eine Lücke zum Abbau der Verbindung. Neben dem DPM 2 kann in ähnlicher Weise auch der DPM 1 azyklischen Datenaustausch mit Slaves durchführen (MS1-Kanal).

**Dienste  
azyklischer  
Datenverkehr**

Nachfolgend sind die Dienste für den azyklischen Datenverkehr aufgeführt. Nähere Informationen zu den Diensten und zu den DP-V0/1-Kommunikationsprinzipien finden Sie in der Profibus-Norm IEC 61158.

## DPM 1 (MSAC-C1)

Dienste für azyklischen Datenverkehr zwischen DPM 1 und Slaves	
Read	Der Master liest einen Datenblock beim Slave.
Write	Der Master schreibt einen Datenblock beim Slave.
Alarm	Ein Alarm wird vom Slave zum Master übertragen und von diesem explizit bestätigt. Erst nach Erhalt dieser Bestätigung kann der Slave eine neue Alarmmeldung senden; dadurch ist ein Überschreiben von Alarmen verhindert.
Alarm_Acknowledge	Der Master bestätigt den Erhalt einer Alarmmeldung an den Slave.
Status	Eine Statusmeldung wird vom Slave zum Master übertragen. Es erfolgt keine Bestätigung.
Die Datenübertragung erfolgt verbindungsorientiert über eine MS1-Verbindung. Diese wird vom DPM 1 aufgebaut und ist sehr eng an die Verbindung für den zyklischen Datenverkehr gekoppelt. Sie kann nur von demjenigen Master benutzt werden, der den jeweiligen Slave auch parametrisiert und konfiguriert hat.	

## DPM 2 (MSAC-C2)

Dienste für azyklischen Datenverkehr zwischen DPM 2 und Slaves	
Initiate / Abort	Aufbau bzw. Abbau einer Verbindung für azyklischen Datenverkehr zwischen dem DPM 2 und dem Slave
Read	Der Master liest einen Datenblock beim Slave.
Write	Der Master schreibt einen Datenblock beim Slave.
Data_Transport	Der Master kann anwenderspezifische Daten (in Profilen festgelegt) azyklisch an den Slave schreiben und bei Bedarf im selben Zyklus auch Daten vom Slave lesen.
Die Datenübertragung erfolgt verbindungsorientiert über eine MS2-Verbindung. Diese wird vom DPM 2 vor Beginn des azyklischen Datenverkehrs mit dem Dienst Initiate aufgebaut. Dadurch ist die Verbindung für die Dienste Read, Write und Data_Transport nutzbar. Der Aufbau der Verbindung erfolgt entsprechend. Ein Slave kann mehrere aktive MS2-Verbindungen zeitgleich unterhalten. Eine Begrenzung ist durch die im Slave verfügbaren Ressourcen gegeben.	

---

**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 "Montage und 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 1 und 126 liegen. An den VIPA Profibus-Kopplern stellen Sie die Adresse mit dem Adressierungsschalter an der Front ein.

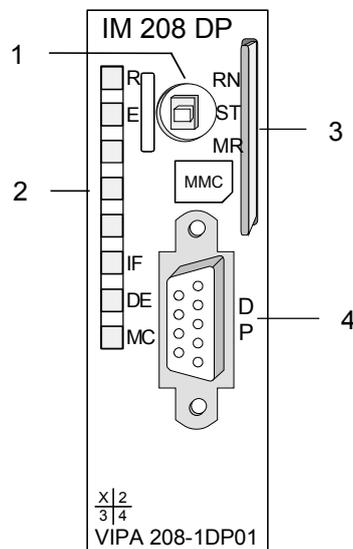
Dem VIPA Profibus-Master müssen Sie die Adresse bei der Projektierung zuteilen.

## IM 208DP - Master - Aufbau

### Eigenschaften

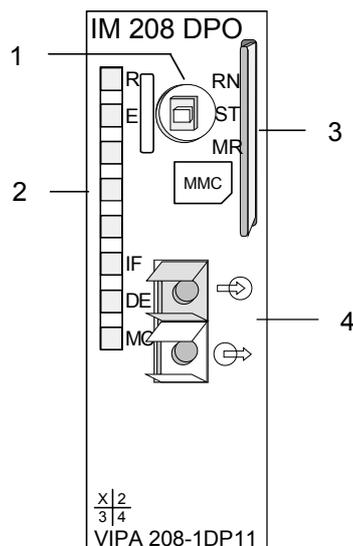
- Profibus-DP-Master der Klasse 1
- 125 DP-Slaves (16 bei DPO) an einen DP-Master ankoppelbar
- Blendet Datenbereich der Slaves über den V-Bus im Adressbereich der CPU ein
- Projektierung unter WinNCS von VIPA bzw. Siemens SIMATIC Manager und ComProfibus von Siemens
- Diagnosefähig

### Frontansicht IM 208DP



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

### Frontansicht IM 208DPO



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

**Komponenten**

**LEDs**

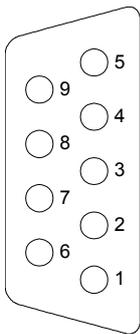
Die Master-Module besitzen verschiedene LEDs, die der Busdiagnose dienen und den eigenen Betriebszustand anzeigen. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

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	Grün	DE (DataExchange) zeigt an, dass eine Kommunikation über Profibus stattfindet. Beim DP-Master mit der Best.-Nr. 208-1DP01 ist diese LED gelb.
MC	Gelb	Zeigt Bereitschaft für Datentransfer von einer MMC an.

**RS485-Schnittstelle (bei IM 208DP)**

Der Profibus-Master IM 208DP wird über eine 9polige Buchse in das Profibus-System eingebunden.

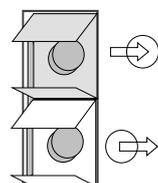
Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung



Pin	Belegung
1	Schirm
2	n.c.
3	RxD/TxD-P (Leitung B)
4	RTS
5	M5V
6	P5V
7	n.c.
8	RxD/TxD-N (Leitung A)
9	n.c.

**LWL-Schnittstelle (bei IM 208DPO)**

Die Einbindung des optischen IM 208DPO erfolgt über die integrierte LWL-Schnittstelle. Die Belegung dieser Schnittstelle zeigt die Abbildung links.



Schließen Sie hier die Empfangs-Leitung (Rückleitung) an

Schließen Sie hier die Sende-Leitung (Sendeleitung) an

<b>Spannungsversorgung</b>	Der Profibus-Master bezieht seine Spannungsversorgung über den Rückwandbus.
<b>Betriebsart-Schalter</b>	<p>Mit dem Betriebsart-Schalter können Sie zwischen den Betriebsarten STOP (ST), RUN (RN) und Master Reset (MR) wählen.</p> <p>Bei Betriebsartschalter auf RN und gültigen Parametern geht der Master in den RUN-Zustand über.</p> <p>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.</p> <p>Eine ausführliche Erklärung zu den Übergängen zwischen RUN und STOP finden Sie in diesem Kapitel unter "Betriebszustände".</p> <p>Mit der Tasterstellung MR können Sie folgendes auslösen:</p> <ul style="list-style-type: none"><li>• Datentransfer von MMC in das Flash-ROM</li><li>• Serieller Modus zum Einsatz des Green Cable von VIPA</li><li>• Urlösches des DP-Masters</li></ul> <p>Näheres zu diesen Möglichkeiten finden Sie in diesem Kapitel weiter unten.</p>
<b>MMC als externes Speichermedium</b>	<p>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.</p> <p>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".</p>

**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. Eine schon vorhandene Hardware-Konfiguration wird hierbei nicht gelöscht.

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: 2, Übertragungsrate: 1,5MBaud.**

**Hinweis!**

Bei DP-Master Firmware-Versionen älter V 5.0.0 wird bei einem STOP-RUN-Übergang am DP-Master eine schon vorhandenen Hardware-Konfiguration durch den Bootvorgang im DP-Master gelöscht und ein eventuell im Flash-ROM vorhandenes Projekt verwendet.

Zur erneuten Übertragung einer Hardware-Konfiguration ist an der CPU ein Power ON durchzuführen.

*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 - Einsatz an einer CPU 21x

**Kommunikation** Über die IM 208 DP-Master-Module können bis zu 125 Profibus-DP-Slaves (bis zu 16 bei DPO) 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.

# IM 208DP - Master - Projektierung

## Übersicht

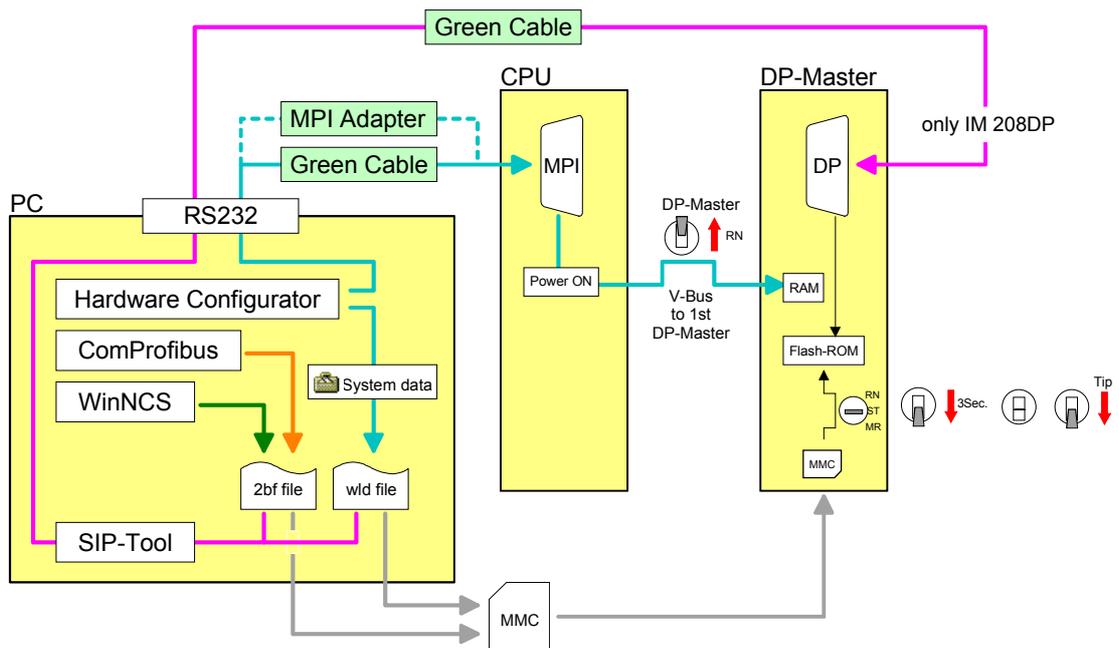
Zur Projektierung haben Sie folgende Möglichkeiten:

- Projektierung des 1. DP-Master im System (CPU 21xDPM, IM 208)

Projektierung im Hardware-Konfigurator von Siemens und Transfer über die Systemdaten in die CPU. Beim Hochlauf der CPU erhält der Master sein Projekt von der CPU.
- Projektierung weiterer DP-Master im System (nur IM 208)

Projektierung im Hardware-Konfigurator von Siemens und Export als wld-Datei. Transfer der Datei über MMC bzw. SIP-Tool und Green Cable in den Master. Mit einer Urlösch-Sequenz am DP-Master wird das Projekt in das Flash-ROM des DP-Master übertragen.
- Projektierung unter WinNCS bzw. ComProfibus

Projektierung unter VIPA WinNCS bzw. unter ComProfibus von Siemens und Export als 2bf-Datei. Transfer der Datei über MMC bzw. Green Cable in den Master. Mit einer Urlösch-Sequenz am DP-Master wird das Projekt in das Flash-ROM des DP-Master übertragen.



## Erforderliche Firmwarestände

DP-Master und CPU sollten 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
V3.0.6	-----	Urlöschen des DP-Master

**Projektierung des 1. DP-Master im System**

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.

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 315-2AF03-0AB0 V1.2)*
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:

Steckplatz	Baugruppe	Bestellnummer	Firmw...	M...	E...	A...	K...
1							
2	CPU 315-2 DP	6ES7 315-2AF03-0AB0	V1.2	2			
X2	DP				7023		
3							
4							
5							
6							
7							
8							
9							
10							



**Hinweis!**

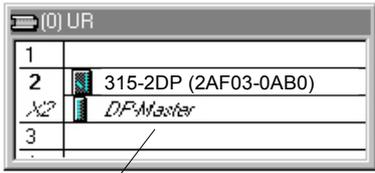
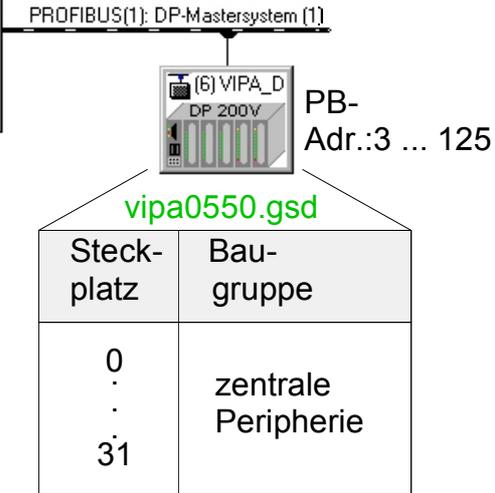
Bei DP-Master Firmware älter als V 5.0.0 muss sich der Betriebsartenschalter des DP-Master in Stellung RN befinden. Ansonsten bootet der Master beim STOP-RUN-Übergang neu und die Projektierung wird gelöscht.

6. Um zum Siemens SIMATIC Manager kompatibel zu sein, ist die System 200V CPU explizit einzubinden.  
 Hängen Sie hierzu an das Subnetz das System "VIPA\_CPU21x". Sie finden dies im Hardware-Katalog unter:  
*PROFIBUS DP > Weitere Feldgeräte > IO > VIPA\_System\_200V > VIPA\_CPU21x.*  
 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\_CPU21x* 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 (Platzhalter).
8. Zur Projektierung von DP-Slaves, die an den DP-Master angekoppelt sind entnehmen Sie aus dem Hardware-Katalog das entsprechende Profibus-System wie beispielsweise *VIPA\_DP200V\_2* und ziehen Sie dies auf das DP-Master Subnetz.  
 Vergeben Sie dem Slave eine Adresse > 2.  
 Platzieren Sie die entsprechenden Module ab Steckplatz 0, indem Sie diese unter *VIPA\_DP200V\_2* dem Hardware-Katalog entnehmen.  
**CPU 21x zentral** vipa\_21x.gsd vipa0550.gsd
- 
- The diagram illustrates a Profibus DP system configuration. On the left, a screenshot of the SIMATIC Manager HW Config shows a DP-Master configuration with a 315-2DP (2AF03-0AB0) module at rack position 2. Below this, a table shows the rack configuration for the central CPU 21x:
- | Steckplatz | Baugruppe                         |
|------------|-----------------------------------|
| 0          | 21x-xxxx                          |
| 1          | zentrale Peripherie mit DP-Master |
| .          |                                   |
| .          |                                   |
| 32         |                                   |
- On the right, a table shows the rack configuration for decentralized DP slaves:
- | Steckplatz | Baugruppe           |
|------------|---------------------|
| 0          | zentrale Peripherie |
| .          |                     |
| .          |                     |
| 31         |                     |
9. Klicken Sie auf (speichern und übersetzen).

Wie Sie Ihr Projekt über MPI in die CPU transferieren, finden Sie auf den Folgeseiten unter "Transfervarianten".

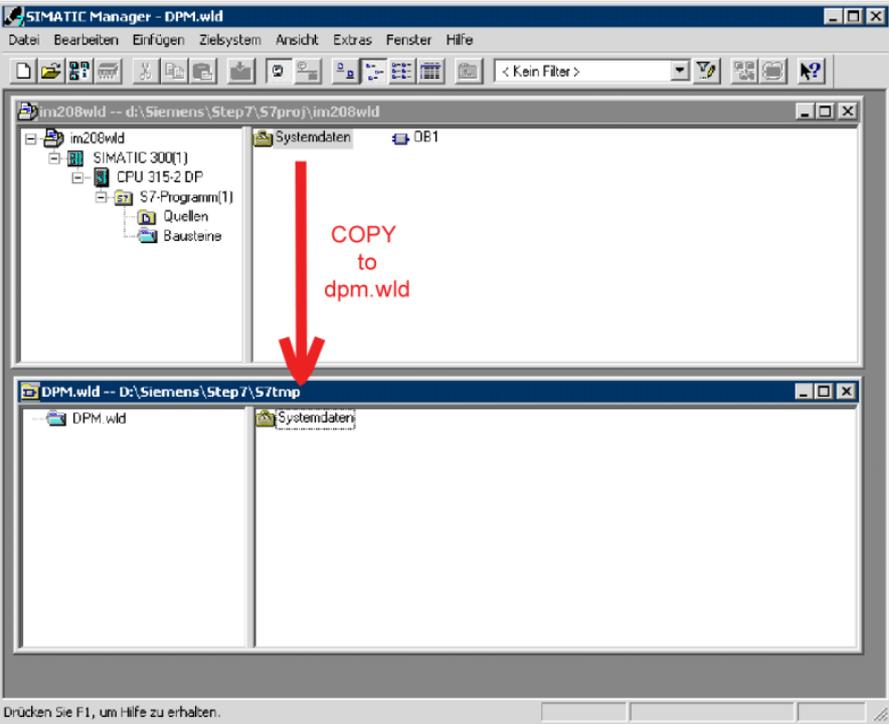
**Projektierung weiterer DP-Master im System**

Befinden sich mehrere IM 208 DP-Master in einem System, so ist für jeden DP-Master ein Projekt zu erstellen. Dieses Projekt ist entweder per MMC oder über Green Cable in den entsprechenden DP-Master zu übertragen. Mit einer Urlösch-Sequenz am DP-Master wird das Projekt dauerhaft im Flash-ROM des DP-Master gespeichert.

1.	Legen Sie für den entsprechenden DP-Master 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: <i>Simatic300&gt;CPU-300&gt;CPU315-2DP (6ES7 315-2AF03-0AB0 V1.2)</i>								
3.	Geben Sie Ihrem Master eine Profibus-Adresse von 2 aufsteigend.								
4.	Klicken Sie auf DP und stellen Sie unter <i>Objekteigenschaften</i> 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.								
6.	Zur Projektierung von DP-Slaves, die an den DP-Master ange-koppelt sind, entnehmen Sie aus dem Hardware-Katalog beispie-lsweise das entsprechende Profibus-System <i>VIPA_DP200V_2</i> und ziehen Sie dies auf das DP-Master Subnetz. Vergeben Sie dem Slave eine Adresse > 2. Platzieren Sie die entsprechenden Module ab Steckplatz 0, indem Sie diese unter <i>VIPA_DP200V_2</i> dem Hardware-Katalog ent-nehmen. <b>CPU 21x zentral</b>  PB-Adr.:2 <b>DP-Slaves dezentral ...</b>  <table border="1" data-bbox="949 1518 1324 1787"> <thead> <tr> <th>Steck-platz</th> <th>Bau-gruppe</th> </tr> </thead> <tbody> <tr> <td>0</td> <td rowspan="3">zentrale Peripherie</td> </tr> <tr> <td>·</td> </tr> <tr> <td>·</td> </tr> <tr> <td>31</td> <td></td> </tr> </tbody> </table> <i>vipa0550.gsd</i>	Steck-platz	Bau-gruppe	0	zentrale Peripherie	·	·	31	
Steck-platz	Bau-gruppe								
0	zentrale Peripherie								
·									
·									
31									
7.	Klicken Sie auf  (speichern und übersetzen).								

dpm.wld exportieren Exportieren Sie Ihr Projekt in Form einer wld-Datei auf eine MMC. Die MMC stecken sie in den entsprechenden DP-Master. Mit einer Urlösch-Sequenz am DP-Master können Sie Ihr Projekt von der MMC in das Flash-ROM des DP-Master übertragen. Nach der Übertragung kann die MMC wieder entnommen werden.

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

8.	<p>Legen Sie mit <b>Datei</b> &gt; <i>Memory Card</i> &gt; <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.</p> <p>→ Die Datei wird zusätzlich zum Projektfenster eingeblendet.</p>
9.	<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>  <p>Soll ein schon bestehendes "Systemdaten"-Verzeichnis überschrieben werden, so ist zuvor das zu überschreibende Verzeichnis "Systemdaten" zu löschen.</p>

Wie Sie Ihre dpm.wld-Datei in den entsprechenden DP-Master transferieren können, finden Sie auf den Folgeseiten unter "Transfervarianten".

### Projektierung unter WinNCS bzw. ComProfibus

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 (nur bei IM 208DP möglich).

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.	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. <b>Bitte beachten Sie bitte, dass es bei der automatischen Adressierung nicht zu Adresskonflikten mit der lokalen Peripherie kommt!</b> Handelt es sich um ein intelligentes Modul, wie z.B. CP240, erscheinen die hierzu einstellbaren Parameter.
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.	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. Die Transfervarianten sind auf den Folgeseiten beschrieben.



#### 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.

**Transfervarianten**

Für den Transfer der wld- bzw. 2bf-Datei in Ihren DP-Master haben Sie folgende Möglichkeiten:

- Transfer über MPI in die CPU (nur 1. DP-Master im System)
- Transfer über MMC
- Transfer über Green Cable und SIP-Tool

**Transfer über MPI in die CPU**

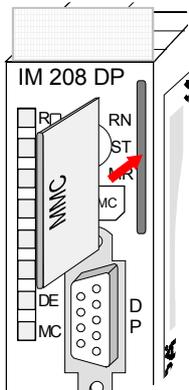
Ab Firmware-Version V 3.0.6 für den DP-Master und V 3.3.0 für die CPU 21x können Sie nach folgender Vorgehensweise über MPI Ihr Projekt in die CPU übertragen. Die CPU reicht bei Power ON die DP-Master-Projektierung automatisch weiter an den 1. DP-Master (IM 208DP oder CPU 21xDPM), der sich am Systembus befindet.

1.	<p>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 bei geeigneten Modulen der VIPA eingesetzt werden. Bitte beachten Sie hierzu die Hinweise zum Green Cable weiter unten!</p> <p>Bei Einsatz des Green Cable von VIPA ist die MPI-Schnittstelle entsprechend zu konfigurieren (PC Adapter MPI, 38400Baud).</p>
2.	Schalten Sie Ihren DP-Master in RUN.
3.	Schalten Sie die Spannungsversorgung Ihrer CPU ein.
4.	<p>Übertragen Sie vom Hardware-Konfigurator von Siemens über den Menüpunkt <b>Zielsystem</b> &gt; <i>Laden in Baugruppe</i> Ihr Projekt in die CPU.</p> <p>Immer mit Power ON bekommt der 1. DP-Master sein Projekt von der CPU.</p> <p>Zur zusätzlichen Sicherung Ihres Projekts auf MMC stecken Sie eine MMC in die CPU und übertragen Sie mit Zielsystem &gt; 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.</p>

**Hinweis!**

Bei DP-Master Firmware älter als V 5.0.0 muss sich der Betriebsartenschalter des DP-Master in Stellung RN befinden. Ansonsten bootet der Master beim STOP-RUN-Übergang neu und die Projektierung wird gelöscht.

**Transfer über MMC**



1.	Übertragen Sie mit einem MMC-Lesegerät Ihre wld- bzw. 2bf-Datei auf eine MMC.
2.	Stecken Sie das MMC-Speichermodul in Ihren IM 208DP-Master.
3.	Schalten Sie die Spannungsversorgung Ihres System 200V ein.
4.	Drücken Sie den Betriebsartenschalter des Master-Moduls in Stellung MR. Halten Sie diesen gedrückt, bis die blinkende MC-LED in Dauerlicht übergeht.
5.	<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>
6.	Die MMC können Sie jetzt entnehmen.
7.	Schalten Sie den Master von STOP in RUN. → Der IM 208DP-Master startet nun mit dem neuen Projekt im internen Flash-ROM. Die RUN-LED (R) und DE leuchtet.



**Hinweis!**

Bitte beachten Sie, dass nur für den 1. Master ein in der CPU befindliches Profibus-Projekt immer vorrangig gegenüber einem Projekt im Flash-ROM des DP-Master behandelt wird.

Bei einer wld- und 2bf-Datei auf einer MMC wird die wld-Datei vorrangig behandelt.

### Transfer über Green Cable und VIPA SIP-Tool

Die hier gezeigte Methode können Sie ausschließlich beim IM 208DP mit RS485-Schnittstelle anwenden. Das SIP-Tool ist ein Transfertools. 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 ein Programmier- und Downloadkabel für VIPA CPUs mit MP<sup>2</sup>I-Buchse 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.
- *Firmware-Updates der CPUs und Feldbus-Master durchführen*  
Über das Green Cable können Sie unter Einsatz eines Upload-Programms die Firmware aller aktuellen VIPA CPUs mit MP<sup>2</sup>I-Buchse 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). Beispielsweise ist vor dem Stecken des Green Cable ein gestecktes MPI-Kabel zu entfernen.

Zurzeit unterstützen folgende Komponenten das Green Cable:

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

#### Hinweis zur Verlängerung

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

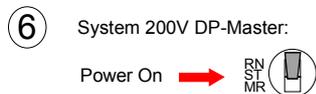
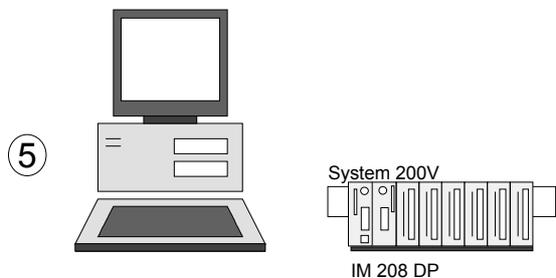
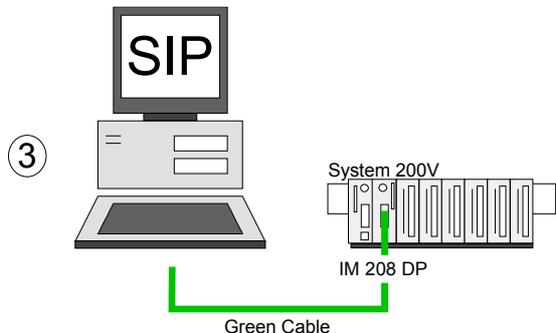
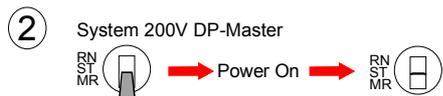
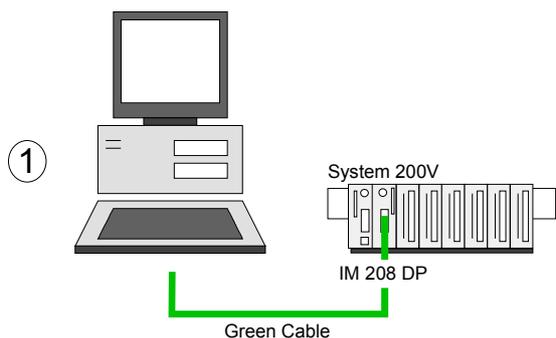
Das Green Cable darf nur mit einem 1:1 Kabel (alle 9 Pin 1:1 verbunden) verlängert werden.

**Fortsetzung  
Transfer Green  
Cable und VIPA  
SIP-Tool**

Falls Sie den IM 208 Profibus-DP-Master mit dem SIP-Tool projektieren möchten, ist dies ab der DP-Master-Firmware V 4.0.0 und mit dem SIP-Tool ab V 1.0.6 möglich.

Mit SIP-Tool können Sie entweder eine wld- oder einer 2bf-Datei in den DP-Master übertragen. Wie schon weiter oben beschrieben, enthält die wld-Datei die exportierten Systemdaten aus dem Siemens SIMATIC Manager.

Mit WinNCS (siehe Folgeseite) von VIPA bzw. über ComProfibus von Siemens können Sie Ihr Projekt als 2bf-Datei exportieren.



1.	Ziehen Sie den Profibus-Stecker von Ihrem DP-Master. Stecken Sie das Green Cable auf die serielle Schnittstelle Ihres PC und auf die Profibus-Schnittstelle des DP-Masters.
2.	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.
3.	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.
4.	Klicken Sie auf [Download], wählen Sie Ihre dpm.2bf- bzw. dpm.wld-Datei und übertragen Sie diese in den DP-Master.
5.	Beenden Sie nach der Datenübertragung die Verbindung und das SIP-Tool. Ziehen Sie das Green Cable vom Master ab.
6.	Schalten Sie die Spannungsversorgung für Ihren Master aus. Verbinden Sie den Master mit dem Profibus-Netz und schalten Sie die Spannungsversorgung wieder ein. Bringen Sie Ihren Master in RUN (RN). → Ihr IM 208DP-Profibus-Master befindet sich nun mit den aktuellen Projektierdaten am Netz. Die Daten sind im internen Flash-ROM abgelegt.

## IM 208DP - Master - Slave-Betrieb

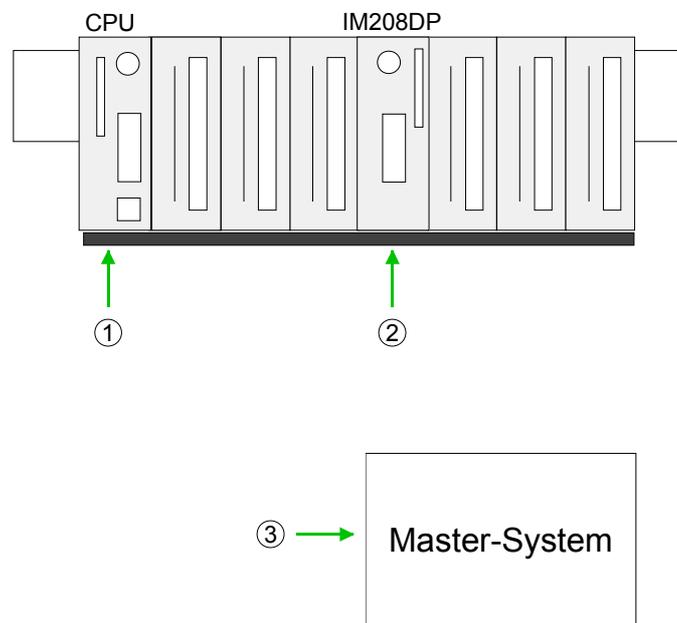
### Übersicht

Ab der CPU-Firmware 3.72 haben Sie die Möglichkeit den IM 208DP als DP-Slave zu betreiben. Für die Anbindung an einen DP-Master ist die Siemens-GSD für die Siemens CPU S7-315-2DP erforderlich.

Aus hardwaretechnischen Gründen ist diese Funktionalität für den IM 208DPO Master mit LWL nicht verfügbar.

Für den Einsatz des IM 208DP als DP-Slave sind folgende 3 Hardware-Konfigurationen durchzuführen, die nachfolgend näher erläutert sind:

### Slave-System



#### 1. Hardware-Konfiguration System 200V

Projektierung Siemens CPU 315-2DP mit virtuellem Profibus-Slave (Adresse 1) für das System 200V. Der DP-Slave beinhaltet CPU 21x, I/O-Peripherie und IM 208DP (Parameter *Projekt an IM 208 übertragen* auf "Nein"). Das Projekt ist in CPU 21x zu übertragen.

#### 2. Hardware-Konfiguration IM 208DP

Projektierung IM 208DP als Siemens CPU 315-2DP mit DP-Teil im Slave-Betrieb. Über die *Eigenschaften* Profibus-Adresse und E/A-Bereich bestimmen und Projekt in IM 208DP übertragen.

#### 3. Hardware-Konfiguration übergeordnetes Master-System

Projektierung des übergeordneten Master-Systems. Die Anbindung des IM 208DP (Slave) erfolgt als Siemens CPU S7-315-2DP. Hierzu ist die Einbindung der Siemens-GSD erforderlich. Über die *Eigenschaften* Profibus-Adresse (identisch mit Hardware-Konfiguration IM 208DP) und E/A-Bereich in Form von "Modulen" angeben. Das Projekt ist in die CPU des Master-Systems zu übertragen.

**Hardware-Konfiguration System 200V**

- Starten Sie den Siemens SIMATIC Manager.
- Installieren Sie zur Projektierung der CPU 21x die GSD *VIPA\_21x.GSD*.
- Installieren Sie zur Anbindung an Ihren DP-Master die GSD *VIPA04D5.GSD*.
- Legen Sie ein neues Projekt System 300 an und fügen Sie aus dem Hardwarekatalog eine Profilschiene ein.
- Fügen Sie die CPU 315-2DP ein. Hardwarekatalog:  
*Simatic300 > CPU-300 > CPU315-2DP (6ES7 315-2AF03-0AB0 V1.2)*
- Legen Sie ein neues Profibus-Subnetz an und geben Sie Ihrem Master eine Profibus-Adresse von 2 aufsteigend.
- Hängen Sie an das Subnetz das System "VIPA\_CPU21x". Sie finden dies im Hardware-Katalog unter *PROFIBUS DP > Weitere Feldgeräte > IO > VIPA\_System\_200V > VIPA\_CPU21x*.
- 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!**
- Zur Einbindung der am VIPA-Bus befindlichen Module ziehen Sie aus dem Hardware-Katalog unter *VIPA\_CPU21x* die entsprechenden System 200V Module auf die Steckplätze unterhalb der CPU. Beginnen Sie mit Steckplatz 1. Auf diese Weise platzieren Sie auch Ihren IM 208DP (Platzhalter).
- Stellen Sie in den Eigenschaften des IM 208DP den Parameter *Projekt an IM 208 übertragen* auf "Nein". Hierdurch wird verhindert, dass die lokal im IM 208DP abgelegte DP-Slave-Projektierung von einem in der CPU befindlichen Projekt überschrieben wird.
- Übertragen Sie Ihr Projekt in die CPU.

**Hardware-Konfiguration System 100V**

Der Einsatz des IM 208DP als DP-Slave in einem System 100V kann ausschließlich über die Systemerweiterung erfolgen. Näheres zur Montage finden Sie im HB100 unter "Erweiterungs- und Klemmen-Module".

Hierbei erfolgt die Hardware-Konfiguration auf die gleiche Weise wie bei dem System 200V unter Einsatz folgender GSD-Dateien für das System 100V:

- Zur Projektierung der CPU 11x die GSD *VIPA\_11x.GSD*
- Zur Anbindung an DP-Master die Siemens-GSD

Stellen Sie auch hier in den Eigenschaften des IM 208DP den Parameter *Projekt an IM 208 übertragen* auf "Nein". Übertragen Sie Ihr Projekt in die CPU 11x.

Fahren Sie wie nachfolgend gezeigt mit der Hardware-Konfiguration des IM 208DP und des übergeordneten Master-Systems fort.

### Hardware-Konfiguration IM 208DP

- Legen Sie ein neues Projekt System 300 an und fügen Sie aus dem Hardwarekatalog eine Profilschiene ein.
- Fügen Sie die CPU 315-2DP ein. Hardwarekatalog:  
*Simatic300 > CPU-300 > CPU315-2DP (6ES7 315-2AF03-0AB0 V1.2)*
- Rufen Sie die *Objekteigenschaften* von DP auf.
- Stellen Sie unter Betriebsart "DP-Slave" ein.
- Vergeben Sie unter *Allgemein* für den DP-Slave eine Profibus-Adresse.
- Legen Sie über *Konfiguration* die Bereiche für den Datentransfer an. Bitte beachten Sie, dass hierbei ausschließlich der "MS"-Mode unterstützt wird.
- Übertragen Sie wie weiter oben unter "Transfervarianten" gezeigt die Systemdaten in Ihren IM 208DP - nicht in die CPU! - und bringen Sie diesen in RUN.



#### Hinweis!

Die Angabe für "Eingang" bzw. "Ausgang" unter *Konfiguration* erfolgt immer aus Sicht der CPU. "Eingang" bezieht sich auf den Eingabe- und "Ausgang" auf den Ausgabe-Bereich der CPU.

### Hardware-Konfiguration übergeordnetes Master-System

Zur Projektierung in einem übergeordneten Master-System ist die Einbindung der Siemens-GSD erforderlich.

- Starten Sie Ihr Projektierwerkzeug mit einem neuen Projekt und projektieren Sie Ihr Profibus-DP-Master, der dem DP-Slave übergeordnet ist.
- Fügen Sie an das Mastersystem das DP-Slave-System "S7-315-2DP" an. Nach der Einbindung der Siemens-GSD finden Sie das DP-Slave-System im Hardware-Katalog unter:  
*Profibus-DP > Weitere Feldgeräte > SPS > SIMATIC > S7-315-2DP*
- Vergeben Sie für den DP-Slave die gleiche Profibus-Adresse, die Sie in der *Hardware-Konfiguration IM 208DP* vergeben haben.
- Legen Sie in Form von "Modulen" für die Profibus-Kommunikation den gleichen E/A-Bereich an, den Sie auf der Slave-Seite parametrieren haben. Beachten Sie dass sich ein Slave-Ausgabe- auf einen Master-Eingabe-Bereich bezieht und umgekehrt. Auch müssen die E/A-Bereiche durchgängig ohne Lücken projektiert sein.
- Speichern Sie Ihr Projekt und übertragen Sie dieses in die CPU Ihres Master-Systems



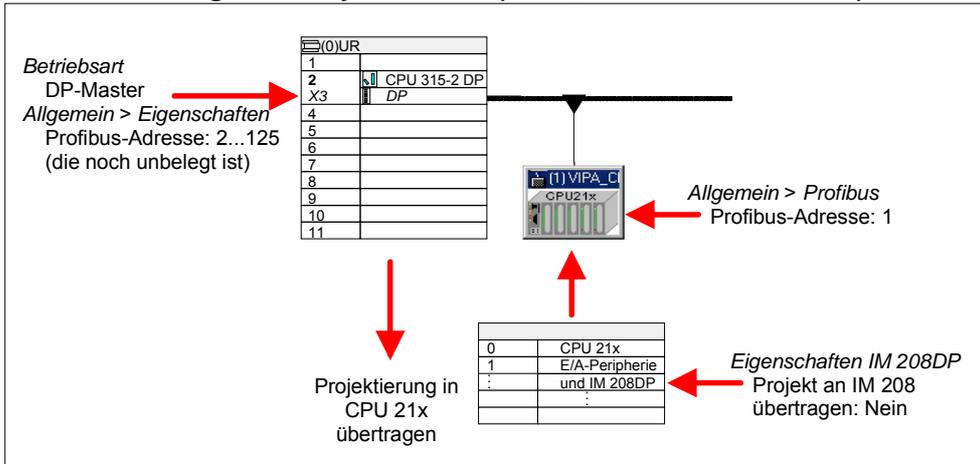
#### Hinweis!

Sollte es sich bei Ihrem DP-Master-System um ein System 200V von VIPA handeln, so können Sie durch Anbindung eines "DP200V"-Slave-Systems die direkt gesteckten Module projektieren. Damit dieses Projekt von der VIPA-CPU als zentrales System erkannt wird, müssen Sie dem "DP200V"-Slave-System die Profibus-Adresse 1 zuweisen!

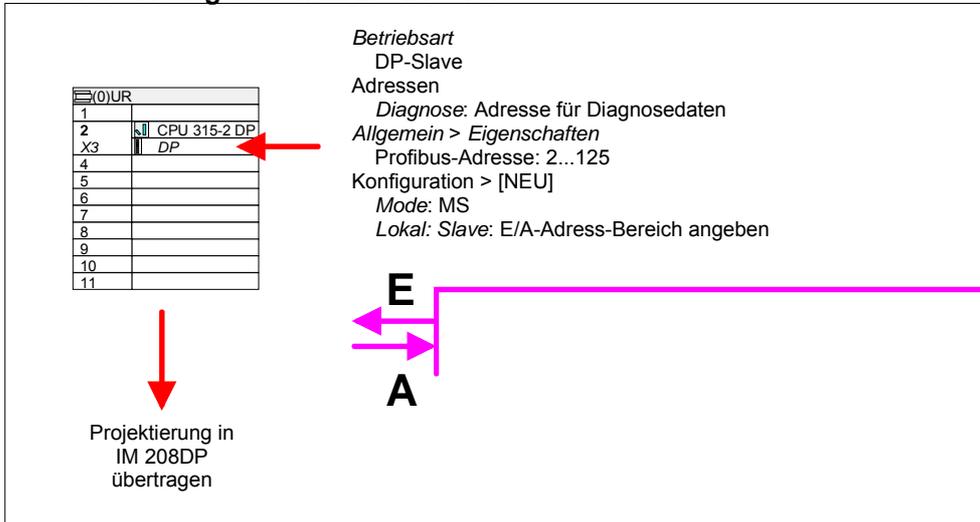
**Bitte beachten Sie bei Einsatz des IM 208 Profibus-DP-Master, dass dieser einen Firmwarestand ab V 3.0 besitzt; ansonsten kann dieser an der CPU 21x mit Firmwarestand ab V 3.0 nicht betrieben werden. Die Firmwarestände entnehmen Sie bitte dem Aufkleber, der sich auf der Rückseite des jeweiligen Moduls befindet.**

**Zusammenfassung**

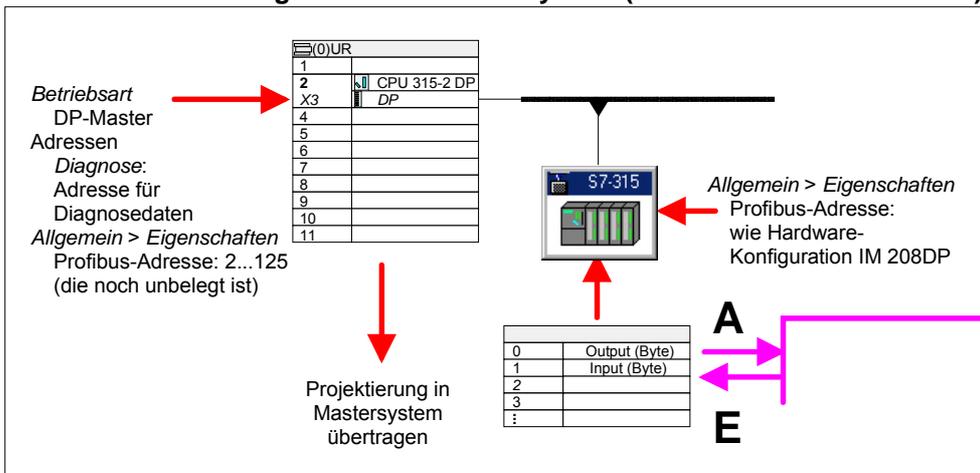
**Hardware-Konfiguration System 200V (VIPA\_21x.GSD erforderlich)**



**Hardware-Konfiguration IM 208DP Slave**



**Hardware-Konf. übergeordnetes Master-System (Siemens-GSD erforderlich)**



**Achtung!**

Die Längenangaben für Ein- und Ausgabe-Bereich im DP-Slave müssen mit den Byteangaben bei der Master-Projektierung übereinstimmen. Ansonsten kann keine Profibus-Kommunikation stattfinden und der Master meldet Slave-Ausfall!

## IM 208DP - Master - Urlöschen

### Allgemeines

Ab der Firmware-Version V 3.0.6 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;"> <p>Das Diagramm zeigt die zeitliche Abfolge der LED-Aktivitäten während des Urlöschvorgangs. Es besteht aus mehreren Spalten, die jeweils eine LED-Anzeige (R, E, IF, DE, MC) darstellen. Ein Schalter (RN, ST, MR) wird in der Stellung MR gedrückt. Ein roter Pfeil zeigt auf den Schalter mit der Beschriftung '9s'. Danach folgen die LED-Aktivitäten: 3x MC (blinkend), 3s MC (Dauerlicht), 3x IF (blinkend), 3s IF (Dauerlicht). Ein weiterer roter Pfeil zeigt auf den Schalter mit der Beschriftung 'max. 3s' und 'Tip'. Danach folgen 3x R (blinkend) und 3s IF (Dauerlicht). Ein grüner Pfeil zeigt von 'Clear' zu 'OK'.</p> </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: 2, Übertragungsrate: 1,5MBAud</b></p>

### Projektierung über CPU nach Netz-Ein an 1. DP-Master

Sollte ein Profibus-Projekt in der CPU vorliegen, so wird dieses automatisch nach einem Netz-Ein über den Rückwandbus in das RAM des 1. DP-Master übertragen - unabhängig von der Stellung des Betriebsartenschalters des Masters.

## IM 208DP - Master - Firmwareupdate

### Übersicht

Ab der CPU-Firmware-Version 3.3.3 haben Sie die Möglichkeit mittels einer MMC über die CPU ein Firmwareupdate unter anderem auch für den DP-Master durchzuführen. Die 2 aktuellsten Firmwarestände finden Sie auf [www.vipa.de](http://www.vipa.de) im Service-Bereich und auf dem VIPA-ftp-Server unter [ftp.vipa.de](ftp://vipa.de).

Hierbei gibt es zur Kennzeichnung einer DP-Master Firmware folgende Namenskonventionen:

**dpmxx.bin** mit xx geben Sie die Nummer des DP-Master Steckplatzes an (00 ... 31)



### Achtung!

Beim Aufspielen einer neuen Firmware ist äußerste Vorsicht geboten. Unter Umständen kann Ihr DP-Master unbrauchbar werden, wenn beispielsweise während der Übertragung die Spannungsversorgung unterbrochen wird oder die Firmware-Datei fehlerhaft ist.

Setzen Sie sich in diesem Fall mit der VIPA-Hotline in Verbindung!

### Firmware-Version ermitteln

Den ausgelieferten Firmwarestand können Sie einem Aufkleber entnehmen, der sich auf der Rückseite des DP-Master-Moduls befindet.

### Firmware laden und als dpmxx.bin auf MMC übertragen

- Gehen Sie auf [www.vipa.de](http://www.vipa.de).
- Klicken Sie auf Service > Download > Firmware Updates.
- Klicken Sie auf "Firmware für Profibus Master System 200V".
- Wählen Sie die entsprechende IM 208 Bestell-Nr. aus und laden Sie die Firmware auf Ihren PC.
- Benennen Sie die Datei um in "**dpmxx.bin**" (xx entspricht dem DP-Master-Steckplatz beginnend mit 00) und übertragen Sie diese Datei auf eine MMC.



### Hinweis!

Auf dem Server sind immer die 2 aktuellsten Firmware-Versionen abgelegt.

**Voraussetzungen für ftp-Zugriff**

Zur Ansicht von ftp-Seiten in Ihrem Web-Browser sind ggf. folgende Einstellungen vorzunehmen:

*Internet Explorer*

ftp-Zugriff nur möglich ab Version 5.5

**Extras** > *Internetoptionen*, Register "Erweitert" im Bereich Browsing:

- aktivieren: "Ordneransicht für ftp-Sites aktivieren"
- aktivieren: "Passives ftp verwenden ..."

*Netscape*

ftp-Zugriff nur möglich ab Version 6.0

Es sind keine zusätzlichen Einstellungen erforderlich

Sollte es immer noch Probleme mit dem ftp-Zugriff geben, fragen Sie Ihren Systemverwalter.

**Firmware von MMC in DP-Master übertragen**

- Bringen Sie den RUN-STOP-Schalter Ihrer CPU in Stellung STOP.
- Schalten Sie die Spannungsversorgung aus.
- Stecken Sie die MMC mit der Firmware in die CPU. Achten Sie hierbei auf die Steckrichtung der MMC.
- Schalten Sie die Spannungsversorgung ein.
- Nach einer kurzen Hochlaufzeit zeigt das abwechselnde Blinken der CPU-LEDs SF und FC an, dass auf der MMC die Firmware-Datei gefunden wurde.
- Sie starten die Übertragung der Firmware, sobald Sie innerhalb von 10s den RUN/STOP-Schalter der CPU kurz nach MRES tippen. Die CPU zeigt die Übertragung über ein LED-Lauflicht an.
- Während des Update-Vorgangs blinken die LEDs SF, FC und MC abwechselnd. Dieser Vorgang kann mehrere Minuten dauern.
- Das Update ist fehlerfrei beendet, wenn alle CPU-LEDs leuchten. Blinken diese schnell, ist ein Fehler aufgetreten.
- Nach einem Power OFF - ON steht Ihnen der Master mit neuer Firmware zur Verfügung.

**Hinweis!**

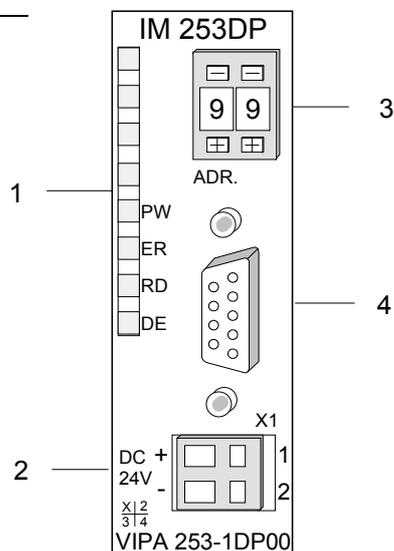
Näheres zum Firmwareupdate finden Sie auf [ftp.vipa.de](http://ftp.vipa.de) unter *support*.

## IM 253-1DPx0 - DP-V0-Slave - Aufbau

### Eigenschaften

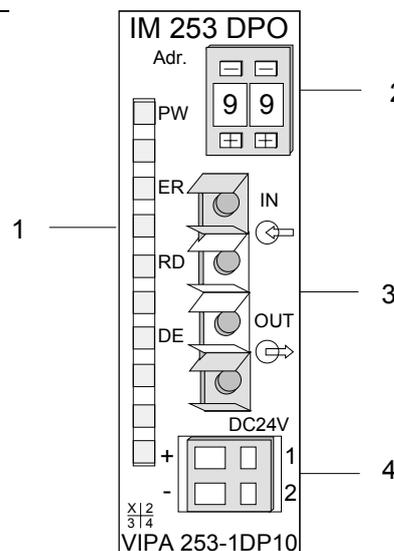
- Profibus (DP-V0)
- Profibus-DP-Slave für max. 32 Peripherie-Module (max. 16 Analog-Module)
- Max. 152Byte Eingabe- und 152Byte Ausgabe-Daten
- Internes Diagnoseprotokoll
- Integriertes DC 24V-Netzteil zur Versorgung der Peripherie-Module (max. 3,5A)
- Unterstützung aller Profibus-Datenraten

### Frontansicht 253-1DP00



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

### Frontansicht 253-1DP10



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

## Komponenten

### LEDs

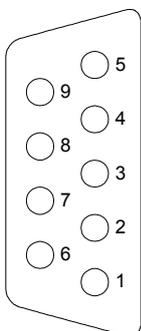
Die Profibus-Slave-Module besitzen verschiedene LEDs, die unter anderem auch der Busdiagnose dienen. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
PW	Grün	Signalisiert eine anliegende Betriebsspannung (Power).
ER	Rot	Leuchtet kurz bei Neustart und dauerhaft bei internem Fehler. Blinkt bei Initialisierungsfehler. Blinkt abwechselnd mit RD bei fehlerhafter Konfiguration vom Master (Projektierungsfehler).
RD	Grün	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. Blinkt abwechselnd mit ER bei fehlerhafter Konfiguration vom Master (Projektierungsfehler).
DE	Grün	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.

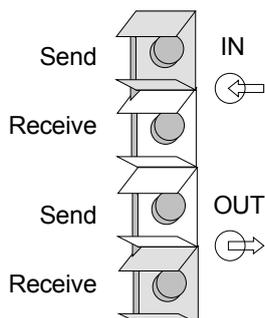
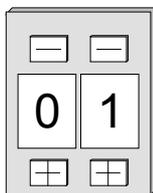
Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:



Pin	Belegung
1	Schirm
2	n.c.
3	RxD/TxD-P (Leitung B)
4	RTS
5	M5V
6	P5V
7	n.c.
8	RxD/TxD-N (Leitung A)
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 253-2DP20 - DP-V0-Slave mit DO 24xDC 24V - Aufbau

### Allgemeines

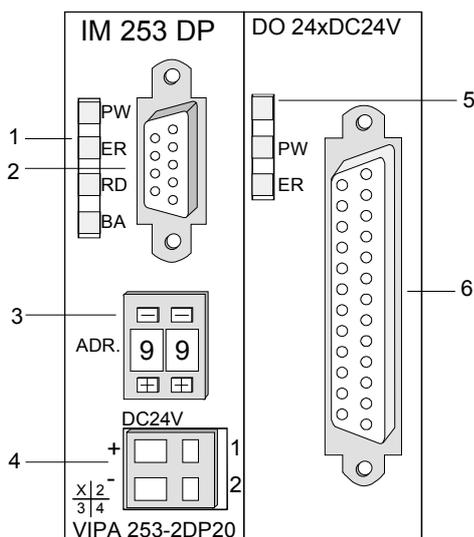
Das Modul besteht aus einem Profibus-Slave mit integrierter 24fach Ausgabe-Einheit. Direkt über den Profibus werden die 24 Ausgabekanäle angesteuert. Ein Ausgabekanal kann mit maximal 1A belastet werden. Hierbei ist zu beachten, dass ein Summenstrom von 4A nicht überschritten wird. Die Ausgänge sind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

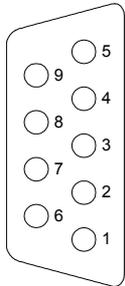
Auf dem digitalen Ausgabe-Teil befinden sich 2 LEDs, die folgende Funktion haben:

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.

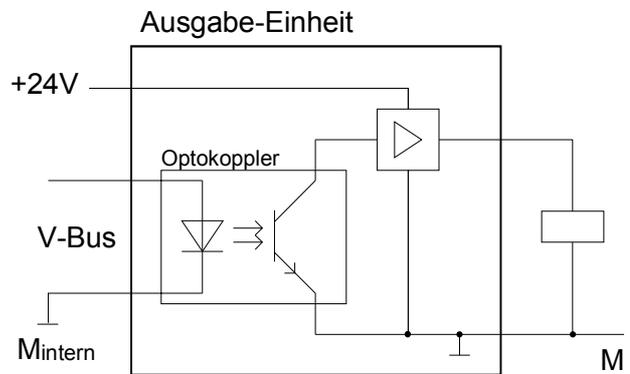
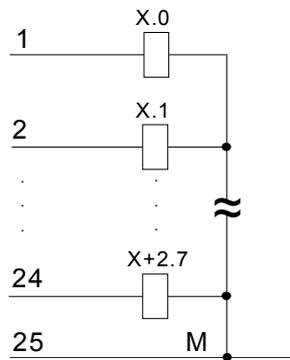
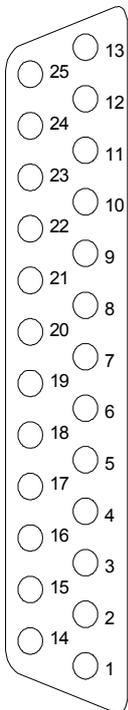
Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:

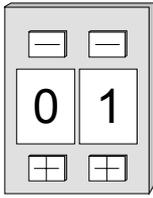


Pin	Belegung
1	Schirm
2	n.c.
3	RxD/TxD-P (Leitung B)
4	RTS
5	M5V
6	P5V
7	n.c.
8	RxD/TxD-N (Leitung A)
9	n.c.

**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.

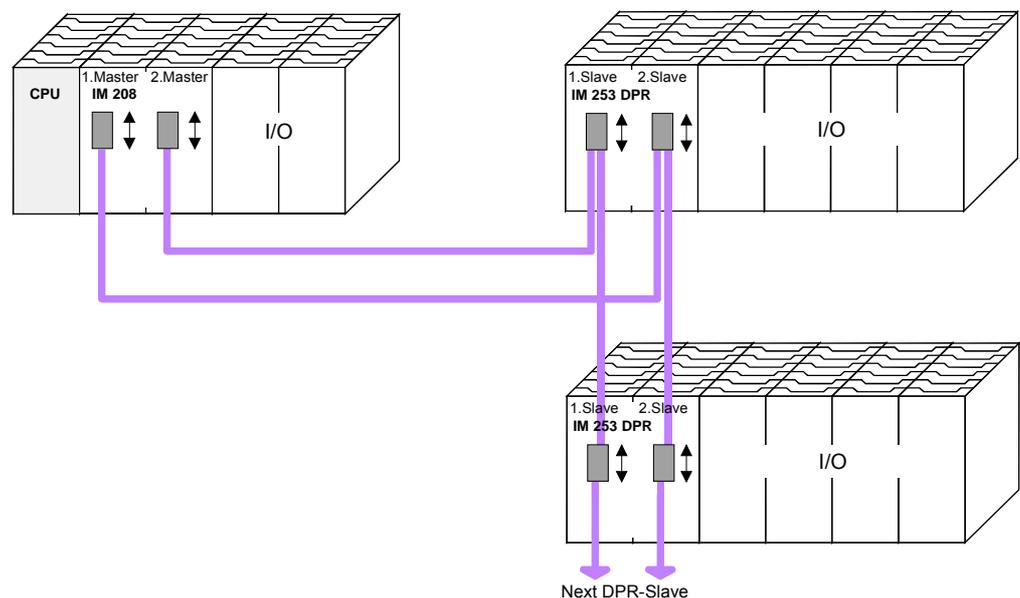
Zur Einbindung der 24 Ausgänge projektieren Sie zusätzlich das Modul VIPA 253-2DP20 auf dem ersten Steckplatz. Das Modul befindet sich hardwaretechnisch gesehen direkt neben dem Slave.

## IM 253-2DP50 - DP-V0-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.



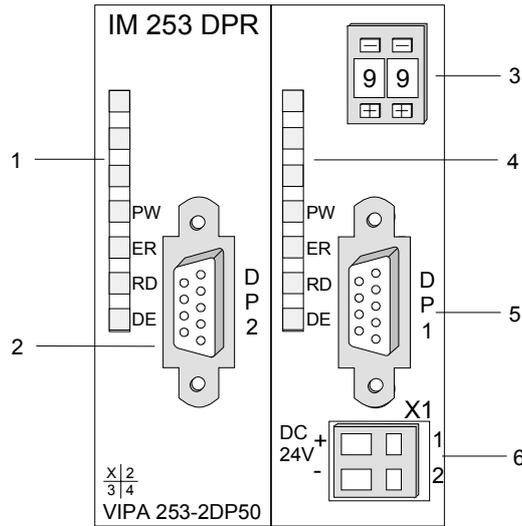
### Voraussetzungen für den Einsatz

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 253DPR

- 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**

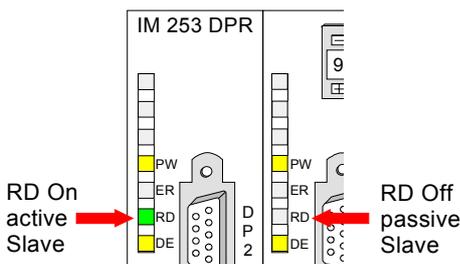
**LEDs**

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	Grün	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	Grün	DE (DataExchange) zeigt Kommunikation mit dem Profibus an.

**LEDs bei  
redundantem  
Betrieb**

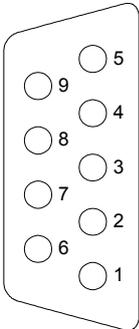
Im redundanten Betrieb zeigt der aktive Slave seine Aktivität über die grüne RD-LED an. Beim passiven Slave ist die RD-LED ausgeschaltet. Bei beiden Slaves leuchtet die PW- und DE-LED.



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

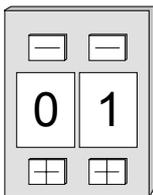
**RS485-Schnittstelle**

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 (Leitung B)
4	RTS
5	M5V
6	P5V
7	n.c.
8	RxD/TxD-N (Leitung A)
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.

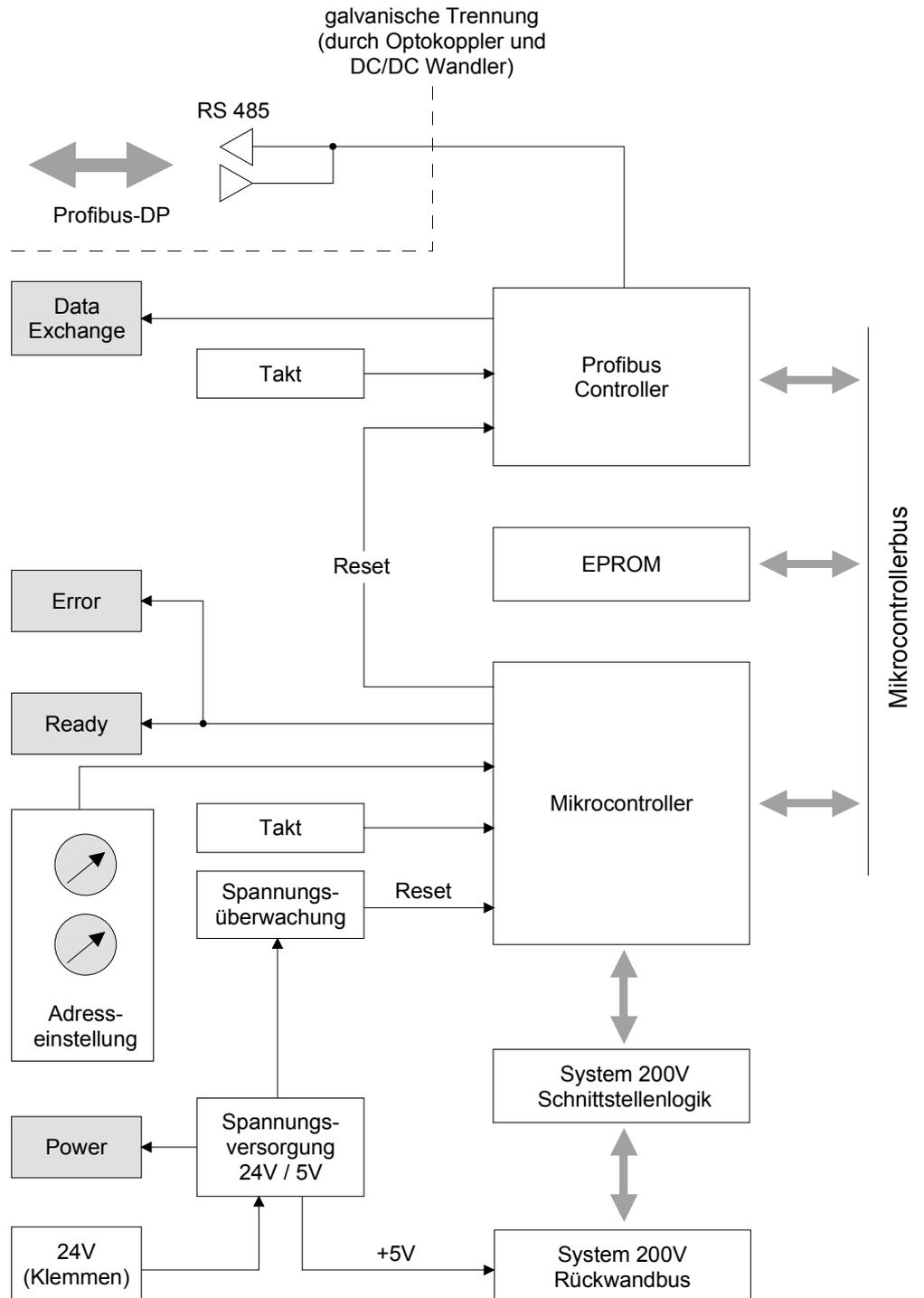


**Achtung!**

Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung!

### IM 253-xDPx0 - DP-V0-Slave - Blockschaltbild

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



## IM 253-xDPx0 - DP-V0-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 einstellen können.

Bei der Hardwarekonfiguration werden die hier vorgestellten Slaves über die GSD-Datei projektiert.

### 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 unter Cx000023\_Vxxx.ZIP. Die Zuordnung der GSD-Datei zu Ihrem DP-Slave entnehmen Sie bitte der "Liesmich.txt"-Datei von Cx000023\_Vxxx.ZIP.

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!

Sie können die GSD-Datei auch über den ftp-server downloaden:

*ftp: //ftp.vipa.de/support/profibus\_gsd\_files*

Nach Installation der GSD finden Sie beispielsweise den DP-V0-Slave im Hardware-Katalog von Siemens unter:

*Profibus-DP>Weitere Feldgeräte>I/O>VIPA\_System\_200V>  
VIPA 253-1DP00*

### 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 projektiert sein, da sich das Modul hardwaretechnisch gesehen direkt neben dem Slave befindet.

### 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.

### Parametrierung im redundanten System

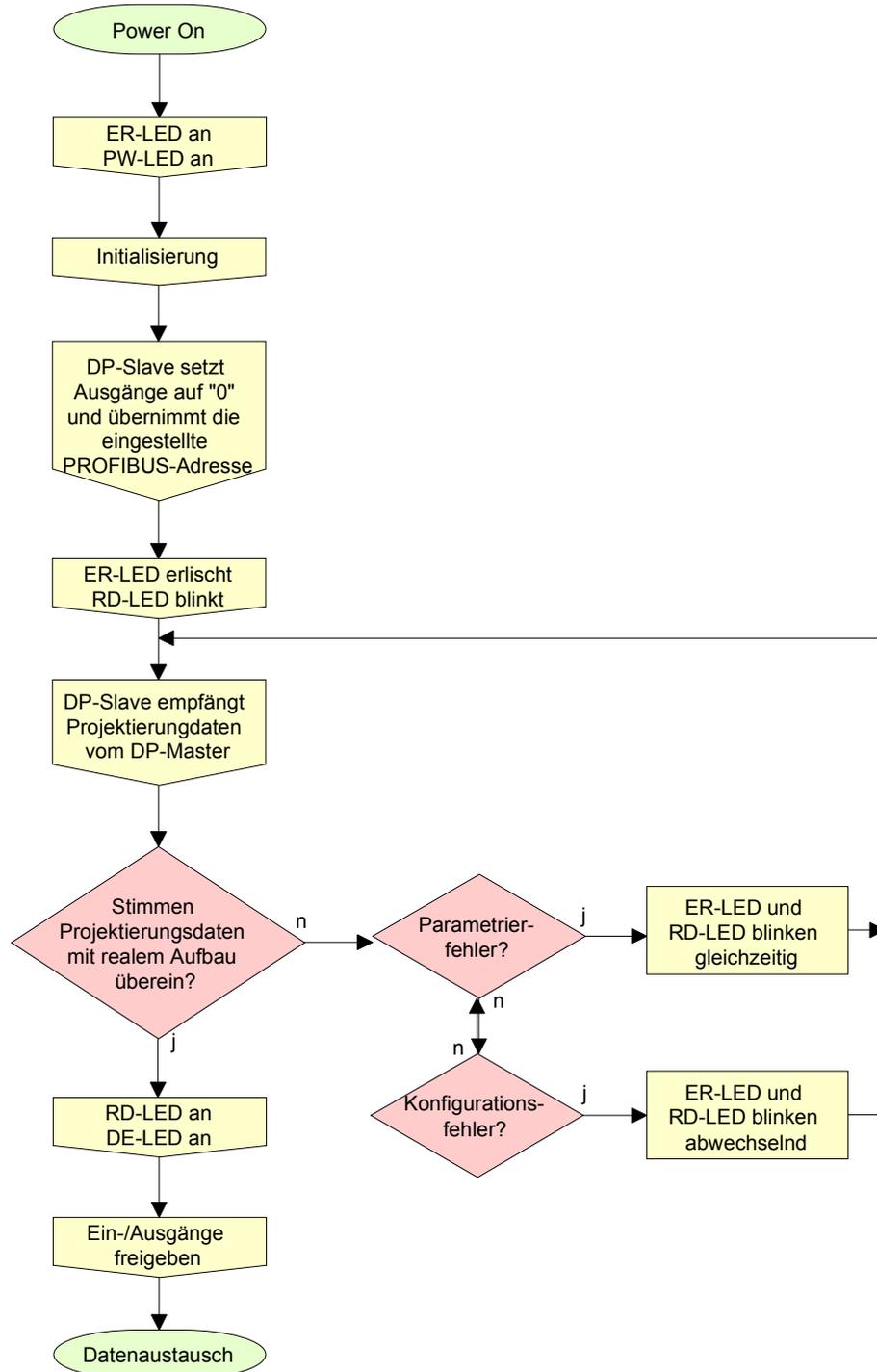
Nur der Slave-Teil, der zuerst in DataExchange geht (systembedingt immer der linke Slave), wird automatisch zum aktiven Slave, hat Zugriff auf die Peripherie-Module und kann diese parametrieren.

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 253-xDPx0 - DP-V0-Slave - Parameter

### Übersicht

Bei Einsatz der in diesem Handbuch aufgeführten DP-V0-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 253-xDPx0 - DP-V0-Slave - Diagnosefunktionen

- Übersicht** Die umfangreichen Diagnosefunktionen von Profibus-DP ermöglichen eine schnelle Fehlerlokalisierung. Die Diagnosemeldungen werden über den Bus übertragen und beim Master zusammengefasst.  
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.  
Setzen Sie sich hierzu bitte mit der VIPA-Hotline in Verbindung!
- Interne Diagnose Systemmeldungen** Das System legt auch Diagnosemeldungen ab wie die Zustände "Ready" bzw. "DataExchange", die nicht an den Master weitergeleitet werden.  
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.
- Diagnosedaten manuell sichern** Über die kurzzeitige Einstellung von 00 am Adress-Schalter können Sie die Diagnose-Daten während des "DataExchange" im Flash-ROM sichern.
- Diagnosemeldung bei Spannungsausfall** 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 Unterspannung/Abschaltung-Diagnosemeldung aus dem Zeitstempel des EEPROMs generiert und im Diagnose-RAM abgelegt.
- Diagnosezusatz des IM 253DPR** 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.  
Dieses virtuelle Status-"Modul" ist ab GSD-Version 1.30 verfügbar.

**Aufbau der DP-V0-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.

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.

## IM 253-xDPx1 - DP-V1-Slave - Aufbau

### Eigenschaften 253-1DP01 253-1DP11

- Profibus (DP-V0, DP-V1)
- Profibus-DP-Slave für max. 32 Peripherie-Module (max. 16 Analog-Module).
- Max. 244Byte Eingabe- und 244Byte Ausgabe-Daten
- Internes Diagnoseprotokoll
- Integriertes DC 24V-Netzteil zur Versorgung der Peripherie-Module mit max. 3,5A.
- Unterstützung aller Profibus-Datenraten

### Einsatz als DP-V1-Slave

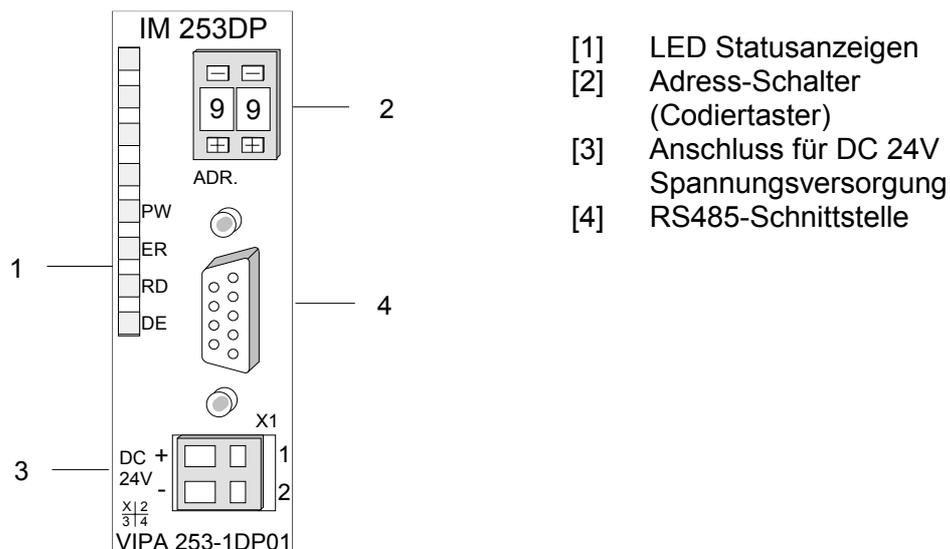
- 1 MSAC\_C1-Verbindung (Read, Write) mit 244Byte Daten (4 Byte DP-V1-Header + 240Byte Nutzdaten)
- 3 MSAC\_C2-Verbindungen (Initiale, Read, Write, DataTransport, Abort) mit jeweils 244Byte Daten (4 Byte DP-V1-Header + 240Byte Nutzdaten)

### Einschränkungen 253-1DP31 - ECO

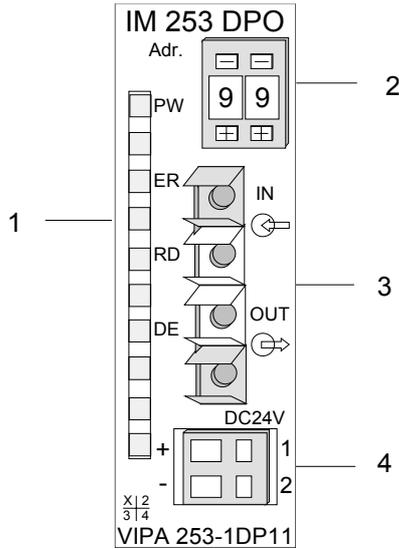
Der IM 253-1DP31 - ECO ist funktional identisch mit dem IM 253-1DP01 und hat folgende Einschränkungen:

- Profibus-DP-Slave für max. 8 Peripherie-Module
- Integriertes DC 24V-Netzteil zur Versorgung der Peripherie-Module mit max. 0,8A.
- Vorgabe der Profibus-Adresse über DIP-Schalter

### Frontansicht 253-1DP01

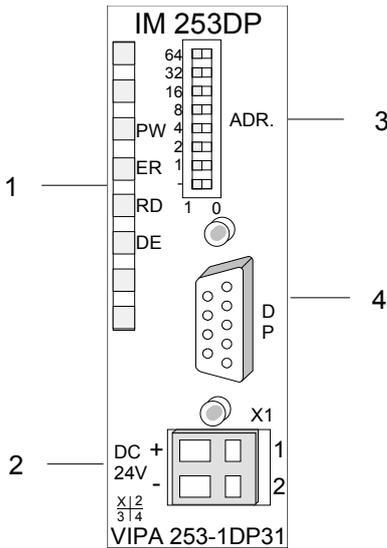


**Frontansicht  
253-1DP11**



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

**Frontansicht  
253-1DP31 - ECO**



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

## Komponenten

### LEDs

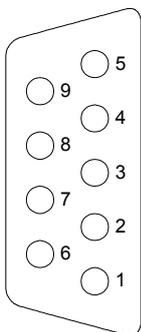
Die Profibus-Slave-Module besitzen verschiedene LEDs, die unter anderem auch der Busdiagnose dienen. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bezeichnung	Farbe	Bedeutung
PW	Grün	Signalisiert eine anliegende Betriebsspannung (Power).
ER	Rot	Leuchtet kurz bei Neustart und dauerhaft bei internem Fehler. Blinkt bei Initialisierungsfehler. Blinkt abwechselnd mit RD bei fehlerhafter Konfiguration vom Master (Projektierungsfehler).
RD	Grün	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. Blinkt abwechselnd mit ER bei fehlerhafter Konfiguration vom Master (Projektierungsfehler).
DE	Grün	Blinkt gleichzeitig mit ER bei fehlerhafter Parametrierung DE (DataExchange) zeigt an, dass eine Kommunikation mit dem Profibus stattfindet.

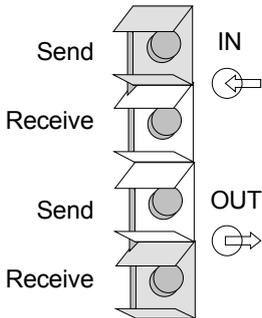
### RS485-Schnittstelle

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

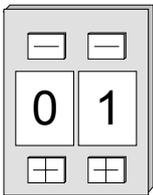
Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:



Pin	Belegung
1	n.c.
2	n.c.
3	RxD/TxD-P (Leitung B)
4	RTS
5	M5V
6	P5V
7	n.c.
8	RxD/TxD-N (Leitung A)
9	n.c.

**LWL-Schnittstelle**

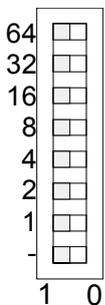
Über diese Buchse binden Sie den Profibus-Koppler über Lichtwellenleiter in den Profibus ein. Die Anschlussbelegung dieser Schnittstelle zeigt die Abbildung links.

**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.

**Adress-Schalter  
IM 253-1DP31 - ECO**

Im Gegensatz zu dem oben beschriebenen Codiertaster besitzt der IM 253-1DP31 - ECO zur Adresseinstellung einen DIL-Schalter. Erlaubte Adressen sind 1 bis 125. 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.

**Spannungs-  
versorgung**

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. Beim IM 253-1DP31 - ECO ist der Strom auf max. 0,8A begrenzt.

Das Netzteil ist gegen Verpolung und Überstrom geschützt.

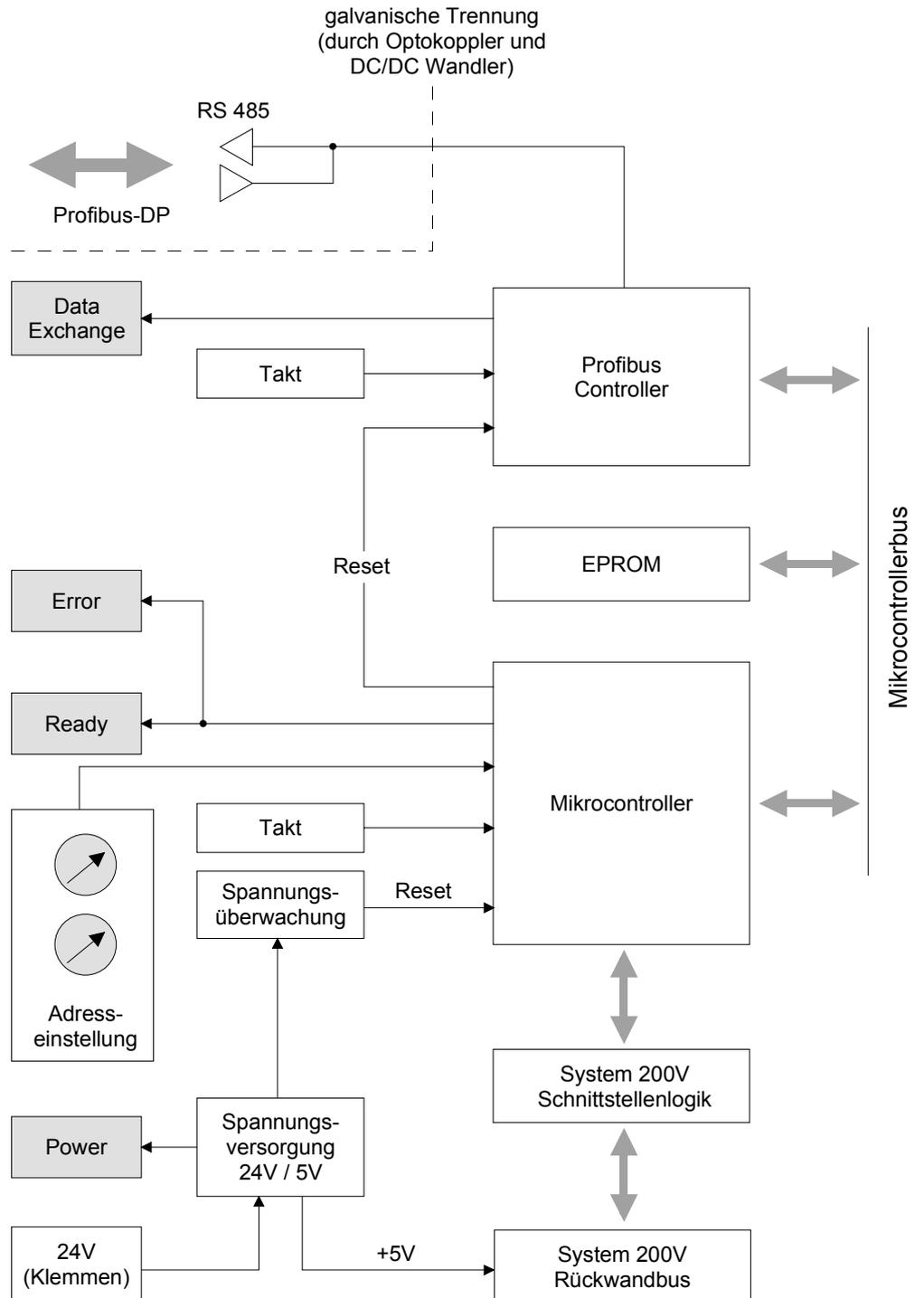
Profibus und Rückwandbus sind galvanisch voneinander getrennt.

**Achtung!**

Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung!

## IM 253-xDPx1 - DP-V1-Slave - Blockschaltbild

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



## IM 253-xDPx1 - DP-V1-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 einstellen können.

Bei der Hardwarekonfiguration werden die hier vorgestellten Slaves über die GSD-Datei projektiert.

### Zuordnung GSD-Datei > DP-Slave

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 unter Cx000023\_Vxxx.ZIP

Die Zuordnung der GSD-Datei zu Ihrem DP-Slave entnehmen Sie bitte der "Liesmich.txt"-Datei von Cx000023\_Vxxx.ZIP.

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.

Sie können die GSD-Datei auch über den ftp-server

*ftp://ftp.vipa.de/support/profibus\_gsd\_files*

downloaden.

Nach Installation der GSD finden Sie beispielsweise den DP-V1-Slave im Hardware-Katalog von Siemens unter:

*Profibus-DP>Weitere Feldgeräte>I/O>VIPA\_System\_200V>  
VIPA 253-1DP01*



### Hinweis!

Bitte verwenden Sie immer für Profibus-DP-Master, die kein DP-V1 unterstützen, die entsprechende GSD für DP-V0.

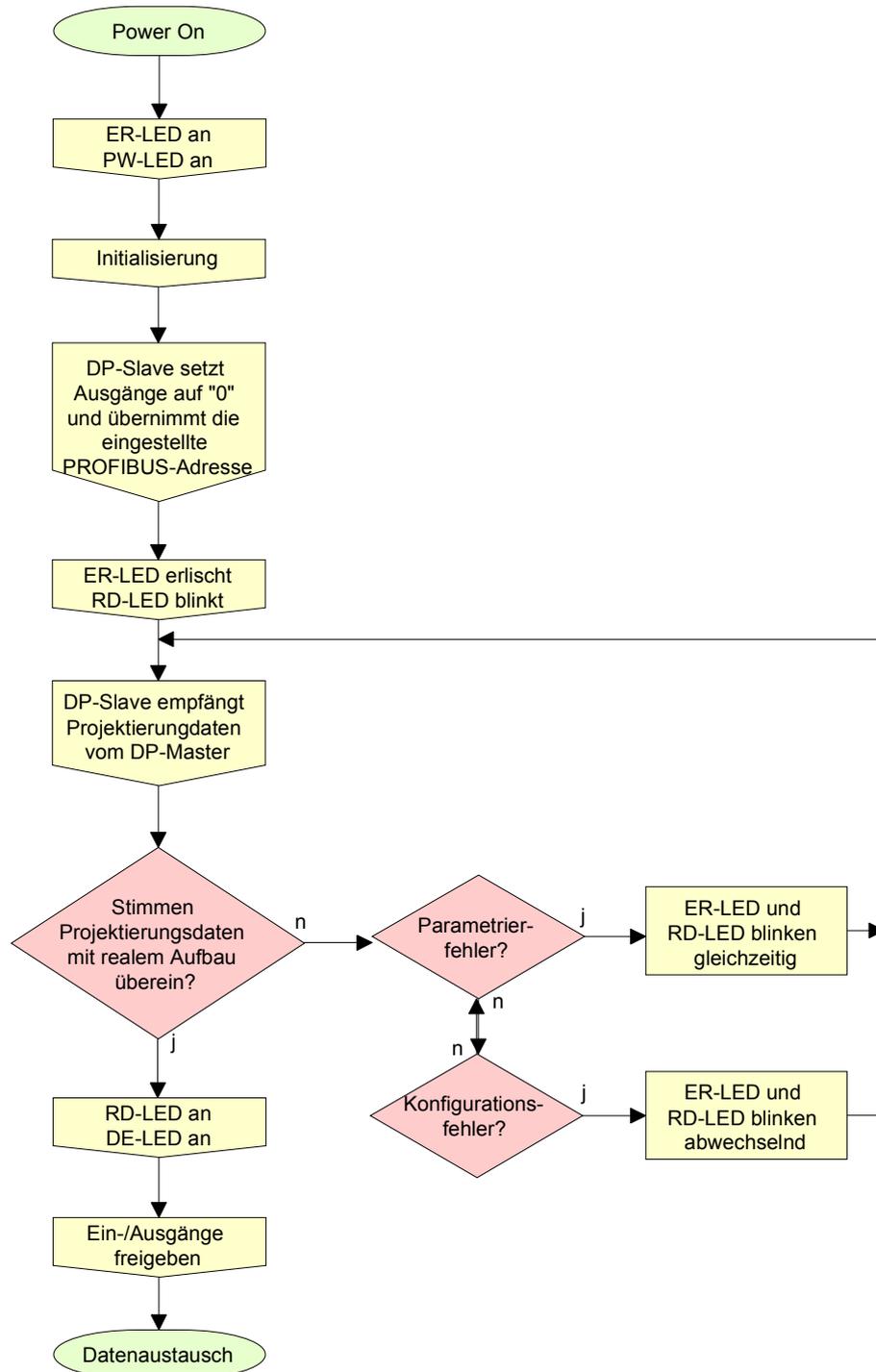
### Einsatz an einem IM 208 DP-Master von VIPA

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

**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 253-xDPx1 - DP-V1-Slave - Parameter

### Übersicht

Die in diesem Kapitel aufgeführten DP-V1-Slaves können durch entsprechende GSD-Wahl auch als DP-V0-Slave eingesetzt werden. Je nach DP-Slave stehen Ihnen dann folgende Parameter zur Verfügung:

### Parameterdaten DP-V0

Bei Verwendung der entsprechenden GSD für DP-V0-Betrieb haben Sie folgende Parameterdaten:

Byte	Bit 7 ... Bit 0	Default
0	Bit 1 ... 0: 0 (fix) Bit 2: 0 = WD-Timebase 10ms 1 = WD-Timebase 1ms Bit 4 ... 3: 0 (fix) Bit 5: 0 = Publisher-Mode wird nicht unterstützt 1 = Publisher-Mode wird unterstützt Bit 7 ... 6: 0 (fix)	00h
1	00h (fix)	00h
2	08h (fix)	08h
3	0Ah (fix)	0Ah
4	81h (fix)	81h
5	00h (fix)	00h
6	00h (fix)	00h
7	Bit 0: 0 = Kennungsbezogene Diagnose freigeben 1 = Kennungsbezogene Diagnose sperren Bit 1: 0 = Modulstatus freigeben 1 = Modulstatus sperren Bit 2: 0 = Kanalbezogene Diagnose freigeben 1 = Kanalbezogene Diagnose sperren Bit 3: 0 (fix) Bit 4: 0 = V0: Herstellerspez. Alarm wird nicht unterstützt 1 = V0: Herstellerspez. Alarm wird unterstützt Bit 5: 0 = V0: Diagnosealarm wird nicht unterstützt 1 = V0: Diagnosealarm wird unterstützt Bit 6: 0 = V0: Prozessalarm wird nicht unterstützt 1 = V0: Prozessalarm wird unterstützt Bit 7: 0 (fix)	70h
8	Bit 6 ... 0: 0 (fix) Bit 7: 0 = Datenformat Motorola 1 = Datenformat Intel (nur bei Analog-Modulen)	00h
9 ... 12	00h (fix)	00h

**DP-V1  
UserPrmData**

Bei Verwendung einer GSD für DP-V1-Betrieb haben Sie folgende Parameterdaten:

Byte	Bit 7 ... Bit 0	Default
0	Bit 1 ... 0: 0 (fix) Bit 2: 0 = WD-Timebase 10ms 1 = WD-Timebase 1ms Bit 4 ... 3: 0 (fix) Bit 5: 0 = Publisher-Mode wird nicht unterstützt 1 = Publisher-Mode wird unterstützt Bit 6: 0 = Fail-Safe-Mode wird nicht unterstützt 1 = Fail-Safe-Mode wird unterstützt Bit 7: 0 = DP-V1-Betrieb sperren 1 = DP-V1-Betrieb freigeben	80h
1	Bit 3 ... 0: 0 (fix) Bit 4: 0 = V1: Herstellerspez. Alarm wird nicht unterstützt 1 = V1: Herstellerspez. Alarm* wird unterstützt Bit 5: 0 = V1: Diagnosealarm wird nicht unterstützt 1 = V1: Diagnosealarm wird unterstützt Bit 6: 0 = V1: Prozessalarm wird nicht unterstützt 1 = V1: Prozessalarm wird unterstützt Bit 7: 0 (fix)	00h
2	08h (fix)	08h
3	0Ah (fix)	0Ah
4	81h (fix)	81h
5	00h (fix)	00h
6	00h (fix)	00h
7	Bit 0: 0 = Kennungsbezogene Diagnose freigeben 1 = Kennungsbezogene Diagnose sperren Bit 1: 0 = Modulstatus freigeben 1 = Modulstatus sperren Bit 2: 0 = Kanalbezogene Diagnose freigeben 1 = Kanalbezogene Diagnose sperren Bit 7 ... 3: 0 (fix)	00h
8	Bit 6 ... 0: 0 (fix) Bit 7: 0 = Datenformat Motorola 1 = Datenformat Intel (nur bei Analog-Modulen)	00h
9 ... 12	00h (fix)	00h

\*) Der IM 253-1DP31 unterstützt keinen Herstellerspezifischen Alarm.

**Datenformat  
Motorola/Intel**

Dieser Parameter wird ausschließlich bei Einsatz von Analog-Modulen ausgewertet und bezieht sich darauf, wie ein Wert im CPU-Adressbereich abgelegt wird.

Im *Motorola-Format* (default) werden die Byte in absteigender Wertigkeit abgelegt d.h. das 1. Byte beinhaltet das High-Byte und das 2. Byte das Low-Byte.

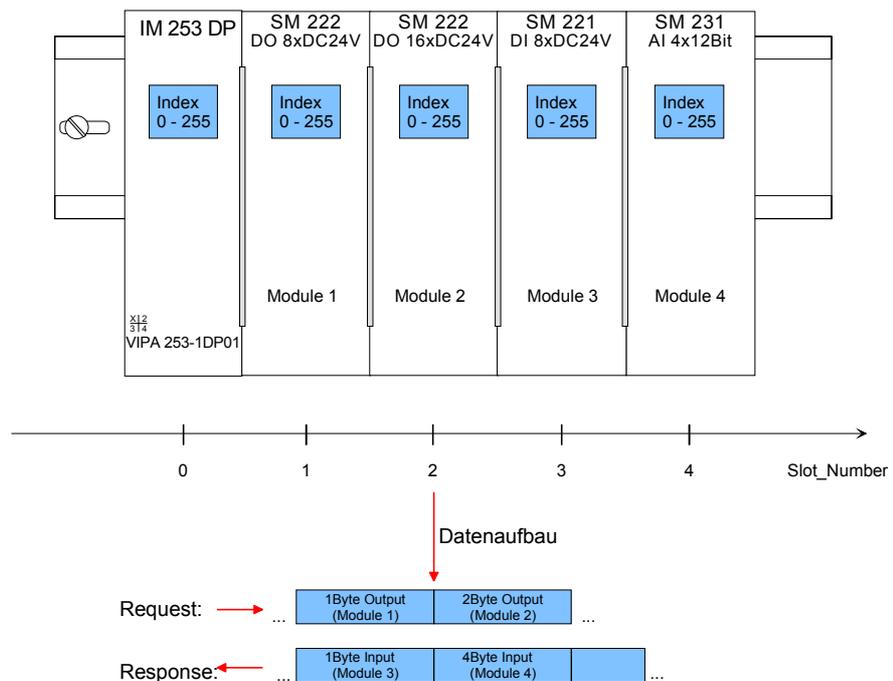
Im *Intel-Format* wird der Wert gedreht und mit aufsteigender Wertigkeit gearbeitet d.h. das 1. Byte beinhaltet das Low-Byte und das 2. Byte das High-Byte.

**Adressierung mit Slot und Index**

Bei der Adressierung von Daten geht PROFIBUS davon aus, dass die Slaves physikalisch modular aufgebaut sind oder aber intern in logische Funktionseinheiten, sogenannte Module strukturiert werden können. Dieses Modell spiegelt sich in den DP-Grundfunktionen für den zyklischen Datenverkehr wieder, bei denen jedes Modul eine konstante Anzahl Ein-/Ausgabebytes besitzt, die an eine feste Position im Nutzdatentelegramm übertragen werden. Das Adressierungsverfahren basiert auf Kennungen, die den Typ eines Moduls als Input, Output oder aus einer Kombination aus beiden kennzeichnen. Alle Kennungen zusammen ergeben die Konfiguration eines Slaves, die im Hochlauf des Systems auch vom DPM 1 überprüft wird.

Auch beim azyklischen Datenverkehr wird dieses Modell zugrunde gelegt. Alle für Schreib- oder Lesezugriffe freigegebenen Datenblöcke werden ebenfalls als den Modulen zugehörig betrachtet und können mit Hilfe von Slot\_Number und Index adressiert werden.

Die Slot\_Number (ID) adressiert dabei das Modul, und der Index die einem Modul zugehörigen Datenblöcke. Die Slot\_Number = 0 adressiert Daten des PROFIBUS-Kopplers, Slot\_Number > 0 adressiert die Daten der/des Funktionsmodule(s).



Jeder Datenblock kann bis zu 244Byte groß sein. Kompaktgeräte werden als eine Einheit von virtuellen Modulen betrachtet. Auch hier gilt die Adressierung mit Slot\_Number und Index.

Durch die Längenangabe im Lese- bzw. Schreib-Befehl können auch nur Teile eines Datenblocks gelesen bzw. geschrieben werden.

**Lese- bzw. Schreibzugriff über SFB 52 bzw. 53**

Ab der Firmware-Version 1.3.0 hat Ihre CPU für DP-V1-Lese- bzw. -Schreibzugriffe den SFB 52 bzw. 53 integriert. Hier können Sie durch Angabe von ID (Slot-Number als Adresse) und Index auf die entsprechende Komponente Ihres Systems zugreifen.

Näheres hierzu finden Sie in der Beschreibung der SFB 52/53.

**Datenübertragung** Es werden defaultmäßig je eine Klasse-1-Master- und eine Klasse-2-Master-Verbindung mit 244 Byte Daten (4 Byte DP-V1-Header plus 240 Byte Nutzdaten) unterstützt. Die Klasse-1-Master-Verbindung wird mit der zyklischen Verbindung zusammen aufgebaut und ist über die Parametrierung zu aktivieren. Die Klasse-2-Master-Verbindung kann von einem C2-Master, der dann nur azyklisch mit dem Slave kommuniziert, benutzt werden und verfügt über einen eigenen Verbindungsaufbau.

**Daten des DP-V1-Slave** Bei Zugriff auf den DP-V1-Koppler über Slot\_Number = 0 haben Sie über Index Zugriff auf folgende Elemente:

Index	Zugriff	Beschreibung
A0h	R	Gerätename (VIPA 253-1DP01)
A1h	R	Hardware-Ausgabestand (V1.00)
A2h	R	Software-Ausgabestand (V1.00)
A3h	R	Serien-Nummer des Gerätes (z.B. 000347 = 30h, 30h, 30h, 33h, 34h, 37h)
A4h	R	Gerätekonfiguration (Modulkonfiguration und Modultypen siehe Folgeseite)
D0h	R	Anzahl der gespeicherten Diagnosen
	W	Löscht Diagnoseeinträge
D1h	R	Diagnoseeintrag der Reihe nach lesen
	W	Speichert Diagnoseeinträge dauerhaft im FLASH-ROM
FFh	R	I&M-Funktionen
	W	

R = Read / lesen; W = Write / schreiben

**Aufbau gespeicherter Diagnoseeintrag** Mit jedem D1h-Aufruf wird ein gespeicherter Diagnoseeintrag, beginnend mit dem jüngsten, mit max. 26Byte ausgegeben.

Grundsätzlich hat jeder gespeicherter Diagnoseeintrag folgenden Aufbau:

Bezeichnung	Typ	Beschreibung
Länge	Wort	Länge der Diagnosedaten
Zeitstempel	Doppelwort	interner Zeitstempel
Diagnose (max. 20Byte)	Byte	Diagnoseeintrag (Alarm), der intern abgelegt wird

**Daten der Funktionsmodule**

Index	Zugriff	Beschreibung
00h	R	Diagnose – Datensatz 0
	W	Modulparameter
01h	R	Über "Index" können Sie durch Vorgabe einer Datensatz-Nr. auf die entsprechende Diagnose eines Moduls zugreifen.  Beispiel: Index=01h → Zugriff auf Diagnose Datensatz 01
F1h	R	Modulparameter
F2h	R	Modulprozessabbild lesen

R = Read (lesen); W = Write (schreiben)

Modulkonfiguration Mit dem Index A3h können Sie die Modulkonfiguration des DP-Slaves ausgeben. Die Zuordnung entnehmen Sie bitte der nachfolgenden Tabelle:

Modultyp	Typkennung	Eingabe-Byte	Ausgabe-Byte
DI 8	9FC1h	1	-
DI 8 - Alarm	1FC1h	1	-
DI 16	9FC2h	2	-
DI 16 / 1C	08C0h	6	6
DI 32	9FC3h	4	-
DO 8	AFC8h	-	1
DO 16	AFD0h	-	2
DO 32	AFD8h	-	4
DIO 8	BFC9h	1	1
DIO 16	BFD2h	2	2
AI2	15C3h	4	-
AI4	15C4h	8	-
AI4 - fast	11C4h	8	-
AI8	15C5h	16	-
AO2	25D8h	-	4
AO4	25E0h	-	8
AO8	25E8h	-	16
AI2 / AO2	45DBh	4	4
AI4 / AO2	45DCh	8	4
SM 238	45DCh	8	4
	38C4h	16	16
CP 240	1CC1h	16	16
FM 250	B5F4h	10	10
FM 250-SSI	B5DBh	4	4
FM 253, FM 254	18CBh	16	16

## IM 253-xDPx1 - DP-V1-Slave - Diagnosefunktionen

### Übersicht

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

Als weitere Funktion wurde bei DP-V1 die gerätebezogene Diagnose verfeinert und in die Kategorien Alarme und Statusmeldungen aufgliedert.

Zusätzlich werden im DP-V1-Slave von VIPA die letzten 100 Alarm-Meldungen mit einem Zeitstempel in einem RAM gespeichert bzw. im Flash gesichert und können mit einer Software ausgewertet werden.

Setzen Sie sich hierzu bitte mit der VIPA-Hotline in Verbindung!

### Interne Diagnose Systemmeldungen

Das System legt auch Diagnosemeldungen ab wie die Zustände "Ready" bzw. "DataExchange", die nicht an den Master weitergeleitet werden.

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.

### Diagnosedaten manuell sichern

Über die kurzzeitige Einstellung von 00 am Adress-Schalter können Sie die Diagnose-Daten während des "DataExchange" im Flash-ROM sichern.

### Diagnosemeldung bei Spannungsausfall

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 Unterspannung/Abschaltung-Diagnosemeldung aus dem Zeitstempel des EEPROMs generiert und im Diagnose-RAM abgelegt.

**Aufbau der DP-V1-Diagnosedaten über Profibus**

Die Diagnose-Meldungen, die vom Profibus-Slave erzeugt werden, haben je nach Parametrierung eine Länge von 58Byte.  
Sobald der Profibus-Slave an den Master eine Diagnose sendet, werden den max. 58Byte Diagnosedaten 6Byte Normdiagnose-Daten vorangestellt:

Byte 0 ... Byte 5	Normdiagnose-Daten	
Byte 6 ... 10	Kennungsbezogene Diagnose*	
x ... x+11	Modulstatus*	
7...13 ·(x ... x+2)	Kanalbezogene Diagnose*	
x ... x+19	Alarm*	Diagnose, die intern abgelegt wird

\*) Über Parametrierung sperr- oder freischaltbar

Diagnosedaten IM 253-1DP31 - ECO

Aufgrund der Einschränkungen ergeben sich für den IM 253-1DP31 - ECO folgende Diagnosedaten:

Byte 0 ... Byte 5	Normdiagnose-Daten	
Byte 6 ... 7	Kennungsbezogene Diagnose*	
x ... x+5	Modulstatus*	
10...13 ·(x ... x+2)	Kanalbezogene Diagnose*	
x ... x+19	Alarm*	Diagnose, die intern abgelegt wird

\*) Über Parametrierung sperr- oder freischaltbar

**Norm-Diagnosedaten**

Bei der Übertragung einer Diagnose an den Master werden die *Slave-Norm-Diagnosedaten* den Diagnose-Bytes vorangestellt. Nähere Angaben zum Aufbau der *Slave-Norm-Diagnosedaten* finden Sie in den Normschriften der Profibus Nutzer Organisation.

Die *Slave-Norm-Diagnosedaten* 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

### Kennungs- bezogene Diagnose

Über die *kennungsbezogene Diagnose*, die über die Parametrierung aktiviert werden kann, erhalten Sie Informationen, an welchem Steckplatz (Modul) ein Fehler aufgetreten ist. Nähere Informationen über den Fehler erhalten Sie mit dem *Modulstatus* und der *kanalbezogenen Diagnose*.



#### Hinweis!

Bitte beachten Sie, dass die Länge der *kennungsbezogene Diagnose* beim IM 253-1DP31 - ECO auf 2 begrenzt ist.

#### Kennungsbezogene Diagnose

Byte	Bit 7 ... Bit 0
X	Bit 5 ... 0: 000101 Länge kennungsbezogene Diagnose* Bit 7 ... 6: 01 (fix) Code für kennungsbezogenen Diagnose
X+1	Die Bits der Module je Steckplatz werden gesetzt, wenn: - das Modul gezogen wird - ein nicht projektiertes Modul gesteckt wird - auf ein Modul nicht zugegriffen werden kann - ein Modul einen Diagnosealarm meldet Bit 0: Eintrag Modul Steckplatz 1 Bit 1: Eintrag Modul Steckplatz 2 Bit 2: Eintrag Modul Steckplatz 3 Bit 3: Eintrag Modul Steckplatz 4 Bit 4: Eintrag Modul Steckplatz 5 Bit 5: Eintrag Modul Steckplatz 6 Bit 6: Eintrag Modul Steckplatz 7 Bit 7: Eintrag Modul Steckplatz 8
X+2	Bit 0: Eintrag Modul Steckplatz 9 Bit 1: Eintrag Modul Steckplatz 10 Bit 2: Eintrag Modul Steckplatz 11 Bit 3: Eintrag Modul Steckplatz 12 Bit 4: Eintrag Modul Steckplatz 13 Bit 5: Eintrag Modul Steckplatz 14 Bit 6: Eintrag Modul Steckplatz 15 Bit 7: Eintrag Modul Steckplatz 16
X+3	Bit 0: Eintrag Modul Steckplatz 17 Bit 1: Eintrag Modul Steckplatz 18 Bit 2: Eintrag Modul Steckplatz 19 Bit 3: Eintrag Modul Steckplatz 20 Bit 4: Eintrag Modul Steckplatz 21 Bit 5: Eintrag Modul Steckplatz 22 Bit 6: Eintrag Modul Steckplatz 23 Bit 7: Eintrag Modul Steckplatz 24
X+4	Bit 0: Eintrag Modul Steckplatz 25 Bit 1: Eintrag Modul Steckplatz 26 Bit 2: Eintrag Modul Steckplatz 27 Bit 3: Eintrag Modul Steckplatz 28 Bit 4: Eintrag Modul Steckplatz 29 Bit 5: Eintrag Modul Steckplatz 30 Bit 6: Eintrag Modul Steckplatz 31 Bit 7: Eintrag Modul Steckplatz 32

\*) Bit 5 ... 0: 000010 bei 253-1DP31 - ECO

**Modulstatus**

Mit dem *Modulstatus*, der über die Parametrierung aktiviert werden kann, erhalten Sie nähere Informationen zum Fehler, der in einem Modul aufgetreten ist.

**Hinweis!**

Bitte beachten Sie, dass die Länge des *Modulstatus* beim IM 253-1DP31 - ECO auf 6 begrenzt ist.

*Modulstatus*

Byte	Bit 7 ... Bit 0
X	Bit 5 ... 0: 001100 (fix) Länge des Modulstatus* Bit 7 ... 6: 00 (fix) Code für Modulstatus
X+1	82h (fix) Statustyp Modulstatus
X+2	00h (fix)
X+3	00h (fix)
X+4	Für Steckplatz 1 ... 32 sind folgende Fehler spezifiziert: 00: Modul hat gültige Daten 01: Modulfehler - ungültige Daten (Modul defekt) 10: Falsches Modul - ungültige Daten 11: kein Modul gesteckt - ungültige Daten Bit 1, 0: Modulstatus Modul Steckplatz 1 Bit 3, 2: Modulstatus Modul Steckplatz 2 Bit 5, 4: Modulstatus Modul Steckplatz 3 Bit 7, 6: Modulstatus Modul Steckplatz 4
X+5	Bit 1, 0: Modulstatus Modul Steckplatz 5 Bit 3, 2: Modulstatus Modul Steckplatz 6 Bit 5, 4: Modulstatus Modul Steckplatz 7 Bit 7, 6: Modulstatus Modul Steckplatz 8
X+6	Bit 1, 0: Modulstatus Modul Steckplatz 9 Bit 3, 2: Modulstatus Modul Steckplatz 10 Bit 5, 4: Modulstatus Modul Steckplatz 11 Bit 7, 6: Modulstatus Modul Steckplatz 12
X+7	Bit 1, 0: Modulstatus Modul Steckplatz 13 Bit 3, 2: Modulstatus Modul Steckplatz 14 Bit 5, 4: Modulstatus Modul Steckplatz 15 Bit 7, 6: Modulstatus Modul Steckplatz 16
X+8	Bit 1, 0: Modulstatus Modul Steckplatz 17 Bit 3, 2: Modulstatus Modul Steckplatz 18 Bit 5, 4: Modulstatus Modul Steckplatz 19 Bit 7, 6: Modulstatus Modul Steckplatz 20
X+9	Bit 1, 0: Modulstatus Modul Steckplatz 21 Bit 3, 2: Modulstatus Modul Steckplatz 22 Bit 5, 4: Modulstatus Modul Steckplatz 23 Bit 7, 6: Modulstatus Modul Steckplatz 24
X+10	Bit 1, 0: Modulstatus Modul Steckplatz 25 Bit 3, 2: Modulstatus Modul Steckplatz 26 Bit 5, 4: Modulstatus Modul Steckplatz 27 Bit 7, 6: Modulstatus Modul Steckplatz 28
X+11	Bit 1, 0: Modulstatus Modul Steckplatz 29 Bit 3, 2: Modulstatus Modul Steckplatz 30 Bit 5, 4: Modulstatus Modul Steckplatz 31 Bit 7, 6: Modulstatus Modul Steckplatz 32

\*) Bit 5 ... 0: 000110 bei 253-1DP31 - ECO

**Kanalbezogene Diagnose**

Mit der *kanalbezogene Diagnose* erhalten Sie detaillierte Informationen über Kanal-Fehler innerhalb eines Moduls. Für den Einsatz der *kanalbezogenen Diagnose* muss für jedes Modul über die Parametrierung der Diagnosealarm freigegeben werden. Die *kanalbezogene Diagnose* kann über die Parametrierung aktiviert werden und hat folgenden Aufbau:

*Kanalbezogene Diagnose für einen Kanal*

Byte	Bit 7 ... Bit 0
X	Bit 5 ... 0: Kennungsnummer des Moduls, das die kanalbezogene Diagnose liefert (000001 ... 011111)* z.B.: Steckplatz 1 hat die Kennungsnr. 0 Steckplatz 32 hat die Kennungsnr. 31 Bit 7, 6: 10 (fix) Code für kanalbezogene Diagnose
X+1	Bit 5 ... 0: Nummer des Kanals bzw. der Kanalgruppe, der die Diagnose liefert (00000 .... 11111) Bit 7 ... 6: 01=Eingabe Modul 10=Ausgabe Modul 11=Ein-/Ausgabe Modul
X+2	Bit 4 ... 0: <i>Fehlertyp nach Profibus-Norm</i> 00001: Kurzschluss 00010: Unterspannung (Versorgungsspannung) 00011: Überspannung (Versorgungsspannung) 00100: Ausgabe Modul ist überlastet 00101: Übertemperatur Ausgabe-Modul 00110: Leitungsbruch des Sensors oder Aktors 00111: Oberer Grenzwert überschritten 01000: Unterer Grenzwert überschritten 01001: Fehler - Lastspannung am Ausgang - Geberversorgung - Hardwarefehler des Moduls  <i>Fehlertyp herstellerspezifisch</i> 10000: Parametrierfehler 10001: Geber oder Lastspannung fehlt 10010: Sicherung defekt 10100: Massefehler 10101: Referenzkanalfehler 10110: Prozessalarm verloren 11001: Sicherheitsgerichtete Abschaltung 11010: Externer Fehler 11010: Unklarer Fehler - nicht spezifizierbar  Bit 7 ... 5: Kanaltyp 001: Bit 010: 2 Bit 011: 4 Bit 100: Byte 101: Wort 110: 2 Worte

\*) Bit 5 ... 0: 000001...001000 (Steckplatz 1...8) bei 253-1DP31 - ECO

Die maximale Anzahl von *kanalbezogenen Diagnosen* ist begrenzt durch die 58Byte maximale Gesamtlänge der Diagnose. Durch Deaktivierung anderer Diagnosebereiche können Sie diese Bereiche für weitere *kanalbezogenen Diagnosen* freigeben. Pro Kanal werden immer 3Byte verwendet.

**Alarme**

Der Alarmteil der Slave-Diagnose gibt Auskunft über den Alarmtyp und die Ursache, die zum Auslösen eines Alarms geführt hat. Der Alarmteil besteht aus maximal 20Byte. Pro Slave-Diagnose kann maximal 1 Alarm gemeldet werden. Der Alarmteil ist immer der letzte Teil im Diagnosetelegramm.

**Inhalt**

Der Inhalt der Alarminformation ist abhängig vom Alarmtyp:

- Bei *Diagnosealarmen* werden als Alarmzusatzinformation ab Byte x+4 16Byte angehängt, die dem Datensatz 1 der CPU-Diagnose entsprechen, gesendet.
- Bei *Prozessalarmen* beträgt die Länge der Alarmzusatzinformation 4Byte. Diese Daten sind modulspezifisch und bei dem jeweiligen Modul beschrieben.

**Alarmstatus**

Liegt ein Diagnoseereignis für Kanal/Kanalgruppe 0 eines Moduls vor, so kann neben einem Kanalfehler auch ein Modulfehler vorliegen.

Ein Eintrag erfolgt in diesem Fall auch dann, wenn Sie für Kanal/Kanalgruppe 0 des Moduls die Diagnose nicht freigegeben haben.

Der Alarmteil ist wie folgt aufgebaut:

*Alarmstatus Byte x ... x+3*

Byte	Bit 7 ... Bit 0
x	Bit 5 ... 0: 010100: Länge des Alarmteils inkl. Byte x Bit 7 ... 6: Code für gerätebezogene Diagnose
x+1	Bit 6 ... 0: Alarmtyp 0000001: Diagnosealarm 0000010: Prozessalarm Bit 7: Code für Alarm
x+2	Bit 7 ... 0: Steckplatznummer des Moduls, das Alarm liefert 1 ... 32
x+3	Bit 1, 0: 00: Prozessalarm 01: Diagnosealarm <sub>kommend</sub> 10: Diagnosealarm <sub>gehend</sub> 11: reserviert Bit 2: 0 (fix) Bit 7 ... 3: Alarmsequenznummer 1...32

*Alarmstatus bei Diagnosealarm Bytes x+4 bis x+7  
(entspricht CPU-Diagnose-Datensatz 0)*

Byte	Bit 7 ... Bit 0
x+4	Bit 0: Modulstörung, d.h. ein Fehler wurde erkannt Bit 1: Interner Fehler im Modul Bit 2: Externer Fehler - Modul nicht mehr ansprechbar Bit 3: Kanalfehler im Modul Bit 4: Lastspannungsversorgung fehlt Bit 5: Frontstecker fehlt Bit 6: Modul ist nicht parametrierbar Bit 7: Parametrierfehler
x+5	Bit 3 ... 0: Modulkategorie 1111: Digitalmodul 0101: Analogmodul 1000: FM 1100: CP Bit 4: Kanalinformation vorhanden Bit 5: Anwenderinformation vorhanden Bit 6: "0" Bit 7: "0"
x+6	Bit 0: Speicher- bzw. Messbereichsmodul Analogmodul fehlt Bit 1: Kommunikationsstörung Bit 2: Betriebszustand 0: RUN 1: STOP Bit 3: Zyklusüberwachungszeit Bit 4: Modul Spannungsversorgung fehlt Bit 5: Batterie leer Bit 6: Pufferung ausgefallen Bit 7: "0"
x+7	Bit 0: reserviert Bit 1: reserviert Bit 2: reserviert Bit 3: reserviert Bit 4: reserviert Bit 5: reserviert Bit 6: Prozessalarm verloren Bit 7: reserviert

*Fortsetzung ...*

... Fortsetzung

Alarmstatus bei Diagnosealarm Bytes x+8 bis x+19  
(entspricht CPU-Diagnose-Datensatz 1)

Byte	Bit 7 ... Bit 0
x+8	70h: Modul mit Digitaleingängen 71h: Modul mit Analogeingängen 72h: Modul mit Digitalausgängen 73h: Modul mit Analogausgängen 74h: Modul mit Analogein-/ausgängen 76h: Zähler
x+9	Länge der kanalspezifischen Diagnose
x+10	Anzahl der Kanäle pro Modul
x+11	Position (Kanal) des Diagnoseereignisses
x+12	Diagnoseereignis für Kanal/Kanalgruppe 0 Belegung siehe Modulbeschreibung
x+13	Diagnoseereignis für Kanal/Kanalgruppe 1 Belegung siehe Modulbeschreibung
.	.
.	.
.	.
x+19	Diagnoseereignis für Kanal/Kanalgruppe 7 Belegung siehe Modulbeschreibung

Alarmstatus bei Prozessalarm Bytes x+4 bis x+7

Nähere Angaben zu den Diagnosedaten finden Sie in der jeweiligen Modul-Beschreibungen.

## IM 253-xDPx1 - DP-V1-Slave - Firmware-Update

### Übersicht

Ein Firmwareupdate für den DP-V1-Slave VIPA 253-1DP01 über Profibus ist zurzeit ausschließlich über ein SPS-System mit Siemens CPU möglich. Hierbei wird Ihre Firmware aus dem Hardware-Konfigurator online an die CPU geleitet, die die Firmware mit dem angebundnen DP-Master über Profibus an den entsprechenden DP-Slave weiterleitet.



### Hinweis!

Für den DP-Slave IM 253-1DP31-ECO und IM 253-1DP11 gibt es zurzeit noch keine Firmwareupdate-Möglichkeit.

### Vorgehensweise

- Firmwaredatei bereitstellen
- Hardware-Konfigurator mit Projekt laden
- Firmware übertragen

Firmwaredatei  
*header.upd*  
bereitstellen

Die aktuellste Firmware für die DP-V1-Profibus-Slaves finden Sie unter [ftp.vipa.de/support/firmware/System%20200V/DP\\_Slave/IM253-1DP01](ftp.vipa.de/support/firmware/System%20200V/DP_Slave/IM253-1DP01) als Package P<sub>x</sub>000019\_V<sub>xxx</sub>.zip mit xxx=Version.

Entpacken Sie die Datei und kopieren Sie die Datei *header.upd* in Ihr Arbeitsverzeichnis.

Hardware-  
Konfigurator mit  
Projekt laden

- Öffnen Sie den Hardware-Konfigurator mit dem projektierten DP-Slave
- Klicken Sie auf den DP-Slave und wählen Sie **Zielsystem** > *Firmware aktualisieren*. Dieser Menübefehl ist nur dann aktivierbar, wenn der markierte DP-Slave die Funktion "Firmware aktualisieren" unterstützt.  
→ Es öffnet sich nun das Dialogfeld "Firmware aktualisieren".
- Wählen Sie über die Schaltfläche "Durchsuchen" Ihr Arbeitsverzeichnis an, das die Datei *header.upd* beinhaltet. Wählen Sie die Datei *header.upd* aus. → Sie erhalten Information, für welche Module und ab welcher Firmware-Version die ausgewählte Datei geeignet ist.
- Aktivieren Sie das Kontroll-Feld "Firmware nach Laden aktivieren", denn nur dann wird die neue Firmware in das Flash kopiert, und klicken Sie auf [Ausführen]. → Es wird geprüft, ob die ausgewählte Datei gültig ist und diese bei positiver Prüfung an den DP-Slave übertragen.



### Hinweis!

Im laufenden Betrieb erfolgt nach ca. 3s ein Firmwareupdate auf dem DP-Slave. Bitte beachten Sie, dass hierbei von dem DP-Slave ein Neustart durchgeführt wird, wobei der DP-Master in STOP verbleiben bzw. Ihr Anwenderprogramm beeinträchtigt werden könnte.

## IM 253-xDPx1 - DP-V1-Slave - I&M-Daten

### Übersicht

Identifikations- und Maintenance-Daten (I&M) sind in einer Baugruppe gespeicherte Informationen, die Sie unterstützen beim:

- Überprüfen der Anlagenkonfiguration
- Auffinden von Hardware-Änderungen einer Anlage
- Beheben von Fehlern in einer Anlage

Identifikationsdaten (I-Daten) sind Informationen zur Baugruppe, wie z. B. Bestellnummer und Seriennummer, die zum Teil auch auf dem Gehäuse der Baugruppe aufgedruckt sind. I-Daten sind Herstellerinformationen zur Baugruppe und können nur gelesen werden.

Maintenance-Daten (M-Daten) sind anlagenabhängige Informationen, wie z.B. Einbauort und Einbaudatum. M-Daten werden während der Projektierung erstellt und auf die Baugruppe geschrieben.

Mit den I&M-Daten können Baugruppen online eindeutig identifiziert werden. Ab Profibus-Firmwareversion V110 sind diese Daten auf den Profibus-Kopplern verfügbar.



### Hinweis!

Auf die I&M-Daten eines Profibus-Kopplers darf zu einem Zeitpunkt nur ein DP-Master zugreifen.

### Aufbau

Die Datenstrukturen der I&M-Daten entsprechen den Festlegungen der Profibus Guideline - Best.-Nr. 3.502, Version 1.1 vom Mai 2003.

I&M-Daten	Zugriff	Voreinstellung	Erläuterung
Identifikationsdaten 0: IM_INDEX: 65000			
MANUFACTURER_ID	lesen (2Byte)	22B hex (555 dez)	Hier ist der Name des Herstellers gespeichert. (555 dez = VIPA GmbH)
ORDER_ID	lesen (20Byte)	abhängig von der Baugruppe	Hier ist die Bestellnummer der Baugruppe gespeichert. VIPA 253-1DP01/31
SERIAL_NUMBER	lesen (16Byte)	abhängig von der Baugruppe	Hier ist die Seriennummer der Baugruppe gespeichert. Damit ist eine eindeutige Identifikation der Baugruppe möglich.
HARDWARE_REVISION	lesen (2Byte)	abhängig von der Baugruppe	Hier ist der Erzeugnisstand der Baugruppe gespeichert. Wird hochgezählt, wenn sich Erzeugnisstand bzw. Firmware der Baugruppe ändert.

Fortsetzung ...

... Fortsetzung

SOFTWARE_REVISION	lesen (4Byte)	Firmware-Version Vxyz	Gibt Auskunft über die Firmware-Version der Baugruppe. Wird die Firmware-Version hochgezählt, dann erhöht sich ebenfalls der Erzeugnisstand (HARDWARE_REVISION) der Baugruppe.
REVISION_COUNTER	lesen (2Byte)	0000 hex	reserviert
PROFILE_ID	lesen (2Byte)	F600 hex	Generic Device
PROFILE_SPECIFIC_TYPE	lesen (2Byte)	0003 hex	auf I/O-Module
IM_VERSION	lesen (2Byte)	0101 hex	Gibt Auskunft über die Version der I&M-Daten. (0101 hex = Version 1.1)
IM_SUPPORTED	lesen (2Byte)	001F hex	Gibt Auskunft über die vorhandenen I&M-Daten. (IM_INDEX: 650000 ...65004)
Maintenance-Daten 1: IM_INDEX: 65001			
TAG_FUNCTION	lesen / schreiben (32Byte)	–	Geben Sie hier eine anlagenweit eindeutige Kennzeichnung für die Baugruppe ein.
TAG_LOCATION	lesen / schreiben (22Byte)	–	Geben Sie hier den Einbauort der Baugruppe ein.
Maintenance-Daten 2: IM_INDEX: 65002			
INSTALLATION_DATE	lesen / schreiben (16Byte)	–	Geben Sie hier für die Baugruppe das Einbaudatum und ggf. die zugehörige Uhrzeit ein.
RESERVED	lesen / schreiben (38Byte)	–	reserviert
Maintenance-Daten 3: IM_INDEX: 65003			
DESCRIPTOR	lesen / schreiben (54Byte)	–	Geben Sie hier einen Kommentar zur Baugruppe ein.
Maintenance-Daten 4: IM_INDEX: 65004			
SIGNATURE	lesen / schreiben (54Byte)	–	Geben Sie hier einen Kommentar zur Baugruppe ein.

## 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. 126 Teilnehmer.
  - Pro Segment sind max. 32 Teilnehmer zulässig.
  - Die maximale Segmentlänge hängt von der Übertragungsrate ab:
 

9,6 ... 187,5kBaude	→	1000m
500kBaude	→	400m
1,5Mbaude	→	200m
3 ... 12Mbaude	→	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:
 

≤ 1,5Mbaude	→	17 Teilnehmer inkl. Master
3Mbaude	→	15 Teilnehmer inkl. Master
6Mbaude	→	7 Teilnehmer inkl. Master
12Mbaude	→	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**

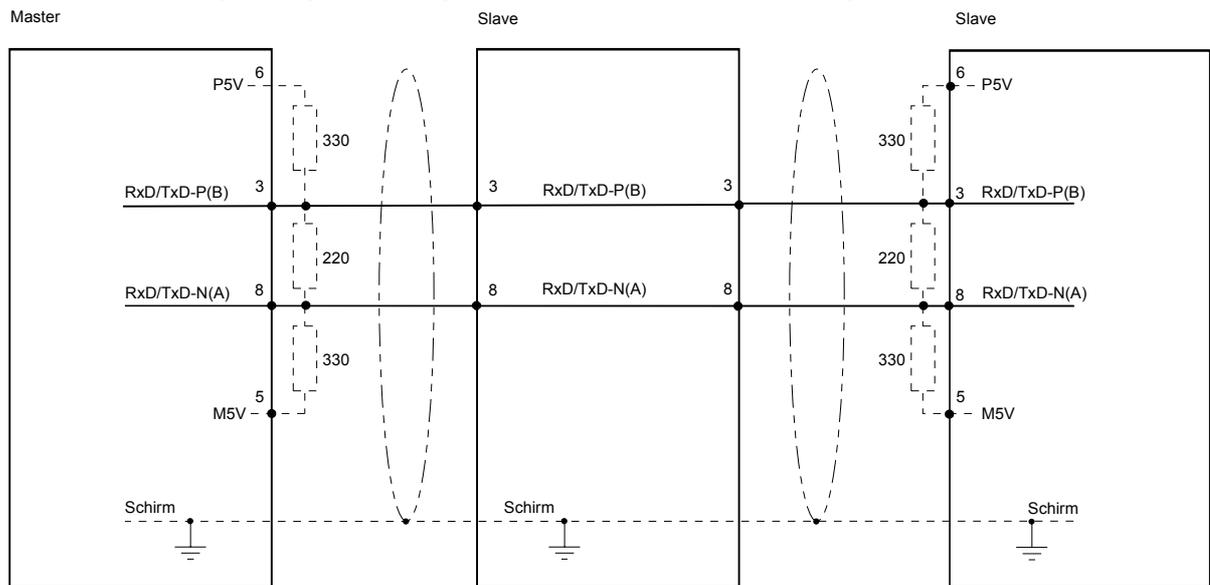
- Bauen Sie Ihr Profibus-System mit den entsprechenden Modulen auf.
- Stellen Sie an Ihrem Buskoppler eine Adresse ein, die in Ihrem Bus noch nicht verwendet wird.
- Übertragen Sie die mitgelieferte GSD-Datei in Ihr Projektiersystem und projektieren Sie Ihr System.
- Übertragen Sie Ihre Projektierung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

**Profibus mit RS485**

Profibus verwendet als Übertragungsmedium eine geschirmte, verdrehte Zweidrahtleitung auf Basis der RS485-Schnittstelle. Die Profibus-Leitung muss mit einem Wellenwiderstand abgeschlossen werden.

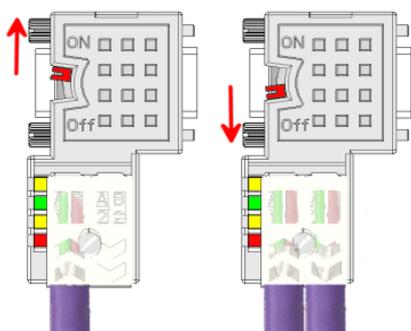
**Busverbindung**

In der nachfolgenden Abbildung sind die Abschlusswiderstände der jeweiligen Anfangs- und Endstation stilisiert dargestellt.



**Leitungsabschluss mit "EasyConn"**

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



**Achtung!**

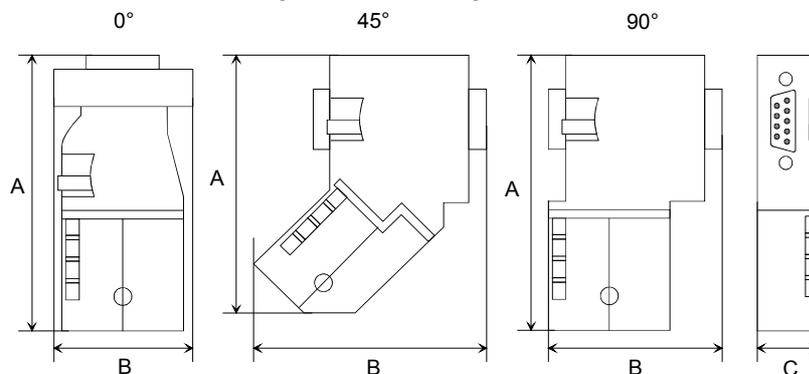
Der Abschlusswiderstand wird nur wirksam, wenn der Stecker an einem Slave gesteckt ist und der Slave mit Spannung versorgt wird.

**Hinweis!**

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

"EasyConn" Busanschluss-Stecker

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 Busanschlusstecker mit zuschaltbarem Abschlusswiderstand und integrierter Busdiagnose.



	0°	45°	90°
A	64	61	66
B	34	53	40
C	15,8	15,8	15,8

Maße in mm



**Hinweis!**

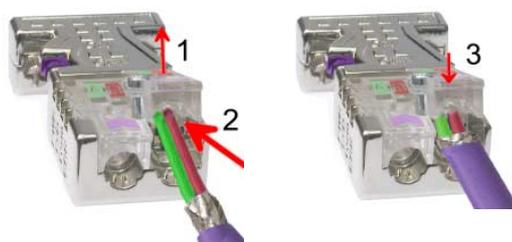
Zum Anschluss dieses Steckers verwenden Sie bitte die Standard Profibus-Leitung Typ A mit Drahtseele nach EN50170.

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



Maße in mm

Montage



- Lösen Sie die Schraube.
- Klappen Sie den Kontaktdeckel auf.
- Stecken Sie beide Adern in die dafür vorgesehenen Öffnungen (Farbzuordnung wie unten beachten!).
- Bitte beachten Sie, dass zwischen Schirm und Datenleitungen kein Kurzschluss entsteht!
- Schließen Sie den Kontaktdeckel.
- Ziehen Sie die Schraube wieder fest (max. Anzugsmoment 4Nm).

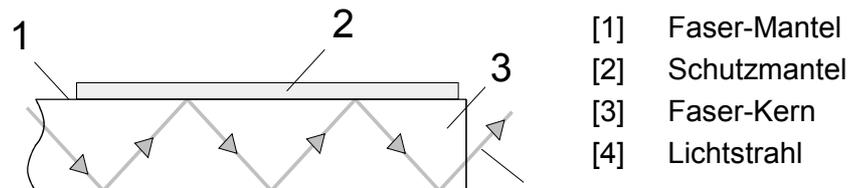
**Bitte beachten:** Den **grünen** Draht immer an **A**, den **roten** immer an **B** anschließen!

**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

**LWL-Kabel  
LWL-Stecker**

VIPA empfiehlt Ihnen, LWL-Stecker und Kabel von der Firma Hewlett Packard (HP) zu verwenden:

*HP-Best.-Nr.: LWL-Kabel*

HFBR-RUS500, HFBR-RUD500, HFBR-EUS500, HFBR-EUD500

*HP-Best.-Nr.: LWL-Stecker*

Mit Crimp-Montage: HFBR-4506 (grau), HFBR-4506B (schwarz)

Ohne Crimp-Montage: HFBR-4531

Näheres hierzu siehe Folgeseite.

**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.

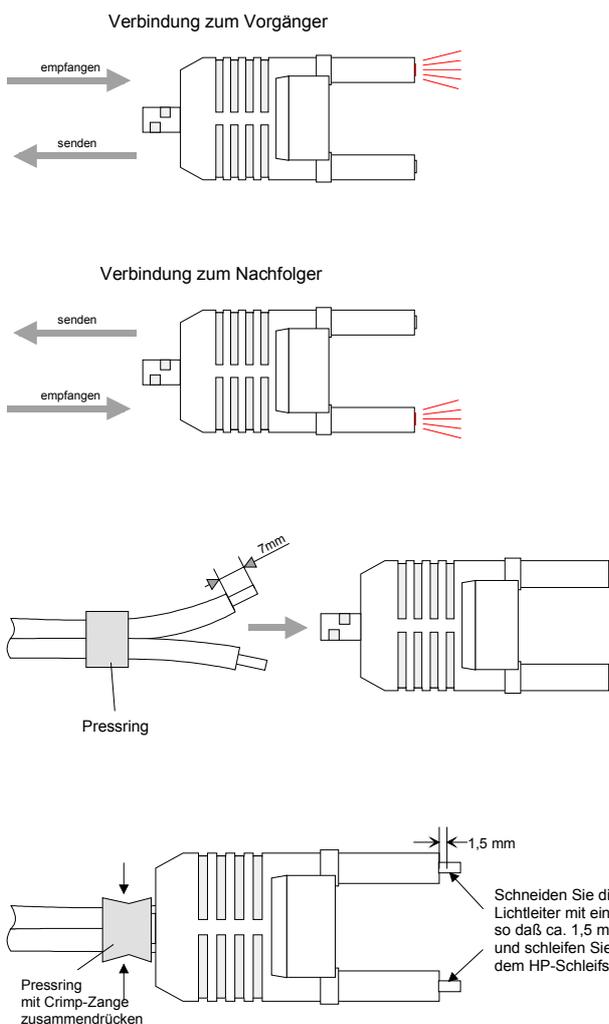
Auf dem VIPA Bus-Koppler befinden sich 4 LWL-Anschlüsse. An der Buchsenfarbe können Sie die Kommunikationsrichtung erkennen (dunkel: empfangen, hell: senden).

Bei eingeschaltetem Bus erkennen Sie am Licht die Faser für den Empfang und die dunkle Faser für das Senden.

Die Stecker der Firma Hewlett Packard (HP) 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.

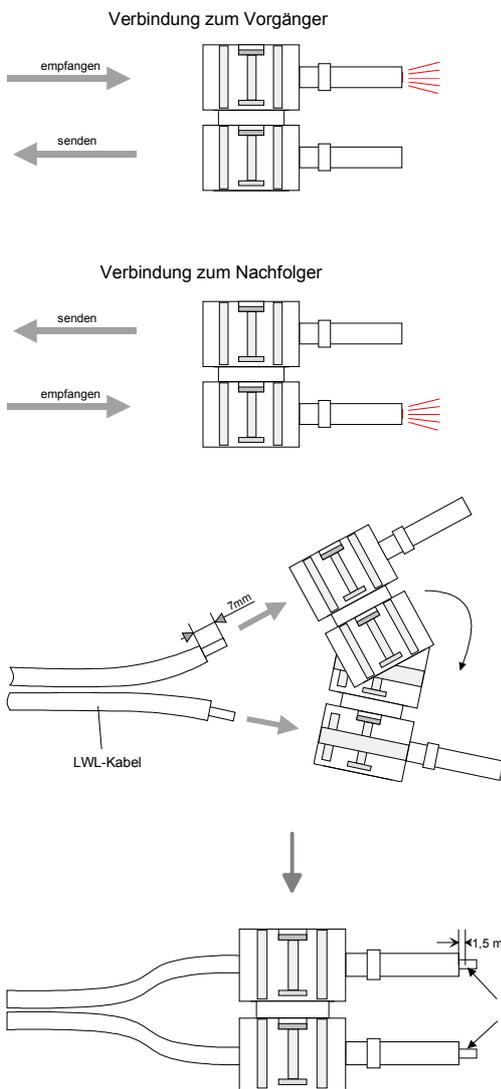
Mit einer Abisolierzange entfernen Sie die Schutzhülle, dass ca. 7mm der Faser sichtbar werden.

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

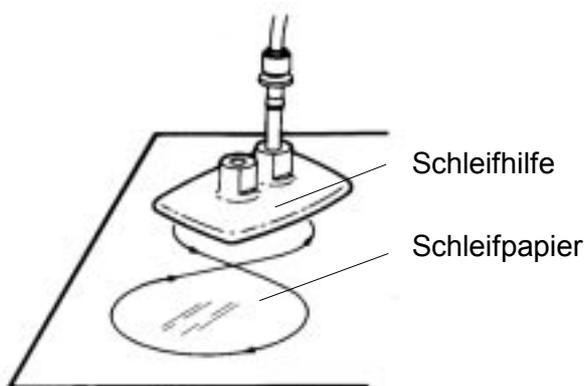
Schieben Sie den Pressring auf den Stecker und quetschen Sie den Ring mit der Crimp-Zange zusammen.

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.).

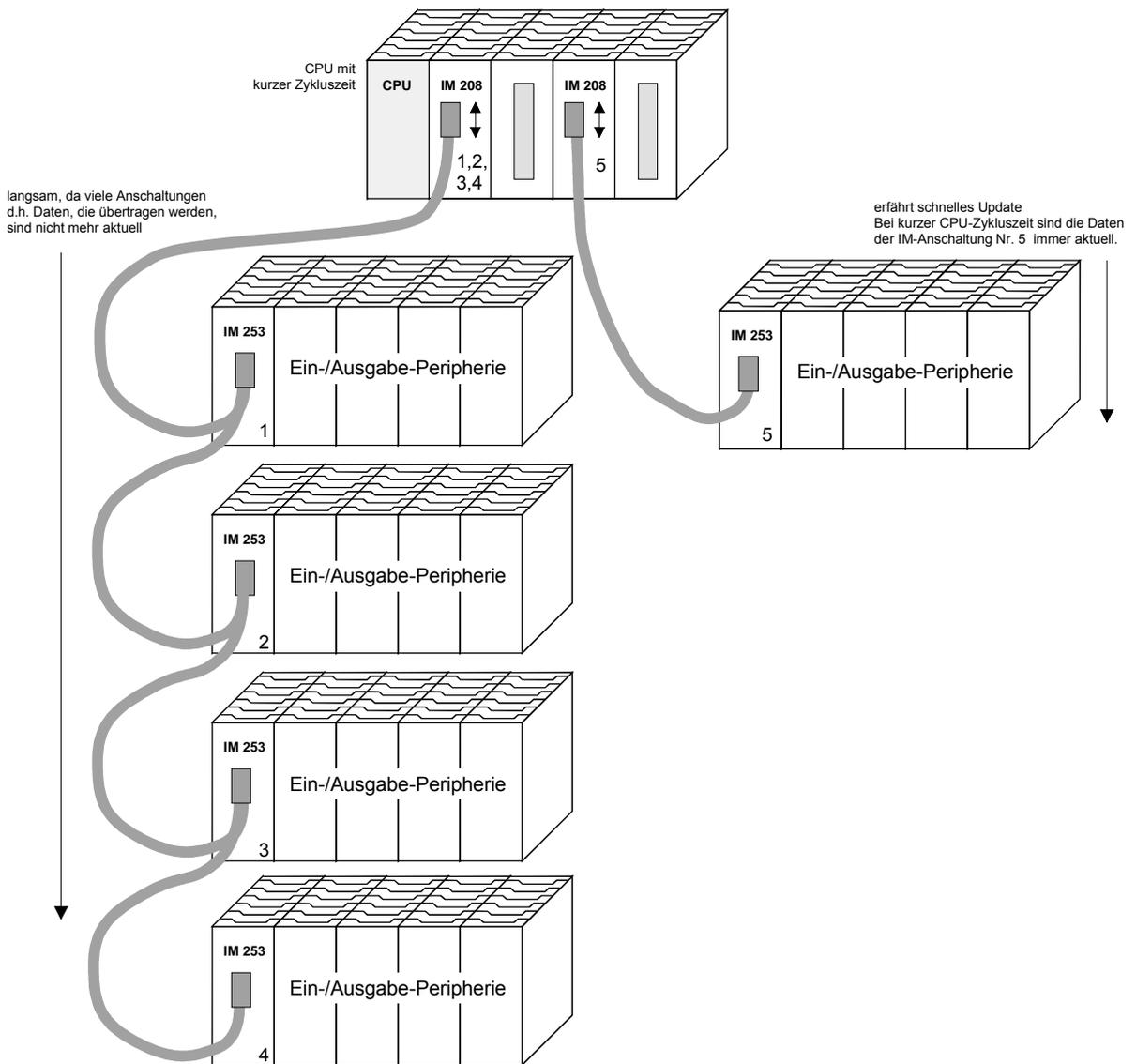
Schneiden Sie mit einer Klinge die Faser ab, so dass 1,5mm Faser noch sichtbar sind. Verwenden Sie zum Planschleifen das Schleifset von HP (HP-Best.-Nr.:HFBR-4593).

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

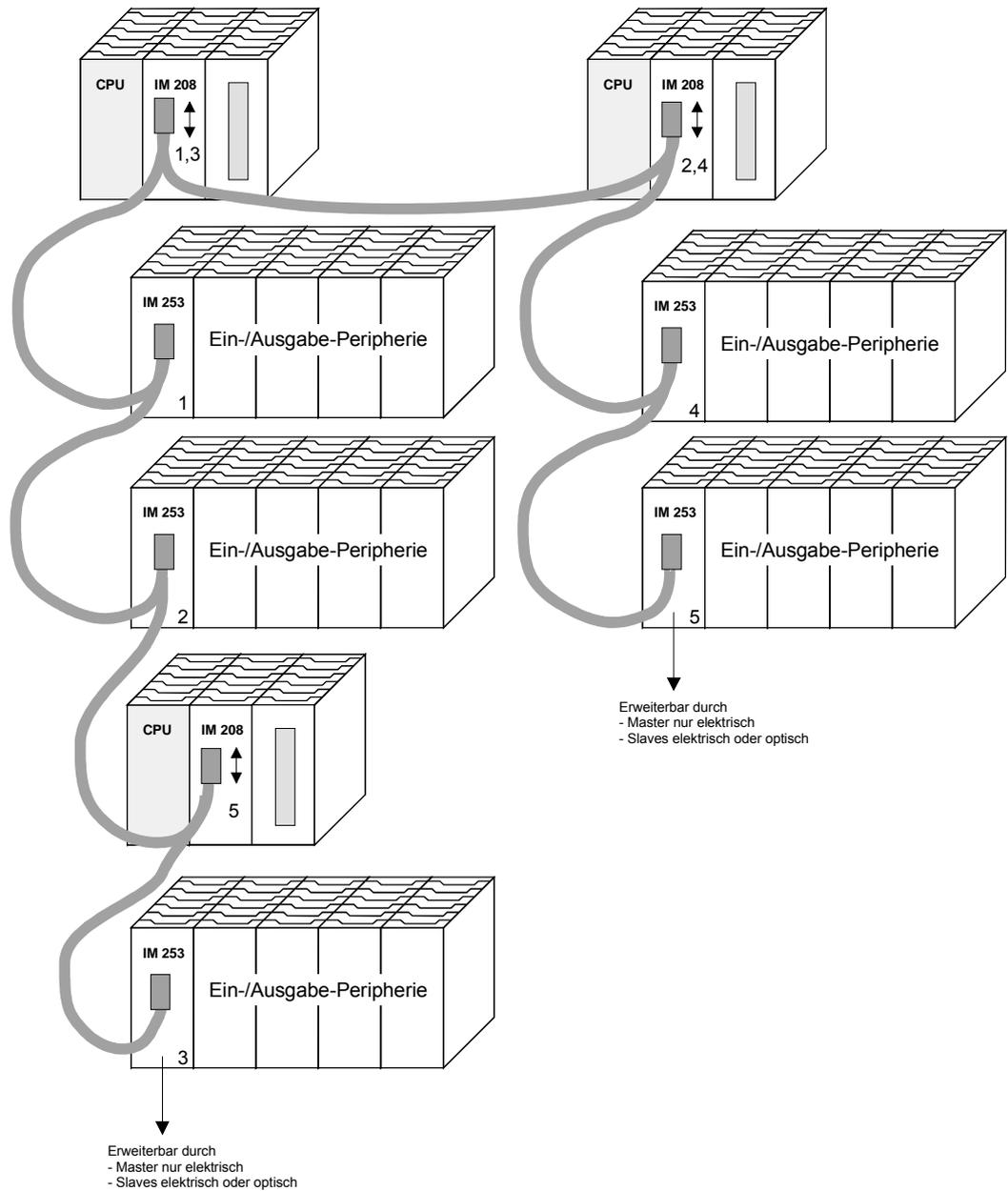
#### Eine CPU und mehreren Master-Anschaltungen

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.

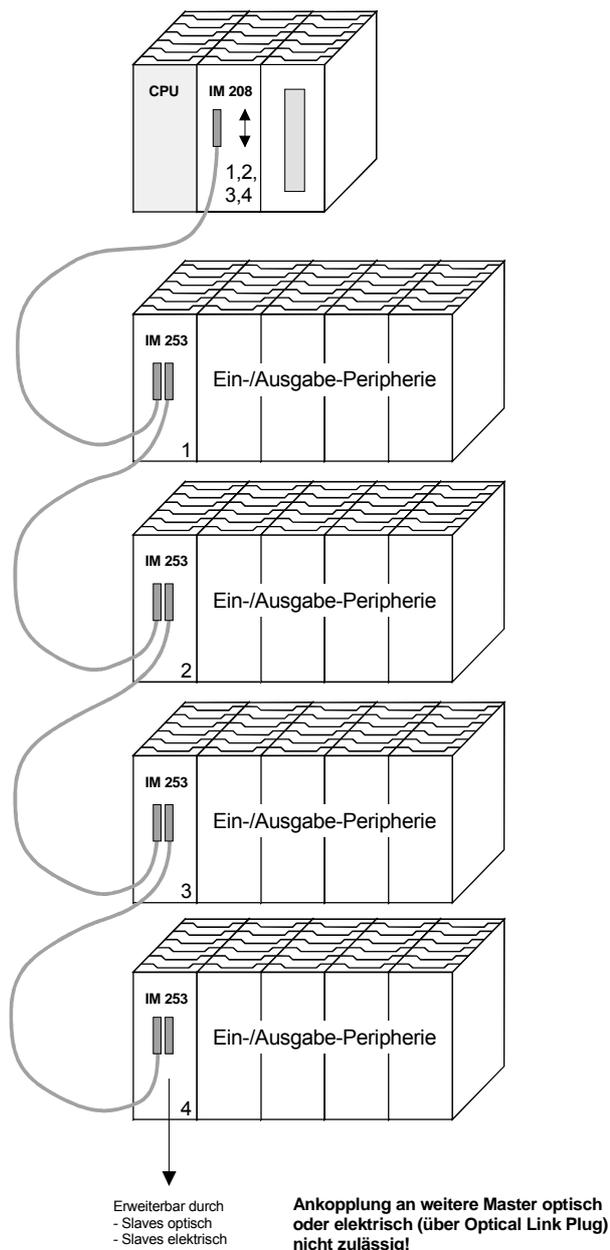


**Multi Master System**

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

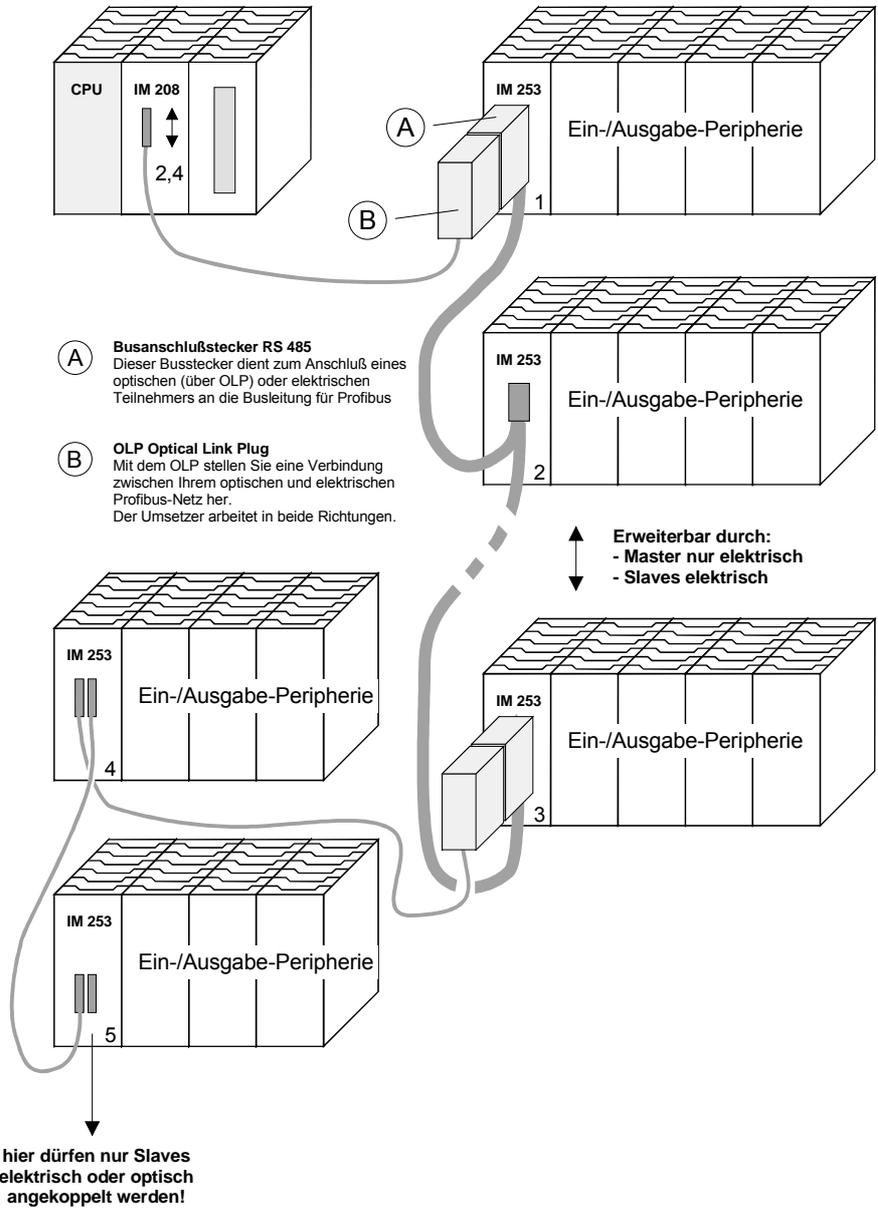


### Optischer Profibus



**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!**

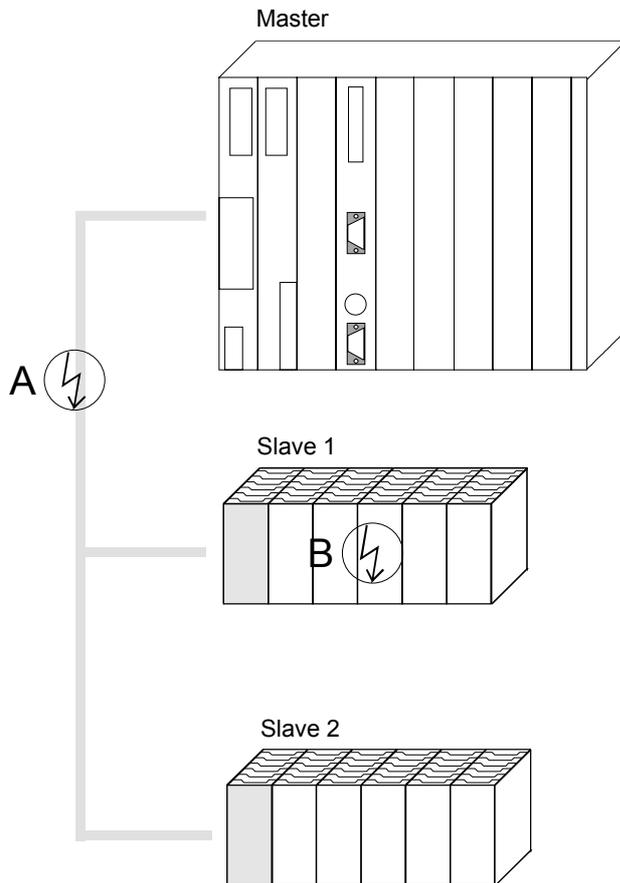


### Hinweis!

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

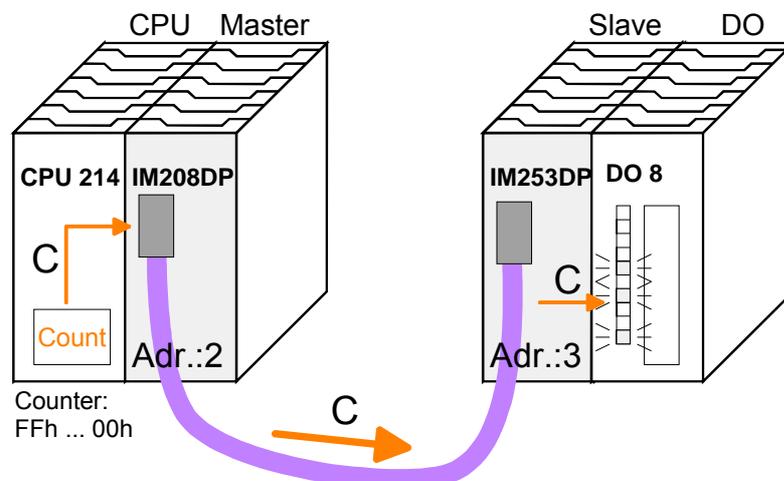
In diesem Beispiel soll eine Kommunikation zwischen einem Master- und einem Slave-System gezeigt werden.

Das Master-System besteht aus einer CPU 21x (hier CPU 214-1BA02) 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 214 und IM 208DP (Master)

Zählerstand: MB 0 (FFh ... 00h)  
 Profibus-Adresse: 2

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

Profibus-Adresse: 3  
 Ausgangsbereich: Adresse 0, Länge: 1Byte

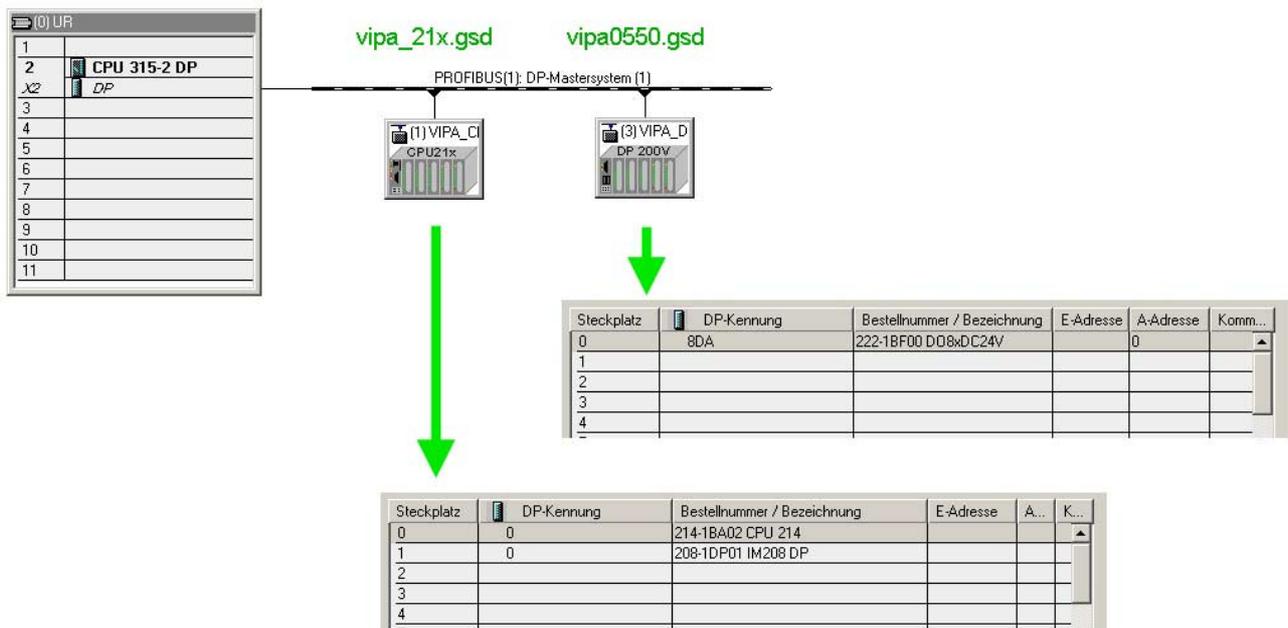
**Projektierung IM 208DP**

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind für das System 200V folgende Schritte durchzuführen:

- Starten Sie den Hardware-Konfigurator von Siemens
- Installieren Sie die GSD-Datei vipa\_21x.gsd
- Projektieren Sie eine CPU 315-2DP mit DP-Master (Adresse 2)
- Fügen Sie einen Profibus-Slave "VIPA\_CPU21x" mit Adresse 1 an
- Binden Sie auf Steckplatz 0 des Slave-Systems die CPU **214-1BA02** ein
- Binden Sie auf Steckplatz 1 den DP-Master 208-1DP01 ein

Zur Ankopplung des IM 253DP sind, nachdem Sie die GSD-Datei vipa0550.gsd eingebunden haben, folgende Schritte erforderlich:

- Fügen Sie den Profibus-Slave "VIPA\_DP200V\_2" mit Adresse 3 an. Sie finden den DP-Slave im Hardware-Katalog unter:  
*Profibus-DP > Weitere Feldgeräte > I/O > VIPA\_System200V*
- 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:

- Verbinden Sie hierzu Ihr PG bzw. Ihren PC über MPI mit Ihrer CPU. Sollte Ihr Programmiergerät keine MPI-Schnittstelle besitzen, können Sie für eine serielle Punkt-zu-Punkt-Übertragung von Ihrem PC an MPI das "Green Cable" von VIPA verwenden.  
Das "Green Cable" hat die Best.-Nr. VIPA 950-0KB00 und darf nur bei den VIPA CPUs der Systeme 100V, 200V, 300V und 500V eingesetzt werden. Für den Einsatz sind folgende Einstellungen erforderlich:
  - Wählen Sie in Ihrem Projektiertool unter **Extras** > *PG/PC-Schnittstelle einstellen* die Schnittstellenparametrierung "PC Adapter (MPI)" aus, ggf. müssen Sie diesen erst hinzufügen.
  - Klicken auf [Eigenschaften] und stellen Sie unter "Lokaler Anschluss" den gewünschten COM-Port und die Baudrate 38400 ein.
- Konfigurieren Sie die MPI-Schnittstelle Ihres PC.
- Mit **Zielsystem** > *Laden in Baugruppe* in Ihrem Projektiertool übertragen Sie Ihr Projekt in die CPU.
- Zur zusätzlichen Sicherung Ihres Projekts auf MMC stecken Sie eine MMC und übertragen Sie mit **Zielsystem** > *RAM nach ROM kopieren* Ihr Anwenderprogramm auf die MMC.  
Während des Schreibvorgangs blinkt die "MC"-LED auf der CPU. Systembedingt wird zu früh ein erfolgter Schreibvorgang gemeldet. Der Schreibvorgang ist erst beendet, wenn die LED erlischt.

Sobald sich die CPU und DP-Master im RUN befinden, werden die Zählerstände über Profibus übertragen und auf dem Ausgabemodul des DP-Slave ausgegeben.

**Beispiel 2**

**Aufgabenstellung**

In diesem Beispiel soll eine Kommunikation zwischen einer CPU 21x (hier CPU 214-1BA02) mit IM 208 DP-Master und einer CPU 21xDP (hier CPU 214-2BP02) gezeigt werden.

Hierbei sollen Zählerstände über den Profibus ausgetauscht und diese auf dem Ausgabe-Modul des jeweiligen Partners dargestellt werden.

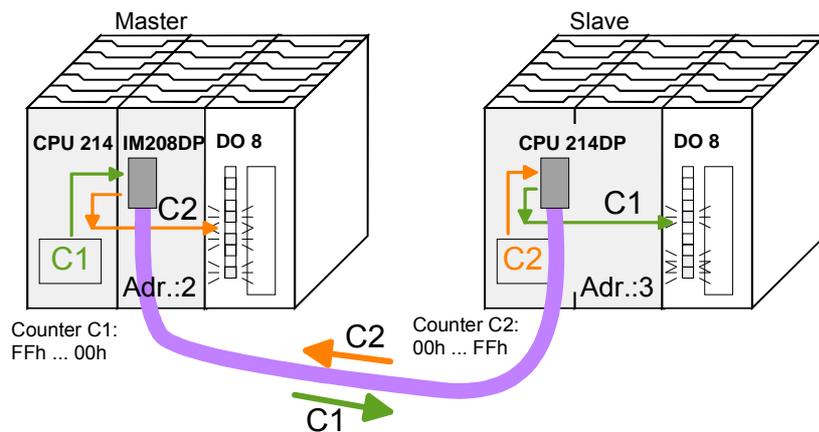
**Aufgabenstellung im Detail**

Die CPU 214 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 214DP 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 214DP 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 214 auszugeben.



**Projektierdaten**

**CPU 214 und DP-Master**

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

**CPU 214DP**

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 214 des DP-Masters**

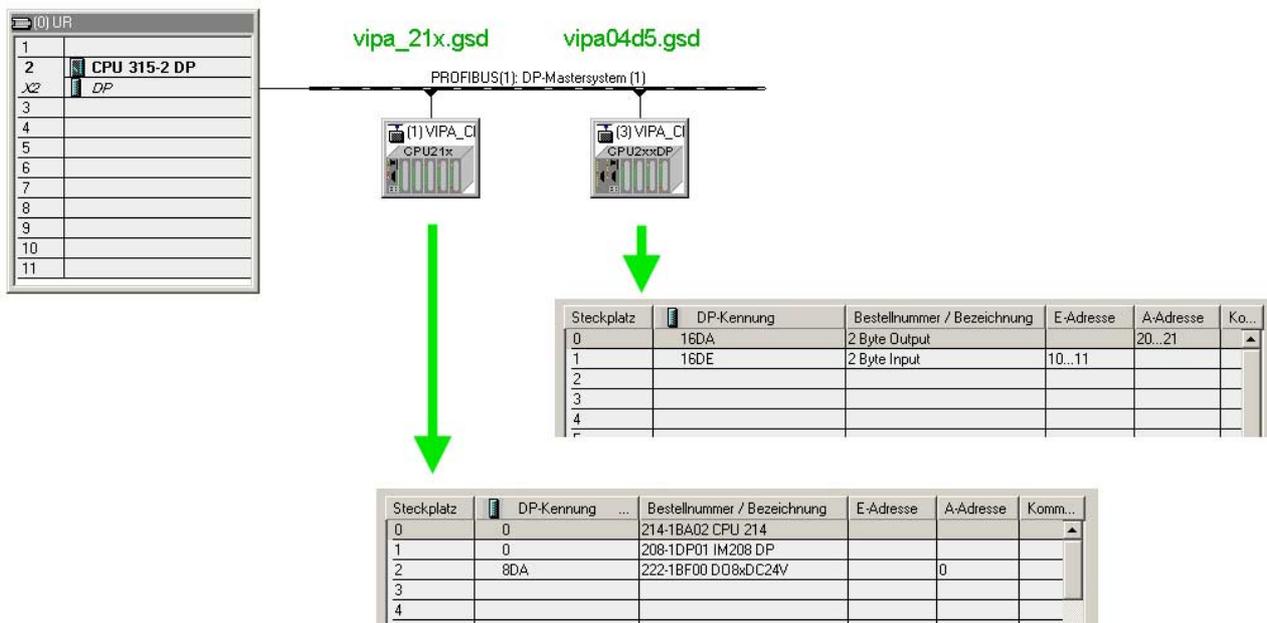
Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind für die CPU 214 und DP-Master folgende Schritte durchzuführen:

- Starten Sie den Hardware-Konfigurator von Siemens
- Installieren Sie die GSD-Datei vipa\_21x.gsd
- Projektieren Sie eine CPU 315-2DP mit DP-Master (Adresse 2)
- Fügen Sie einen Profibus-Slave "VIPA\_CPU21x" mit Adresse 1 an
- Binden Sie auf Steckplatz 0 des Slave-Systems die CPU **214-1BA02** ein
- Binden Sie auf Steckplatz 1 den DP-Master 208-1DP01 (Platzhalter) und auf Steckplatz 2 das Digitale Ausgabe-Modul 222-1BF00 ein
- Geben Sie für das Ausgabe-Modul die Ausgabe-Adresse 0 an

**Profibus-Ankopplung der CPU 214DP**

Zur Ankopplung Ihrer CPU 214DP sind, nachdem Sie die GSD-Datei vipa04d5.gsd 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



Anwenderprogramm  
für die CPU 214

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

```

Projekt trans-  
ferieren und  
ausführen

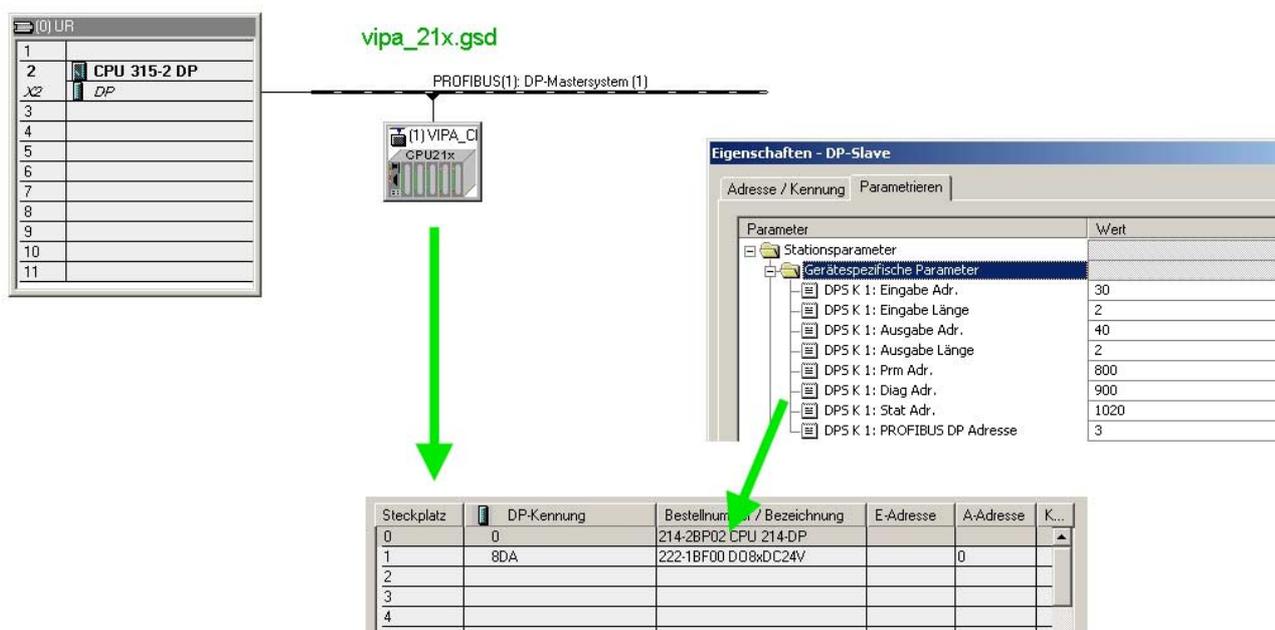
Übertragen Sie Ihr Projekt zusammen mit der Hardware-Konfiguration in die CPU und führen Sie Ihr Programm aus. Die Programmierung der CPU 214 auf der Master-Seite ist jetzt abgeschlossen.

Auf den Folgeseiten ist die Projektierung der CPU 214DP beschrieben.

## Projektierung CPU 214DP

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind für die CPU 214DP folgende Schritte durchzuführen:

- Starten Sie den Hardware-Konfigurator von Siemens
- Installieren Sie die GSD-Datei vipa\_21x.gsd
- Projektieren Sie eine CPU 315-2DP mit DP-Master (Adresse 2)
- Fügen Sie einen Profibus-Slave "VIPA\_CPU21x" mit Adresse 1 an
- Binden Sie auf Steckplatz 0 des Slave-Systems die CPU **214-2BP02** ein
- Geben Sie für die CPU 214DP folgende Parameter an:
  - Eingabe Adr.: 30
  - Eingabe Länge: 2
  - Ausgabe Adr.: 40
  - Ausgabe Länge: 2
  - Prm. Adr.: 800
  - Diag. Adr.: 900
  - Stat. Adr.: 1020
  - Profibus DP Adr.: 3
- Binden Sie auf Steckplatz 1 das Ausgabe-Modul 222-1BF00 ein und weisen Sie diesem die Ausgabe-Adresse 0 zu.
- Speichern Sie Ihr Projekt.



Anwenderprogramm in CPU 214DP

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 1020      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
    
```

Projekt transferieren und ausführen

Übertragen Sie Ihr Projekt zusammen mit der Hardware-Konfiguration in die CPU (siehe Beispiel 1) und führen Sie Ihr Programm aus.

Sobald sich beide CPUs und DP-Master im RUN befinden, werden die Zählerstände über Profibus übertragen und auf dem jeweiligen Ausgabemodul ausgegeben.

## Technische Daten

### Profibus-DP- Master

#### IM 208DP

Elektrische Daten	VIPA 208-1DP01
Spannungsversorgung	über Rückwandbus
Stromaufnahme	max. 450mA
Verlustleistung	2 W
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
Gesamtlänge	ohne Repeater 100m bei 12Mbaude; mit Repeater bis 1000m
max. Teilnehmeranzahl	32 Stationen (ohne Repeater), auf 126 erweiterbar (mit Repeater).
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	25,4x76x78
Gewicht	110g

**IM 208DPO**

Elektrische Daten	VIPA 208-1DP11
Spannungsversorgung	über Rückwandbus
Stromaufnahme	max. 450mA
Verlustleistung	2 W
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	25,4x76x78
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

### IM 253DP

Elektrische Daten	VIPA 253-1DP00 (DP-V0)	VIPA 253-1DP01 (DP-V0/V1)	VIPA 253-1DP31 - ECO (DP-V0/V1)
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil		
Stromaufnahme	max. 1A		max. 0,3A
Ausgangsstrom Rückwandbus	max. 3,5A		max. 0,8A
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		
Medium	Abgeschirmtes verdrilltes Twisted Pair-Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen.		
Übertragungsrate	9,6kBaude bis 12MBaude		
Gesamtlänge	ohne Repeater 100m (12MBaude), mit Repeater bis 1000m		
max. Teilnehmeranzahl	32 Stationen (ohne Repeater), auf 126 erweiterbar (mit Repeater).		
Diagnosefunktionen			
Standard-Diagnose	Speicherung der letzten 100 Diagnosen im Flash-ROM		
Erweiterte Diagnosemöglichkeit	-	möglich	
Daten			
Eingabedaten	max. 152Byte	max. 244Byte	
Ausgabedaten	max. 152Byte	max. 244Byte	
Kombination mit Peripheriemodulen			
max. Modulanzahl*	32	8	
max. digital	32	8	
max. analog	16	8	
Maße und Gewicht			
Abmessungen (BxHxT) in mm	25,4x76x78		
Gewicht	80g		

\* abhängig von der Stromaufnahme

**IM 253DPO**

Elektrische Daten	VIPA 253-1DP10 (DP-V0)	VIPA 253-1DP11 (DP-V0/V1)
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil	
Stromaufnahme (Leerlauf)	50mA	
Stromaufnahme (Nennwert)	max. 1A	
Ausgangsstrom Rückwandbus	max. 3,5A	
Verlustleistung	2,5W	
Potenzialtrennung	≥ AC 500V	
Statusanzeige	über LEDs auf der Frontseite	
Anschlüsse/Schnittstellen	4polige Lichtwellenleiter-Buchse	Profibus-Ankopplung
<b>Profibus Schnittstelle</b>		
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)	
<b>Diagnosefunktionen</b>		
Standard-Diagnose	Speicherung der letzten 100 Diagnosen im Flash-ROM.	
Erweiterte Diagnosemöglichkeit	nein	möglich
<b>Daten</b>		
Eingabedaten	max. 152Byte	max. 244Byte
Ausgabedaten	max. 152Byte	max. 244Byte
<b>Kombination mit Peripheriemodulen</b>		
max. Modulanzahl	32 (abhängig von der Stromaufnahme)	
max. digital	32	
max. analog	16	
<b>Maße und Gewicht</b>		
Abmessungen (BxHxT) in mm	25,4x76x78	
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 DP-V0

Elektrische Daten	VIPA 253-2DP50
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil
Stromaufnahme (Leerlauf)	50mA
Stromaufnahme (Nennwert)	max. 1A
Ausgangsstrom Rückwandbus	max. 3,5A
Verlustleistung	2,5W
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
Medium	Abgeschirmtes verdrilltes Twisted Pair-Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen.
Übertragungsrate	9,6kBaude bis 12MBaude
Gesamtlänge	ohne Repeater 100m (12MBaude), mit Repeater bis 1000m
max. Teilnehmeranzahl	32 Stationen (ohne Repeater), auf 126 erweiterbar (mit Repeater).
Diagnosefunktionen	
Standard-Diagnose	Speicherung der letzten 100 Diagnosen im Flash-ROM.
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	50,8x76x78
Gewicht	120g

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

**IM 253DP**  
**DO 24xDC 24V**  
**DP-V0**

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	9,6kBaude bis 12MBaude
Gesamtlänge	ohne Repeater 100m (12MBaude), mit Repeater bis 1000m
max. Teilnehmeranzahl	32 Stationen (ohne Repeater), auf 126 erweiterbar (mit Repeater).
Diagnosefunktionen	
Standard-Diagnose	Speicherung der letzten 100 Diagnosen im Flash-ROM.
Erweiterte Diagnosemöglichkeit	-
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 (20,4...28,8V) 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	50,8x76x78
Gewicht	150g

## Teil 4 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:

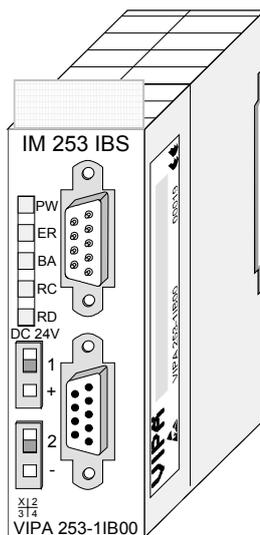
- Systemübersicht und Interbus-Grundlagen
- Hardwareaufbau, Einsatz und Inbetriebnahme des Interbus-Kopplers
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 4 Interbus</b> .....	<b>4-1</b>
Systemübersicht.....	4-2
Grundlagen .....	4-3
IM 253IBS - Interbus-Koppler - Aufbau.....	4-7
Anschluss an Interbus .....	4-10
Einsatz im Interbus.....	4-11
Inbetriebnahme .....	4-15
Technische Daten .....	4-18

## 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 253-11B00	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.

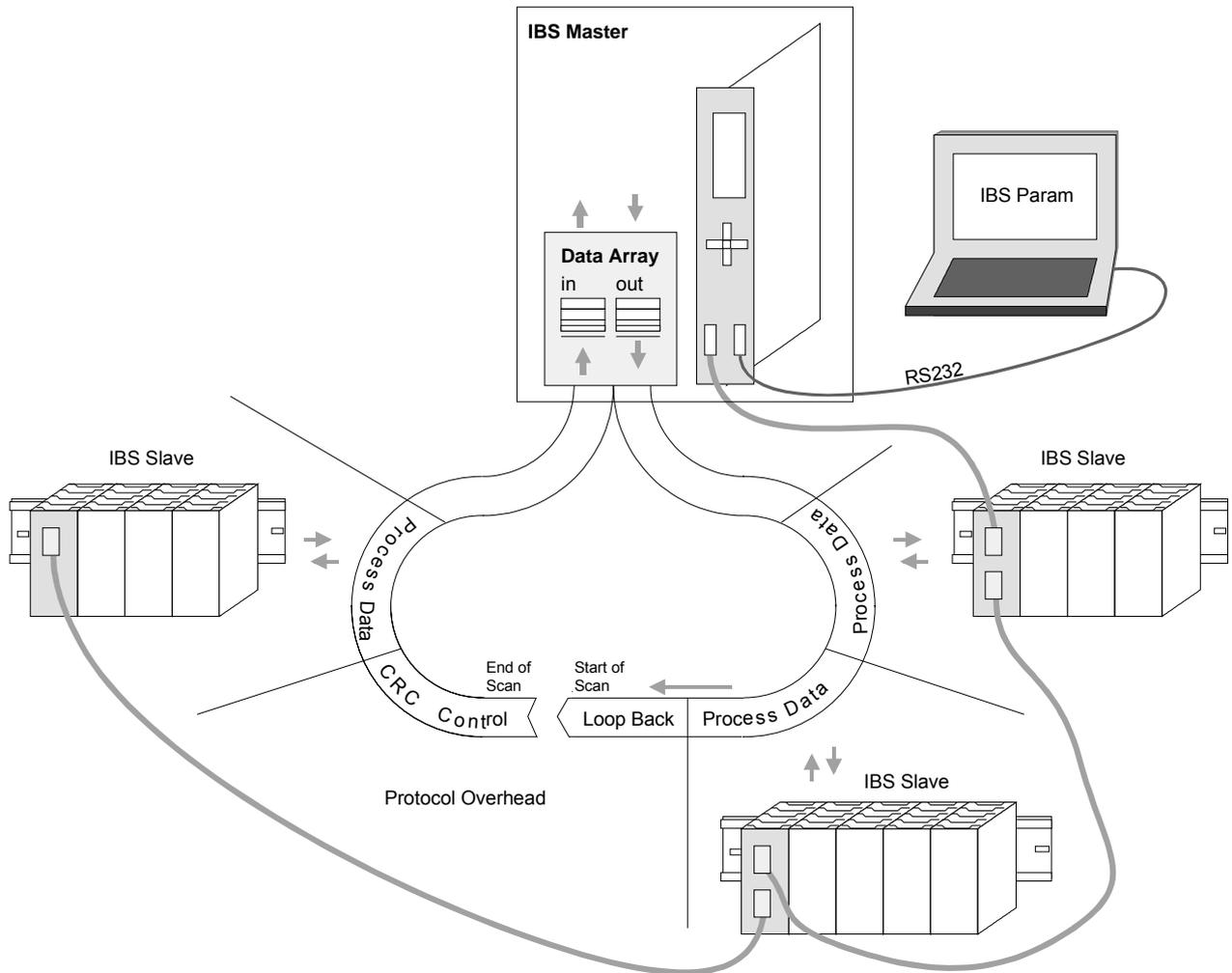
<b>Beschränkung der Datenbreite</b>	<p>Mit zunehmender Datenbreite steigt der Hardwareaufwand für einen Interbus-Teilnehmer. Aus diesem Grund wurde die Datenbreite auf max. 20Byte Ein- und 20Byte Ausgangsdaten beschränkt.</p> <p>Untergeordnete Interbus-Segmente (Peripheriebus) können über die zugehörigen 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 500kBaude 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.

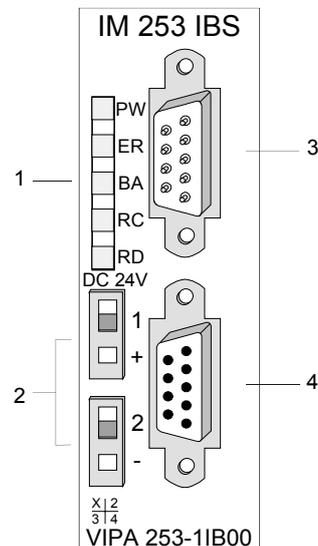
Nachdem so das gesamte Summenrahmentelegramm ausgegeben und gleichzeitig wieder eingelesen wurde, sind alle Ausgabedaten in den Schieberegistern der einzelnen Teilnehmer richtig positioniert. Über ein spezielles Steuerkommando teilt der Master den Teilnehmern das Ende des Übertragungszyklus mit.

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:

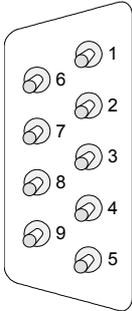
Bez.	Farbe	Bedeutung
PW	Grün	Power-LED Signalisiert eine anliegende Betriebsspannung.
ER	Rot	Error Fehler in der Applikation.
BA	Grün	Bus aktiv Mit Hilfe der BA-LED (bus active) wird ein Datentransfer über Interbus angezeigt.
RC	Grün	Remotebus Check Über die RC LED (Remotebus Check) wird angezeigt, ob 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:

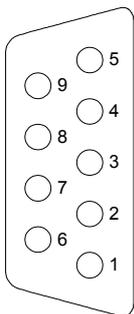
### Ankommende Busleitung (9pol SubD-Stecker)



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.

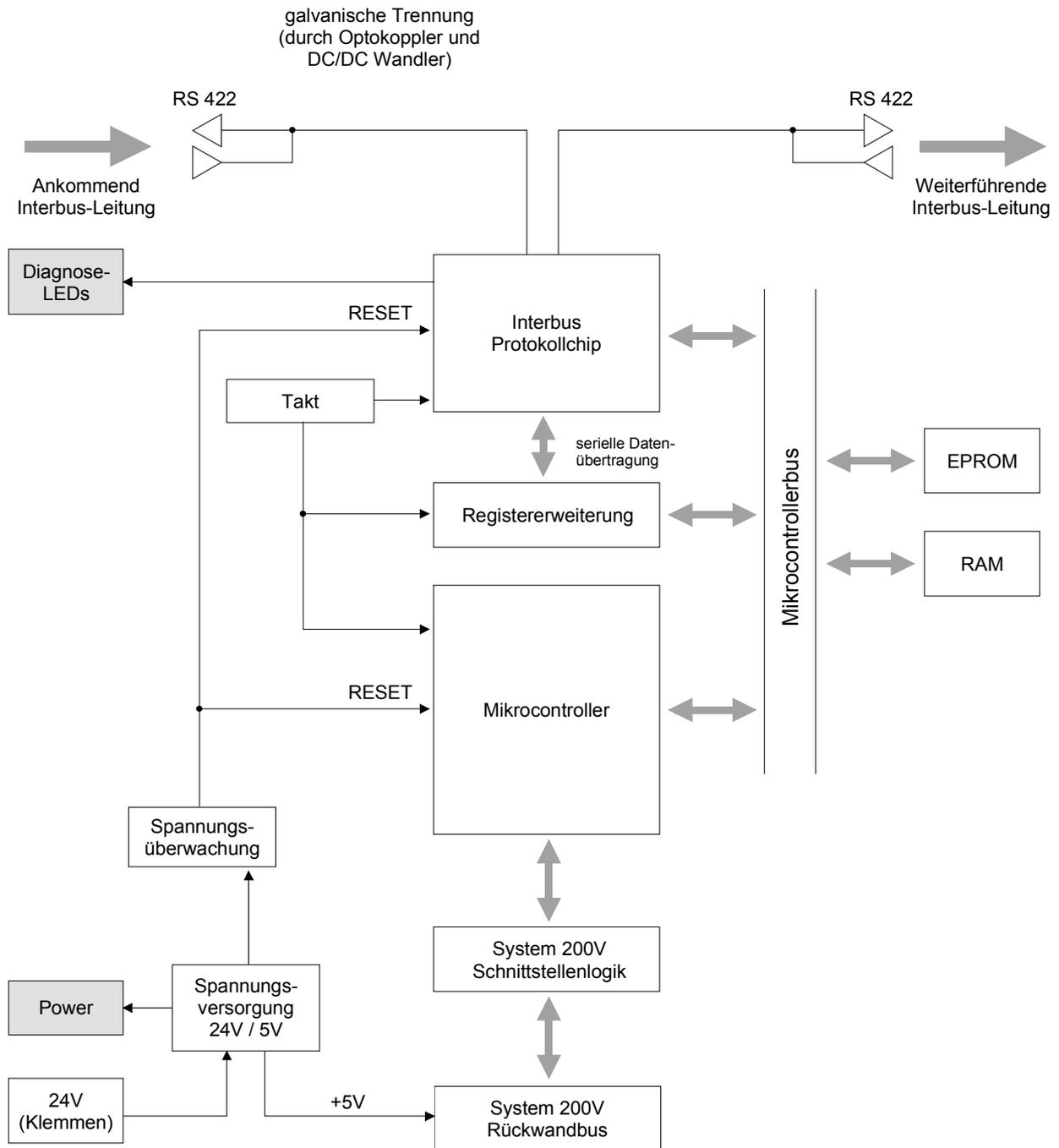
Interbus und Rückwandbus sind galvanisch voneinander getrennt.



### Hinweis!

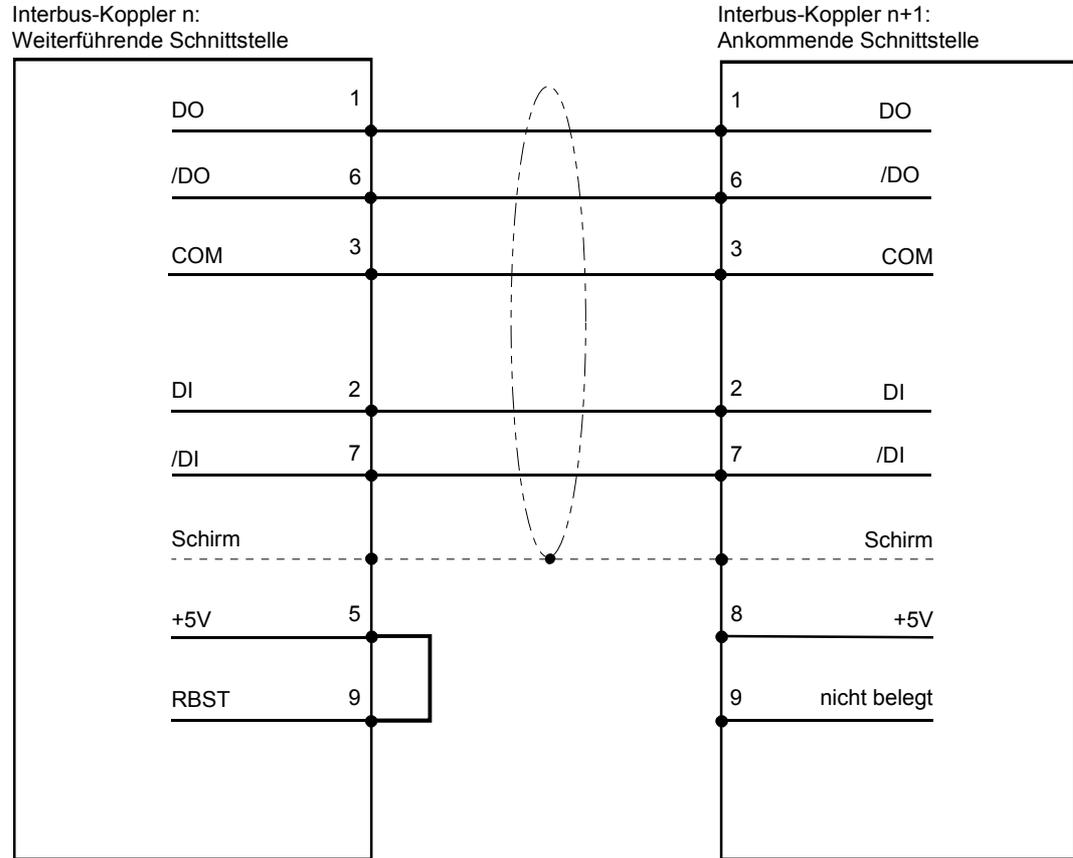
Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung!

**Blockschaltbild** Das nachfolgende Blockschaltbild zeigt den prinzipiellen Hardwareaufbau des Buskopplers:



## Anschluss an Interbus

### Verkabelung unter Interbus



#### Potenzialtrennung

Da Interbus-Fernbussegmente eine große räumliche Ausdehnung erreichen, müssen die einzelnen Segmente zur Vermeidung einer Potenzialverschleppung 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 Potenzial 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

## Prozessabbild

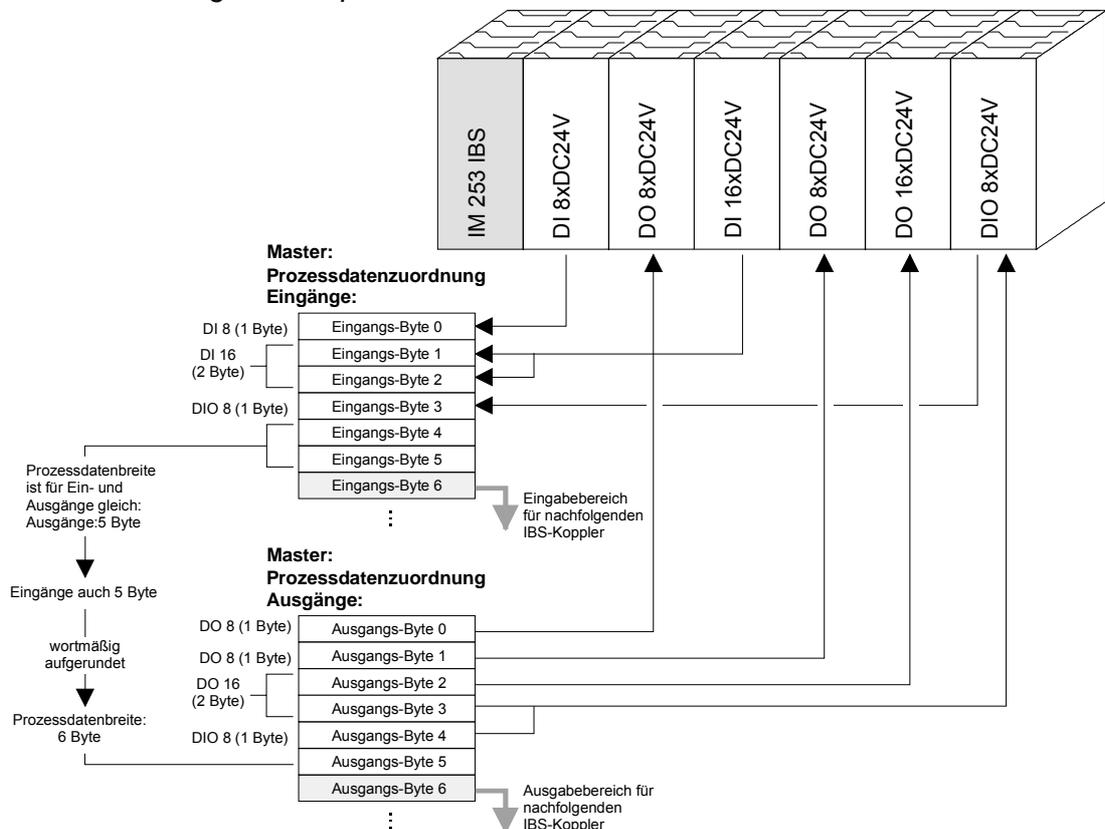
Nach dem Einschalten ermittelt der Buskoppler die Konfiguration der gesteckten Module und trägt diese in ein internes Prozessabbild ein. Dieses Prozessabbild schickt er an den Master. Der Master erstellt aus den Prozessabbildern eine Prozessdatenliste aller am Bus befindlichen Koppler. Die Prozessdatenzuordnungsliste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

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

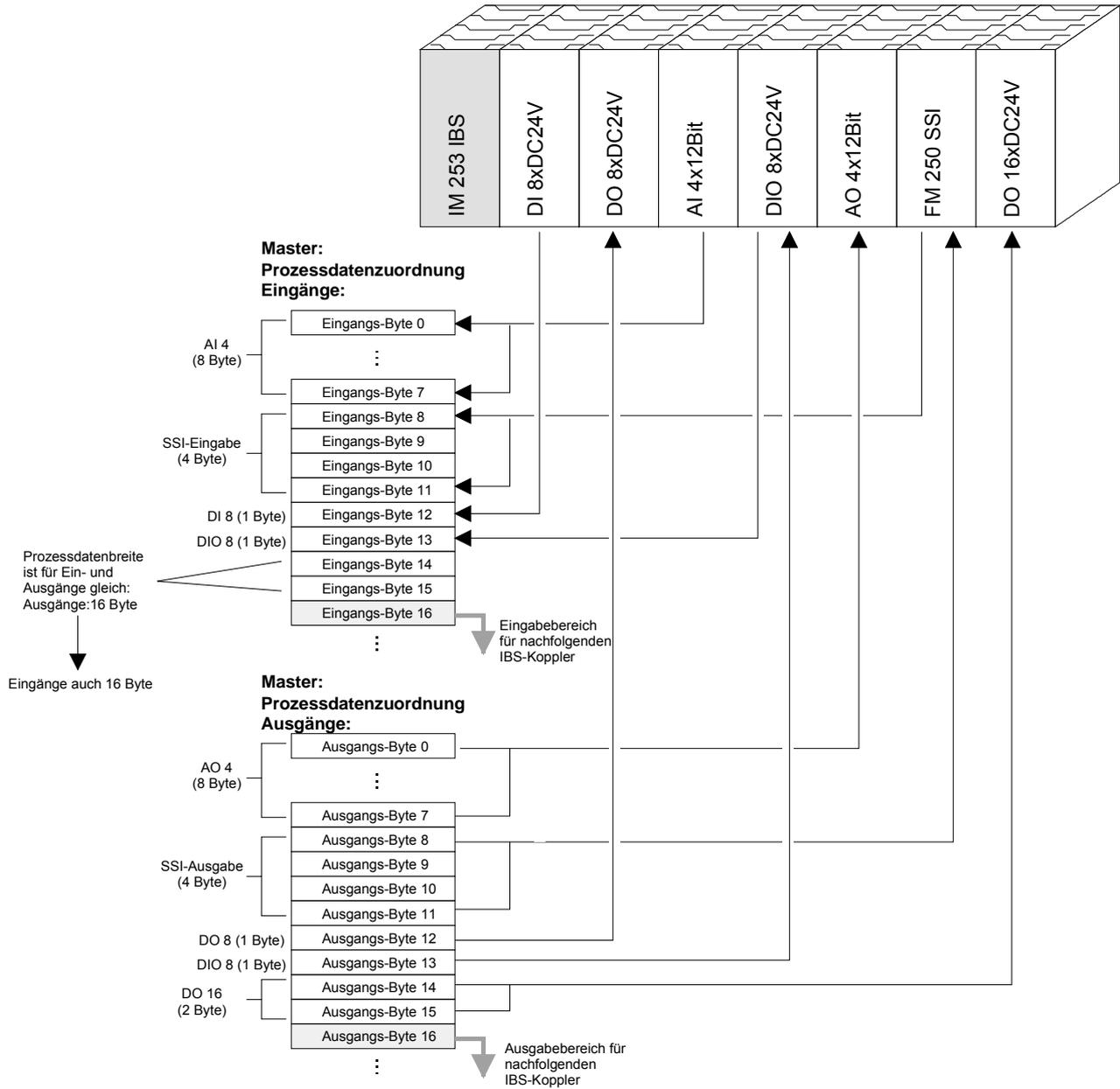
- Digitale Signale sind bitorientiert d.h. jedem Kanal ist ein Bit im Prozessabbild zugeordnet.
- Es gibt getrennte Bereiche für Ein- und Ausgangs-Daten.
- In den Ein- bzw. Ausgangsbereichen kommen an den Anfang immer die nicht digitalen Module und dann die digitalen Module.
- Die Reihenfolge der Zuweisung 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 wortmäÙig 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

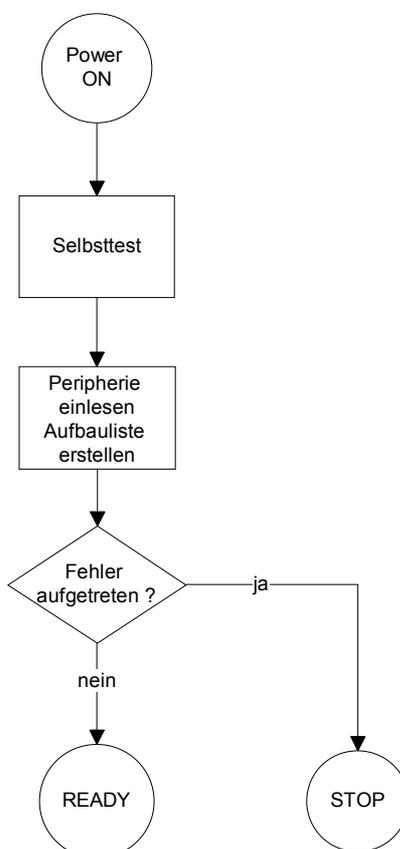
- Bauen Sie Ihren Interbus-Koppler mit den entsprechenden Modulen auf.
- Projektieren Sie den Interbus-Koppler mit dem mit dem Master mitgelieferten Projektierool.
- Schließen Sie das Interbus-Kabel am Koppler an und schalten Sie die Spannungsversorgung ein.

### 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.

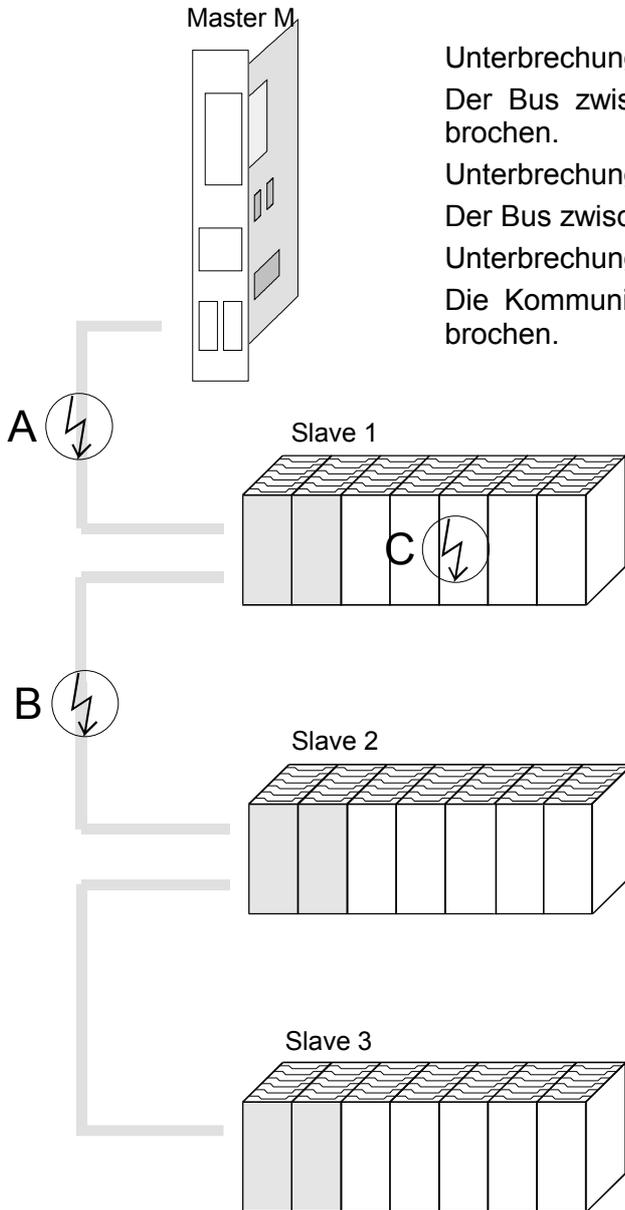
Nun erfolgt das Einlesen des Peripherieaufbaus. Zuerst wird die Anzahl der gesteckten Module eingelesen, anschließend werden die Module anhand ihrer Typkennungen identifiziert. Mit dem registrierten Peripherieaufbau werden Steckplatzkennungen für die Module generiert und über den Rückwandbus in die Module 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 (im Leerlauf)	50mA
Stromaufnahme (Nennwert)	800mA
Ausgangsstrom Rückwandbus	max. 3,5A
Verlustleistung	2W
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	25,4x76x78
Gewicht	80g

## Teil 5 CANopen

### Überblick

Inhalt dieses Kapitels ist die Beschreibung der CANopen-Master/Slave Module 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-Master/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

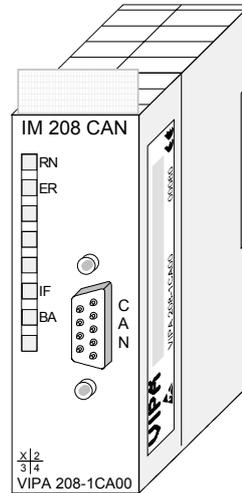
### Inhalt

Thema	Seite
<b>Teil 5 CANopen .....</b>	<b>5-1</b>
Systemübersicht.....	5-2
Grundlagen .....	5-4
IM 208CAN - CANopen-Master - Aufbau.....	5-6
IM 208CAN - CANopen-Master - Projektierung .....	5-8
IM 208CAN - CANopen-Master - Firmwareupdate .....	5-15
IM 208CAN - CANopen-Master - Betriebsarten .....	5-16
IM 208CAN - CANopen-Master - Prozessabbild .....	5-17
IM 208CAN - CANopen-Master - Telegrammaufbau .....	5-19
IM 208CAN - CANopen-Master - Objekt-Verzeichnis .....	5-24
IM 253CAN - CANopen-Slave - Aufbau.....	5-38
IM 253CAN - CANopen-Slave DO 24xDC 24V - Aufbau .....	5-43
IM 253CAN - CANopen-Slave - Schnelleinstieg .....	5-47
IM 253CAN - CANopen-Slave - Baudrate und Modul-ID .....	5-51
IM 253CAN - CANopen-Slave - Telegrammaufbau .....	5-52
IM 253CAN - CANopen-Slave - PDO .....	5-54
IM 253CAN - CANopen-Slave - SDO .....	5-58
IM 253CAN - CANopen-Slave - Objekt-Verzeichnis .....	5-60
IM 253CAN - CANopen-Slave - Emergency Object.....	5-101
IM 253CAN - CANopen-Slave - NMT - Netzwerk Management.....	5-102
Technische Daten .....	5-104

# Systemübersicht

## CANopen-Master IM 208CAN

Folgender CANopen-Master ist von VIPA verfügbar:

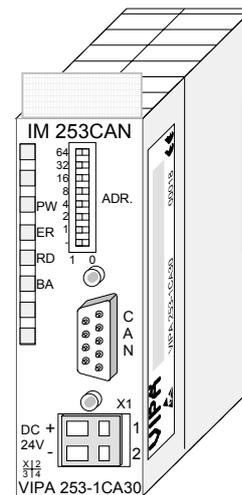
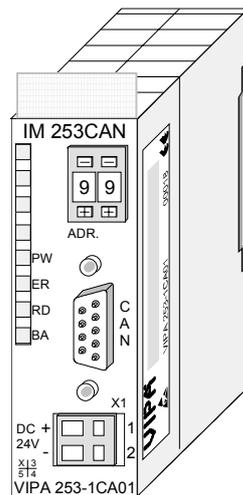


### Bestelldaten

Typ	Bestellnummer	Beschreibung	Seite
IM 208CAN	VIPA 208-1CA00	CAN-Bus CANopen-Master 1Mbaud, bis zu 125 Slaves	5-6

## CANopen-Slave IM 253CAN

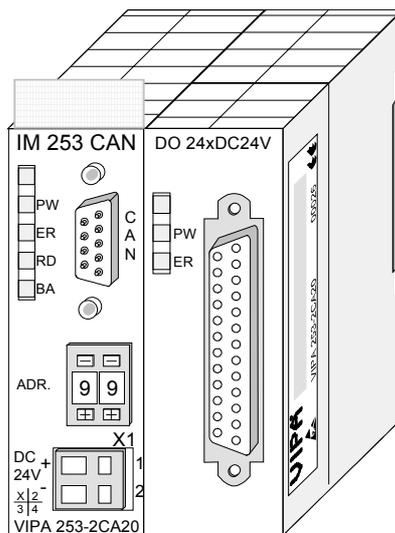
Folgende CAN-Bus-Koppler sind von VIPA verfügbar:



### Bestelldaten

Typ	Bestellnummer	Beschreibung	Seite
IM 253CAN	VIPA 253-1CA01	CAN-Bus CANopen-Slave	5-38
IM 253CAN	VIPA 253-1CA30	CAN-Bus CANopen-Slave - ECO	5-38

**CANopen-Slave  
(Kombimodule)**



Bestelldaten

Typ	Bestellnummer	Beschreibung	Seite
IM 253CAN DO 24xDC24V	VIPA 253-2CA20	CAN-Bus CANopen slave mit DO 24xDC 24V	5-43

## 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 **a**pplication **l**ayer). 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.

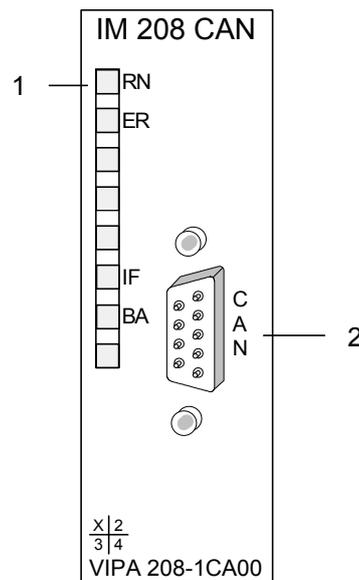
Die Buszugriffssteuerung bei CAN geschieht mit Hilfe der zerstörungsfreien, bitweisen Arbitrierung. Hierbei bedeutet zerstörungsfrei, dass der Gewinner der Arbitrierung sein Telegramm nicht erneut senden muss. Beim gleichzeitigen Mehrfachzugriff von Teilnehmern auf den Bus wird automatisch der wichtigste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

## IM 208CAN - CANopen-Master - Aufbau

### Eigenschaften

- 125 CAN-Slaves an einen CANopen-Master angeschlossen
- Projektierung unter WinCoCT von VIPA
- Diagnosefähig
- 40 Transmit PDOs
- 40 Receive PDOs
- PDO-Linking
- PDO-Mapping
- 1 SDO als Server, 127 SDO als Client
- Emergency Object
- NMT Object
- Node Guarding, Heartbeat
- Ein-/Ausgabe-Bereich 0x6xxx je maximal 64Bytes
- Ein-/Ausgabe-Bereich 0xAxxx je maximal 320 Bytes

### Frontansicht IM 208CAN



- [1] LED Statusanzeigen
- [2] CAN-Schnittstelle

**Komponenten**

**LEDs**

Das CANopen-Master-Modul 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
RN	Grün	AN: CPU befindet sich im RUN AUS: CPU befindet sich im STOP
ER	Rot	AN: Während der Initialisierung und bei Slave-Ausfall AUS: Alle Slaves befinden sich im Zustand "operational"
BA	Gelb	BA ( <b>B</b> us <b>a</b> ktiv) zeigt Kommunikation über CAN-Bus an. AN: Zustand "operational" Blinkt mit 1Hz: zeigt Zustand "pre-operational".
IF	Rot	AN: Initialisierungsfehler bei fehlerhafter Parametrierung. AUS: Initialisierung ist OK.

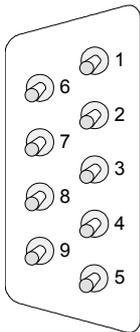


**Hinweis!**

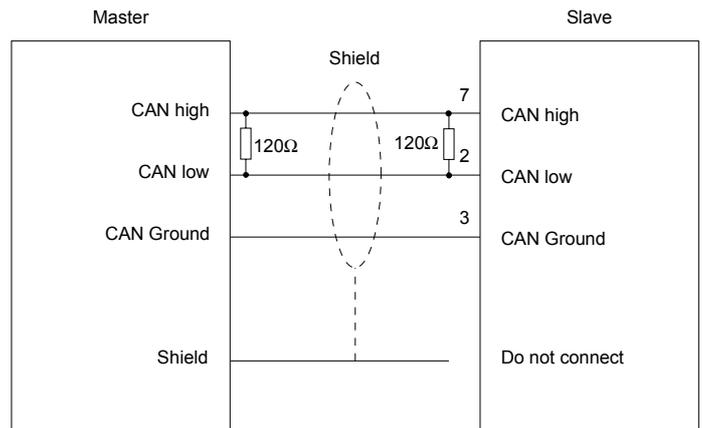
Blinken alle LEDs mit 1Hz, erwartet der CAN-Master gültige Parameter von der CPU. Bekommt der CAN-Master keine Parameter von der CPU, gehen nach 5sec alle LEDs aus.

**CAN-Schnittstelle**

Über die 9polige CAN-Schnittstelle binden Sie den CANopen-Master in Ihren CAN-Bus ein. Die Schnittstelle hat folgende Pinbelegung:



Pin	Belegung
1	reserviert
2	CAN low
3	CAN Ground
4	reserviert
5	Schirm
6	Masse 24V
7	CAN high
8	reserviert
9	+DC 24V



**Hinweis!**

Beachten Sie, dass beide Busenden mit einem 120Ω-Abschlusswiderstand abzuschließen sind!

**Spannungsversorgung**

Der CANopen-Master bezieht seine Spannungsversorgung über den Rückwandbus.

## IM 208CAN - CANopen-Master - Projektierung

Die Projektierung des CANopen-Masters erfolgt unter WinCoCT (**Windows CANopen Configuration Tool**) von VIPA. Aus WinCoCT exportieren Sie Ihr Projekt als wld-Datei. Die wld-Datei können Sie in Ihren Hardware-Konfigurator von Siemens importieren.

### Schnelleinstieg

Für den Einsatz von System 200V Modulen und des CAN-Masters ist die Einbindung der System 200V Module über die GSD-Datei von VIPA im Hardwarekatalog erforderlich. Zur Projektierung im Hardware-Konfigurator sind folgende Schritte durchzuführen:

- WinCoCT starten und CANopen-Netzwerk projektieren
- Hierzu mit  eine "Master"-Gruppe anlegen und mit  einen CANopen-Master einfügen.
- Über "Device Access" mit "Device is NMT Master" die Master-Funktion aktivieren.
- Aktivieren Sie im Register "CANopen Manager" Device is NMT Master und bestätigen Sie Ihre Eingabe.
- Mit "Set PLC Parameters" Parameter vorgeben wie Diagnose-Verhalten und CPU-Adress-Bereiche.
- Eine "Slave"-Gruppe mit  anlegen und mit  Ihre CANopen-Slaves hinzufügen.
- Den Slaves über "Module" Module hinzu und ggf. parametrieren
- Unter "Connections" Prozessdatenverbindungen in der Matrix einstellen. ggf. Eingabe im Prozessabbild des Master überprüfen.
- Projekt speichern und als wld-Datei exportieren.
- In den SIMATIC-Manager von Siemens wechseln und Datenbaustein von CAN-wld-Datei in Bausteine-Verzeichnis kopieren.
- In Hardware-Konfigurator Profibus-DP-Master-System mit folgender Siemens-CPU projektieren: CPU 315-2DP (6ES7 315-2AF03-0AB0)
- DP-Master bekommt Adresse >1.
- An Master-System aus dem Hardware-Katalog das System 200V DP-Slave-System anbinden.
- Das System 200V DP-Slave-System bekommt immer die Adresse 1.
- Alles speichern und SPS-Projekt via MPI zusammen mit der wld-Datei in die CPU übertragen.

Nachfolgend sind diese Schritte näher erläutert.



### Hinweis!

Bitte verwenden Sie zur Projektierung der VIPA-Standard-CPU's der Systeme 100V, 200V, 300V und 500V ab der Firmware-Version 3.5.0 die CPU **6ES7-315-2AF03** V1.2 von Siemens aus dem Hardware-Katalog!

**Voraussetzungen zur Projektierung**

Der Hardware-Konfigurator ist Bestandteil des Siemens SIMATIC Managers. Er dient der Projektierung. Die Module, die hier projiziert werden können, entnehmen Sie dem Hardware-Katalog.

Für den Einsatz der System 200V Module ist die Einbindung der System 200V Module über die GSD-Datei von VIPA im Hardwarekatalog erforderlich.

**Hinweis!**

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

GSD-Datei einbinden

- Kopieren Sie die mitgelieferte VIPA-GSD-Datei VIPA\_21x.gsd 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 hier **VIPA\_21x.GSD** an

Die Module des System 200V von VIPA sind jetzt im Hardwarekatalog integriert und können projiziert werden.

**Hinweis**

Um kompatibel mit dem Siemens SIMATIC Manager zu sein, sind die System 200V CPUs von VIPA als

**CPU 315-2DP (6ES7 315-2AF03-0AB0)**

zu projektieren!

Damit die Module gezielt angesprochen werden können, sind diese im Hardware-Konfigurator von Siemens in Form eines virtuellen Profibus-Systems zu projektieren. Hierbei können Sie durch Einbindung einer GSD-Datei von VIPA auf den Funktionsumfang der Module zurückgreifen.

Die eigentliche Projektierung führen Sie mit dem CANopen-Konfigurations-Tool WinCoCT durch. Ihr Projekt können Sie in Form einer wld-Datei exportieren und als DB in Ihr SPS-Programm übernehmen.

## WinCoCT

WinCoCT (**Windows CANopen Configuration Tool**) ist ein von VIPA entwickeltes Konfigurations-Tool zur komfortablen Projektierung von CANopen-Netzwerken.

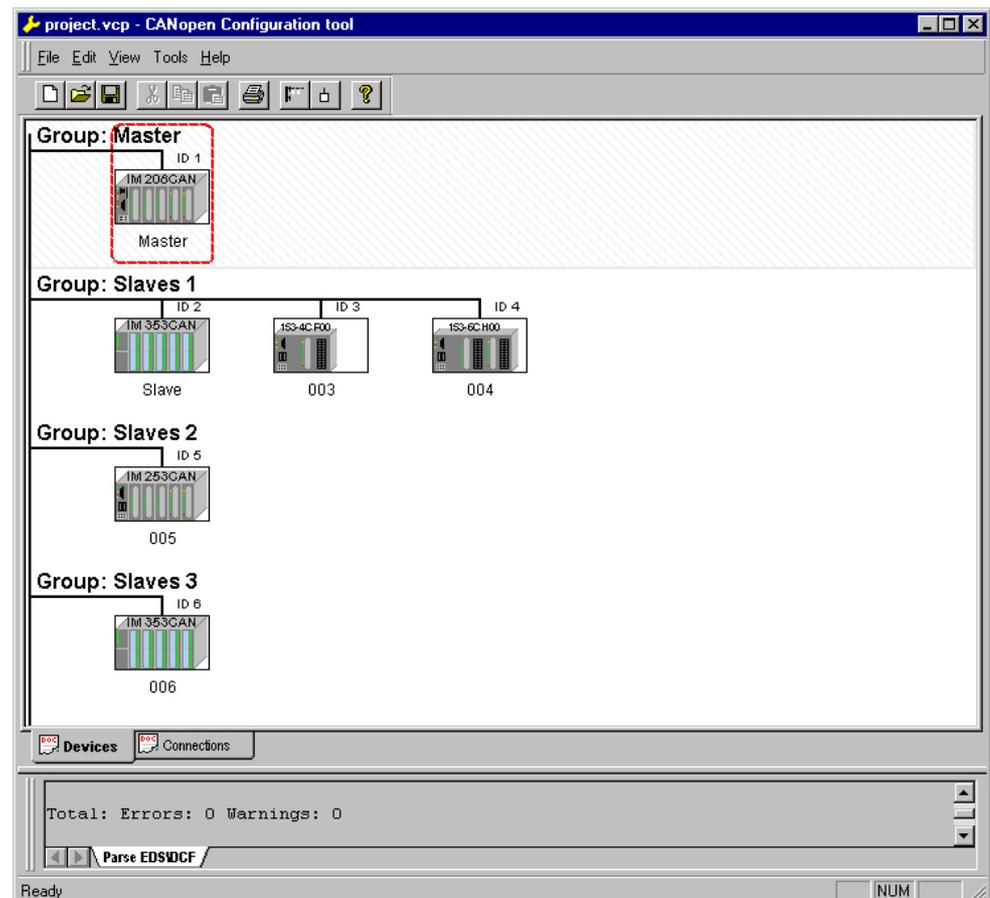
WinCoCT stellt auf einer grafischen Benutzeroberfläche die Topologie Ihres CANopen-Netzwerks dar. Hier können Sie Feldgeräte und Steuerungen platzieren, parametrieren, gruppieren und Verbindungen projektieren.

Die Auswahl der Geräte erfolgt über eine Liste, die jederzeit über eine EDS-Datei (**E**lectronic **D**ata **S**heet) beliebig erweitert werden kann.

Durch Klick mit der rechten Maustaste auf ein Gerät, erscheint für dieses Gerät ein Menü das zu einem Teil aus statischen und zum anderen Teil aus dynamischen Komponenten besteht.

Zur Konfiguration des Prozessdatenaustauschs werden alle Prozessdaten in Form einer Matrix dargestellt, wobei Geräte-Eingänge als Zeile und Geräte-Ausgänge als Spalte ausgegeben werden. Durch einfaches Markieren der Kreuzungspunkte stellen Sie die gewünschte Verbindung her.

Das Zusammenstellen und Optimieren der Telegramme führt WinCoCT selbständig durch.

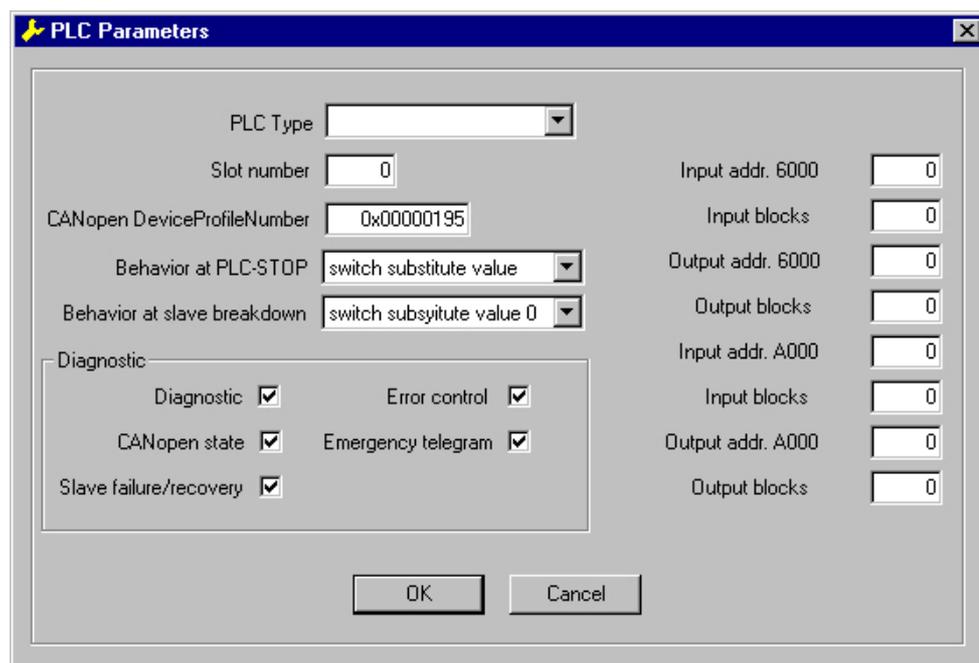


**Projektparameter einstellen**

Über **Tools** > *Project options* können Sie CAN-spezifische Parameter wie Baud-Rate, Auswahl des Masters usw. vorgeben.  
Näheres hierzu finden Sie in der Beschreibung von WinCoCT.

**Parameter CAN-Master**

Sie haben die Möglichkeit über WinCoCT VIPA-spezifische Parameter für den CAN-Master vorzugeben, indem Sie mit der rechten Maustaste auf den Master klicken und mit Set PLC-Parameters den nachfolgenden Dialog aufrufen:



PLC Type                      Reserviert für zukünftige Erweiterungen

Slot number.                 Steckplatz-Nr. auf dem Bus  
0: Zur Adressierung des in die CPU integrierten CAN-Masters  
1 ... 32: Zur Adressierung von CAN-Master am Standard-Bus

CANopen DeviceProfileNumber     Fest eingestellt auf 0x195

Behavior at PLC-STOP                Hier können Sie das Verhalten der Ausgabe-Kanäle einstellen, sobald die CPU in STOP geht. Folgende Werte stehen zur Auswahl:  
*Switch substitute value 0*: Schaltet alle Ausgänge auf 0  
*Keep last value*: Friert den aktuellen Zustand der Ausgänge ein.

Behavior at Slave breakdown	<p>Geben Sie hier an, wie die Handhabung der Slave-Eingangsdaten sein soll, wenn ein Slave ausfällt.</p> <p><i>Switch substitute value 0:</i> Die Daten werden auf 0 gesetzt.</p> <p><i>Keep the last value:</i> Die aktuellen Daten bleiben unverändert.</p>
Diagnostic	<p>In diesem Bereich können Sie das Diagnose-Verhalten des CAN-Masters einstellen.</p> <p><i>Diagnostic:</i> Aktiviert die Diagnosefunktion</p> <p><i>CANopen state:</i> Im aktivierten Zustand sendet der CAN-Master seinen Status "preoperational" oder "operational" an die CPU. Den Status können Sie über SFC 13 abrufen.</p> <p><i>Slave failure/recovery:</i> Wenn Sie diese Option aktiviert haben wird bei Slave-Ausfall und -Wiederkehr der OB 86 in der CPU aufgerufen.</p> <p><i>Error control:</i> Ist diese Option angewählt, so sendet der NMT-Master alle Guarding-Fehler als Diagnose an die CPU, die den OB 82 aufruft.</p> <p><i>Emergency Telegram:</i> Bei aktivierter Option sendet der NMT-Master alle Emergency-Telegramme als Diagnose an die CPU, die den OB 82 aufruft.</p>
Adressbereich in der CPU	<p>Über die nachfolgend aufgeführten Felder können Sie die Adressbereiche vorgeben, die der CAN-Master für seine Ein- und Ausgabe-Bereiche in der CPU belegt. Jeder Block besteht aus 4Byte.</p> <p><i>Input addr. 6000, Input blocks</i></p> <p>PE-Basis-Adresse in der CPU, die von 0x6000-CAN-Eingangsdaten belegt werden. Für Input blocks können max. 16 (64Byte) eingetragen werden.</p> <p><i>Output addr. 6000, Output blocks</i></p> <p>PA-Basis-Adresse in der CPU, die von 0x6000-CAN-Ausgangsdaten belegt werden. Für Output blocks können max. 16 (64Byte) eingetragen werden.</p> <p><i>Input addr. A000, Input blocks</i></p> <p>PE-Basis-Adresse in der CPU, die von 0xA000-CAN-Eingangs-Netzwerk-Variablen belegt werden. Für Input blocks können max. 80 (320Byte) eingetragen werden.</p> <p><i>Output addr. A000, Output blocks</i></p> <p>PA-Basis-Adresse in der CPU, die von 0xA000-CAN-Ausgangs-Netzwerk-Variablen belegt werden. Für Output blocks können max. 80 (320Byte) eingetragen werden.</p>
<b>CANopenSlave in CANopen Manager aktivieren</b>	<p>Damit eine CANopen-Slave vom Master bearbeitet werden kann, ist dieser über WinCoCT bei dem entsprechenden Master anzumelden. Klicken Sie hierzu mit der rechten Maustaste auf Ihren CAN-Master, wählen Sie "Device access" an und gehen Sie in das Register "CANopen Manager".</p> <p>Hier können Sie über [Change] jeden Slave einzeln bzw. über [Global] alle Slaves bei Ihrem Master anmelden und das Fehlerverhalten einstellen.</p> <p><b>Bitte vergessen Sie nicht, nachdem Sie Ihre Einstellungen durchgeführt haben, diese mit [Apply to slaves] in Ihre Projektierung zu übernehmen.</b></p>

**Schritte der Projektierung**

Nachfolgend wird die Vorgehensweise der Projektierung an einem abstrakten Beispiel gezeigt:

Die Projektierung gliedert sich in folgende 3 Teile:

- CAN-Master-Projektierung in WinCoCT und Export als wld-Datei
- CAN-Master-Projektierung importieren
- Projektierung der Module

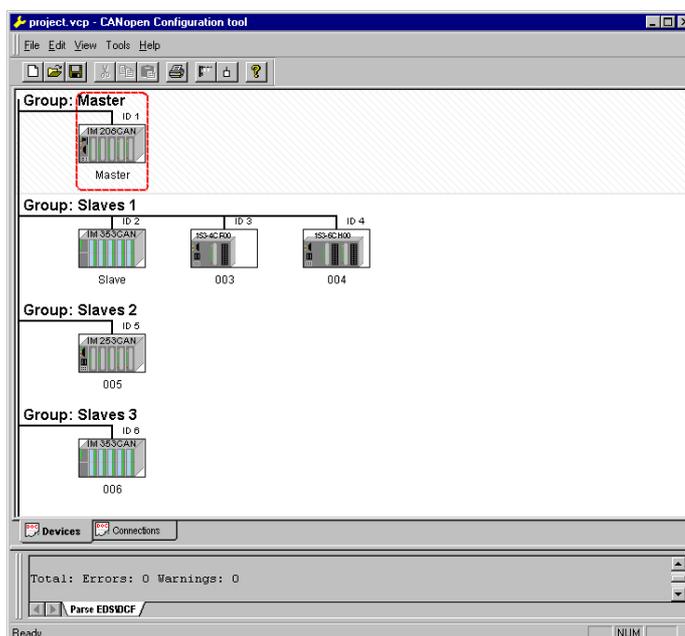
**Voraussetzungen**

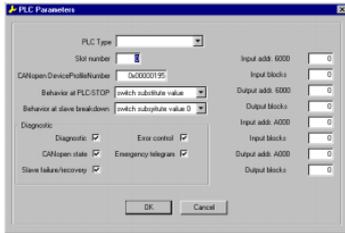
Zur Projektierung eines CANopen-Systems ist die aktuellste EDS-Datei in das EDS-Verzeichnis von WinCoCT zu übertragen.

Für den Einsatz der System 200V Module ist die Einbindung der System 200V Module über die GSD-Datei VIPA\_21x.gsd von VIPA im Hardwarekatalog erforderlich.

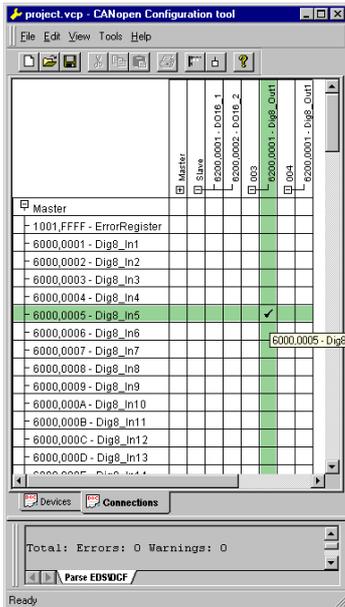
**CAN-Master-Projektierung unter WinCoCT**

- Kopieren Sie die erforderlichen EDS-Dateien in das EDS-Verzeichnis und starten Sie WinCoCT.
- Legen Sie mit  eine "Master"-Gruppe an und fügen Sie mit  einen CANopen-Master ein (VIPA\_208\_1CA00.eds).
- Legen Sie mit  eine "Slave"-Gruppe an und fügen Sie mit  Ihre CANopen-Slaves hinzu.
- Klicken mit der rechten Maustaste auf den entsprechenden Slave und fügen über "Module" Sie die entsprechenden Module hinzu.
- Parametrieren Sie Ihre Module mit [Parameter] bzw. über das entsprechende Objekt-Verzeichnis
- Klicken Sie mit der rechten Maustaste auf den Master und öffnen Sie den Dialog "Device Access".
- Aktivieren Sie im Register "CANopen Manager" Device is NMT Master und melden Sie die entsprechenden Slaves beim Master an. Vergessen Sie nicht Ihre Eingaben mit [Apply to slaves] in Ihre Projektierung zu übernehmen!





- Klicken Sie mit der rechten Maustaste auf den Master und öffnen Sie den VIPA-spezifischen Dialog "Set PLC Parameters". Hier können Sie das Diagnose-Verhalten einstellen und die Adress-Bereiche vorgeben, die vom Master in der CPU belegt werden.  
Unter "Slot number" geben Sie Ihre Slot-Nr. an, auf der Ihr CAN-Master gesteckt ist. Hieraus generiert WinCoCT beim Export die entsprechende DB-Nr. + 2000.



- Wechseln Sie im Hauptfenster in das Register "Connections". Hier werden die Prozessdaten als Eingänge (1. Spalte) und als Ausgänge (1. Zeile) in einer Matrix dargestellt.  
Zur Anzeige der Prozessdaten eines Geräts, dem ein "+" vorangestellt ist, klicken Sie auf das entsprechende Gerät.
- Zu Ihrer Hilfe können Sie immer nur dann eine Verbindung definieren, wenn das Fadenkreuz grün erscheint. Stellen Sie mit der Maus in Zeile und Spalte der Matrix die entsprechende Zelle ein und klicken Sie mit der linken Maustaste → die Zelle wird mit einem "☑" gekennzeichnet. Sie können die projektierte Verbindung überprüfen, indem Sie wieder in "Devices" wechseln, auf den Master klicken und über "Device Access" das Prozessabbild des Masters ausgeben.
- Speichern Sie Ihr Projekt.
- Über **File** > **Export** wird Ihr CANopen-Projekt in eine wld-Datei exportiert. Der Name setzt sich zusammen aus Projektname + Knotenadresse + Kennung **Master/Slave**.

Hiermit ist die CANopen-Projektierung unter WinCoCT abgeschlossen.

### Import in SPS-Programm und Transfer in CAN-Master

- Starten Sie den SIMATIC-Manager von Siemens mit Ihrem SPS-Projekt und öffnen Sie mit **Datei** > **Memory Card Datei** > **öffnen** die wld-Datei.
- Kopiere den DB 2xxx in Ihr Bausteine-Verzeichnis.
- Starten Sie den Hardware-Konfigurator von Siemens mit einem neuen Projekt und fügen Sie aus dem Hardware-Katalog eine Profilschiene ein.
- Platzieren Sie auf Steckplatz 2 folgende Siemens CPU:  
CPU 315-2DP (6ES7 315-2AF03-0AB0). Bitte verwenden Sie zur Projektierung der VIPA-Standard-CPU's der Systeme 100V, 200V, 300V und 500V ab der Firmware-Version 3.5.0 die CPU **6ES7-315-2AF03 V1.2** von Siemens aus dem Hardware-Katalog!
- Befindet sich beispielsweise Ihr CAN-Master-Modul direkt neben der CPU, so projektieren Sie auf Steckplatz 4 Ihren CAN-Master.
- Beginnend mit Steckplatz 5 binden Sie Ihre System 200V Module auf dem Standard-Bus in der gesteckten Reihenfolge ein.
- Parametrieren Sie ggf. CPU bzw. die Module. Das Parameterfenster wird geöffnet, sobald Sie auf das entsprechende Modul doppelklicken.
- Sichern Sie Ihr Projekt und übertragen Sie dies in Ihre CPU.

Nach dem Übertragen erkennt die CPU den DB für den CAN-Master und leitet beim STOP-RUN-Übergang die Inhalte des DBs an den entsprechenden CAN-Master weiter.

## IM 208CAN - CANopen-Master - Firmwareupdate

### Übersicht

Ab der CPU-Firmware-Version 3.4.8 haben Sie die Möglichkeit mittels einer MMC über die CPU ein Firmwareupdate unter anderem auch für den CAN-Master durchzuführen. Die 2 aktuellsten Firmwarestände finden Sie auf [www.vipa.de](http://www.vipa.de) im Service-Bereich und auf dem VIPA-ftp-Server unter <ftp.vipa.de>. Näheres hierzu finden Sie im Handbuch HB97-CPU.

Hierbei gibt es zur Kennzeichnung einer CAN-Master Firmware folgende Namenskonventionen:

`canxx.bin` mit xx geben Sie die Nummer des CAN-Master Steckplatzes an (01 ... 32)



### Achtung!

Beim Aufspielen einer neuen Firmware ist äußerste Vorsicht geboten. Unter Umständen kann Ihr CAN-Master unbrauchbar werden, wenn beispielsweise während der Übertragung die Spannungsversorgung unterbrochen wird oder die Firmware-Datei fehlerhaft ist.

Setzen Sie sich in diesem Fall mit der VIPA-Hotline in Verbindung!

### Firmware-Version ermitteln

Den ausgelieferten Firmwarestand können Sie einem Aufkleber entnehmen, der sich auf der Rückseite des CAN-Master-Moduls befindet.

### Firmware laden und als `canxx.bin` auf MMC übertragen

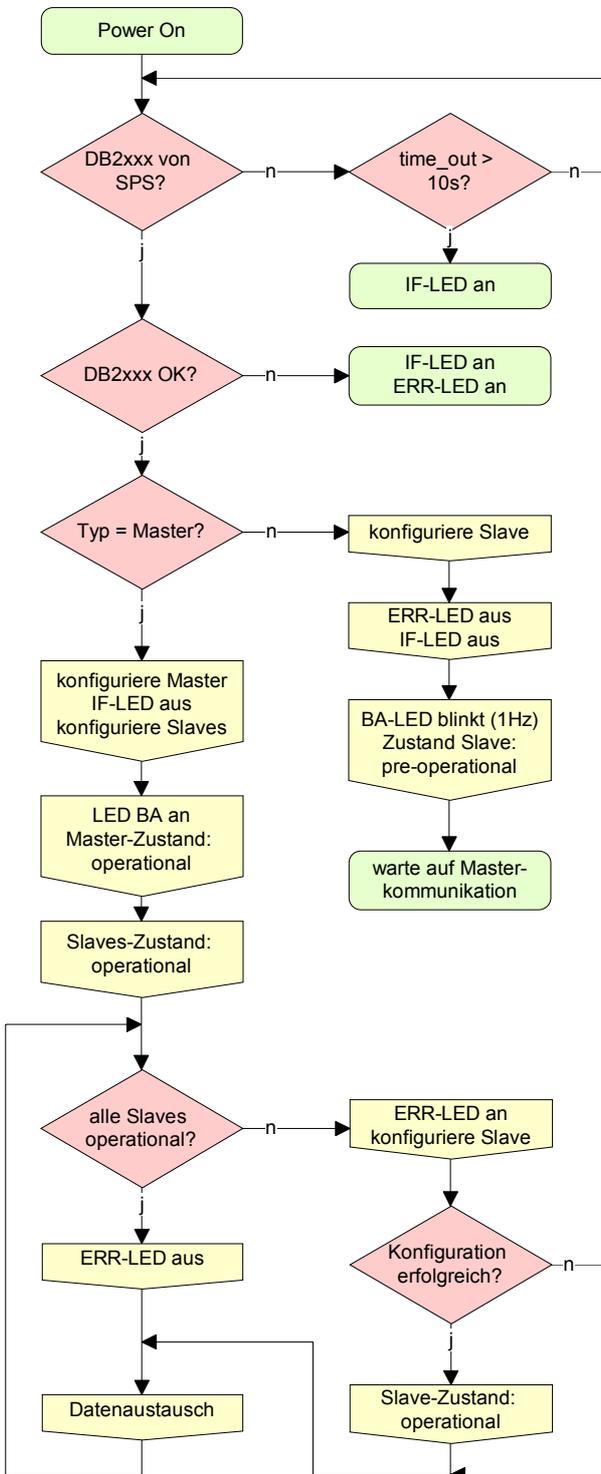
- Gehen Sie auf [www.vipa.de](http://www.vipa.de).
- Klicken Sie auf Service > Download > Firmware Updates.
- Klicken Sie auf "Firmware für CAN Master System 200V"
- Wählen Sie die entsprechende IM 208 Bestell-Nr. aus und laden Sie die Firmware auf Ihren PC.
- Benennen Sie die Datei um in "`canxx.bin`" (xx entspricht dem CAN-Master-Steckplatz beginnend mit 01) und übertragen Sie diese Datei auf eine MMC.



### Hinweis!

Auf dem Server sind immer die 2 aktuellsten Firmware-Versionen abgelegt.

# IM 208CAN - CANopen-Master - Betriebsarten



## STOP → RUN (automatisch)

Nach NETZ EIN und bei gültigen Projektierdaten in der CPU geht der Master automatisch in RUN über. Auf einen Betriebsarten-Schalter für den Master wurde verzichtet.

Nach einem NETZ EIN werden automatisch die Projektierdaten von der CPU an den CAN-Master geschickt. Dieser baut eine Kommunikation zu den CAN-Slaves auf.

Bei erfolgter Kommunikation und gültigen Bus-Parametern, geht der CAN-Master in den Zustand "operational" über. Die LEDs RUN und BA leuchten.

Bei fehlerhaften Parametern bleibt der CAN-Master in STOP und zeigt über die IF-LED einen Parametrierfehler an.

## RUN

Im RUN leuchten die RUN- und BA-LEDs. Jetzt können Daten ausgetauscht werden.

Im Fehlerfall wie z.B. Slave-Ausfall, wird dies am CAN-Master über die ERR-LED angezeigt und ein Alarm an die CPU abgesetzt.

## IM 208CAN - CANopen-Master - Prozessabbild

Das Prozessabbild setzt sich aus folgenden Teilen zusammen:

- Prozessabbild für Eingangs-Daten (PE) für RPDOs
- Prozessabbild für Ausgangsdaten (PA) für TPDOs

Hiervon besteht jeder Teil aus einem 64Byte großen "Digital-Data"- und 320Byte großen "Network Variables"-Bereich.

### Eingabe-Daten

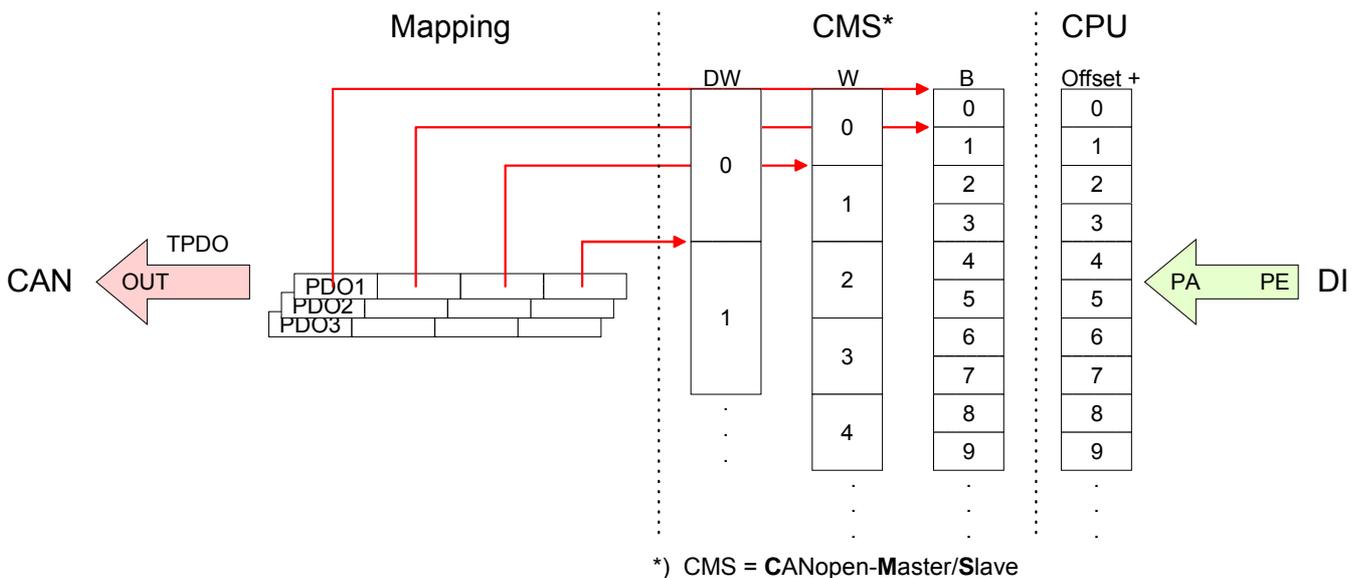
Für Eingabe-Daten gibt es folgende Objekte:

- 8 Bit Digitale Eingabe (Objekt 0x6000)
- 16 Bit Digitale Eingabe (Objekt 0x6100)
- 32 Bit Digitale Eingabe (Objekt 0x6120)
- 8 Bit Eingangs-Netzwerk-Variablen (Objekt 0xA040)
- 16 Bit Eingangs-Netzwerk-Variablen (Objekt 0xA100)
- 32 Bit Eingangs-Netzwerk-Variablen (Objekt 0xA200)
- 64 Bit Eingangs-Netzwerk-Variablen (Objekt 0xA440)

Wie in der nachfolgenden Abbildung zu erkennen ist, wird für die Objekte der digitalen Eingangsdaten der gleiche Speicherbereich in der CPU verwendet.

Beispielsweise würde ein Zugriff auf Index 0x6000 mit Subindex 2 einem Zugriff auf Index 0x6100 mit Subindex 1 entsprechen. Beide Objekte belegen die gleiche Speicherzelle in der CPU.

Bitte beachten Sie, dass auch die Eingangs-Netzwerk-Variablen den gleichen Speicherbereich benutzen.



**Ausgabe-Daten**

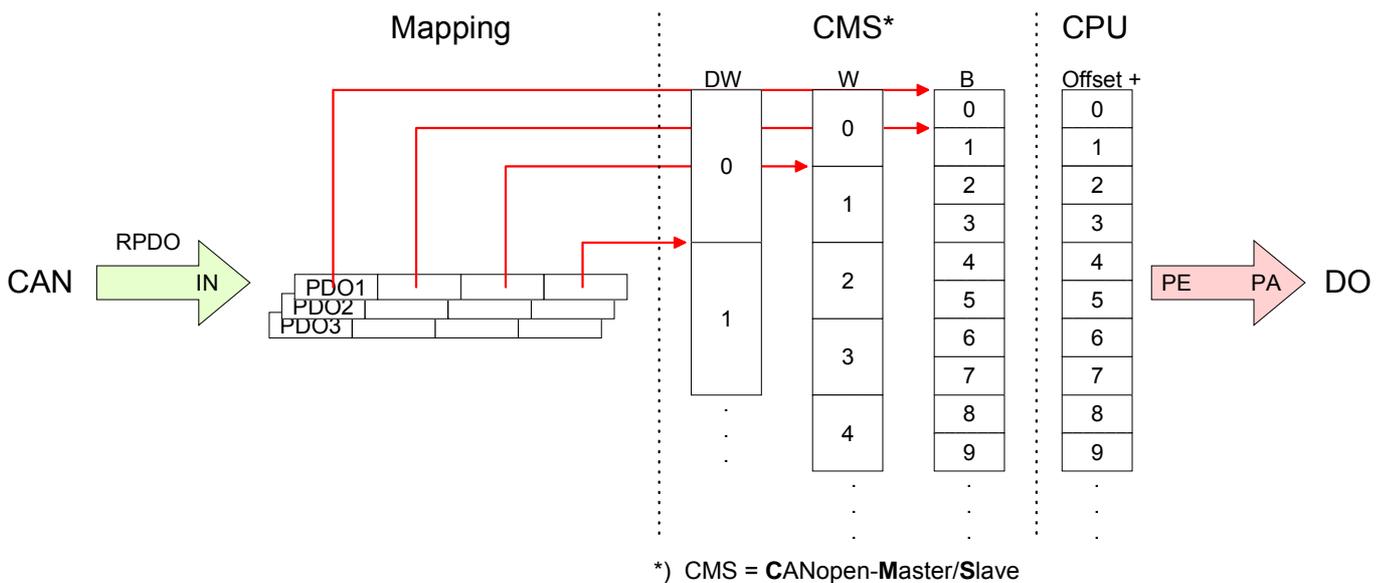
Für die digitalen Ausgabe-Daten wird die Zuordnung ähnlich durchgeführt. Für Ausgabe-Daten gibt es folgende Objekte:

- 8 Bit Digitale Ausgabe (Objekt 0x6200)
- 16 Bit Digitale Ausgabe(Objekt 0x6300)
- 32 Bit Digitale Ausgabe(Objekt 0x6320)
- 8 Bit Ausgangs-Netzwerk-Variablen (Objekt 0xA400)
- 16 Bit Ausgangs-Netzwerk-Variablen (Objekt 0xA580)
- 32 Bit Ausgangs-Netzwerk-Variablen (Objekt 0xA680)
- 64 Bit Ausgangs-Netzwerk-Variablen (Objekt 0xA8C0)

Wie in der nachfolgenden Abbildung zu erkennen ist, wird für die Objekte der digitalen Ausgangsdaten der gleiche Speicherbereich in der CPU verwendet.

Beispielsweise würde ein Zugriff auf Index 0x6200 mit Subindex 2 einem Zugriff auf Index 0x6300 mit Subindex 1 entsprechen. Beide Objekte belegen die gleiche Speicherzelle in der CPU.

Bitte beachten Sie, dass auch die Ausgangs-Netzwerk-Variablen den gleichen Speicherbereich benutzen.



## IM 208CAN - CANopen-Master - 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-Master unterstützt folgende Objekte:

- 40 Transmit PDOs (PDO Linking, PDO Mapping)
- 40 Receive PDOs (PDO Linking, PDO Mapping)
- 2 Standard SDOs (1 Server, 127 Clients)
- 1 Emergency Objekt
- 1 Netzwerkmanagement Objekt NMT
- Node Guarding
- Heartbeat

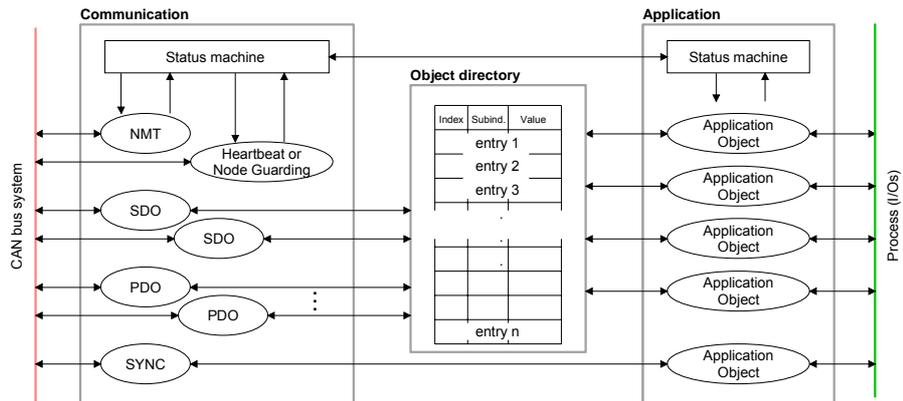


### Hinweis!

Der genaue Aufbau und Dateninhalt aller Objekte ist in den CiA-Profilen DS-301, DS-302, DS-401 und DS-405 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**

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 beim CAN-Master insgesamt 80 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 40 Tx Transmit-PDOs und für Ausgangsdaten 40 Rx Receive-PDOs zur Verfügung. Die PDOs werden aus Sicht des CAN-Masters bezeichnet:

Receive-PDOs (RxDPOs) werden vom CAN-Master empfangen und enthalten Eingangsdaten.

Transmit-PDOs (TxPDOs) werden vom CAN-Master gesendet und enthalten Ausgangsdaten.

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

**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). Ein SDO wird bestätigt übertragen, d.h. jeder Empfang einer Nachricht wird quittiert.



**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.

**SFC 219 CAN\_TLGR SDO-Anforderung an CAN-Master**

Jede CPU hat den SFC 219 integriert. Hiermit können Sie von Ihrem SPS-Programm auf Ihrem CAN-Master einen SDO- Lese- oder Schreibzugriff auslösen. Hierbei adressieren Sie den Master über die Steckplatz-Nr. und den Ziel-Slave über seine CAN-Adresse. Die Prozessdaten bestimmen Sie durch Angabe von Index und Subindex. Über SDO kann pro Zugriff maximal ein Datenwort Prozessdaten übertragen werden. Der SFC 219 beinhaltet folgende Parameter:

Parameter	Deklaration	Typ	Beschreibung
REQUEST	IN	BOOL	
SLOT_MASTER	IN	BYTE	
NODEID	IN	BYTE	
TRANSFERTYP	IN	BYTE	
INDEX	IN	DWORD	
SUBINDEX	IN	DWORD	
CANOPENERROR	OUT	DWORD	
RETVL	OUT	WORD	
BUSY	OUT	BOOL	
DATABUFFER	IN_OUT	ANY	

REQUEST Steuerparameter: 1: Anstoß des Auftrags

SLOT\_MASTER 0: VIPA 21x-2CM01  
1...32: VIPA 208-1CA00, abhängig von der Steckplatznummer

NODELD Adresse des CANopen Knotens (1...127)

TRANSFERTYPE 40h, 60h: Lesen SDO      61h: Schreiben SDO (undefinierte Länge)  
23h: Schreiben SDO (1 DWORD)  
2Bh: Schreiben SDO (1 WORD)  
2Fh: Schreiben SDO ( 1 BYTE)

INDEX CANopen Index

SUBINDEX CANopen Subindex

CANOPENERROR Liegt kein Fehler vor, so liefert *CANOPENERROR* eine 0 zurück.  
Im Fehlerfall beinhaltet *CANOPENERROR* eine der nachfolgend aufgeführten Fehlermeldungen, die vom CAN-Master generiert wird:

Code	Bedeutung
0x05030000	Toggle-Bit nicht geändert
0x05040000	SDO Protokoll Time-out
0x05040001	Client/server Befehlsspezifizierung nicht gültig oder unbekannt
0x05040002	Ungültige Blockgröße (nur Block-Modus)
0x05040003	Ungültige Sequenznummer (nur Block-Modus)
0x05040004	CRC Fehler (nur Block-Modus)
0x05040005	Unzureichender Speicher
0x06010000	Nicht unterstützter Zugriff auf ein Objekt
0x06010001	Lesezugriff auf ein Nur-Schreiben-Objekt
0x06010002	Schreibzugriff auf ein Nur-Lesen-Objekt
0x06020000	Objekt nicht im Objektverzeichnis vorhanden
0x06040041	Objekt kann nicht ins PDO gemappt werden
0x06040042	Anzahl und Länge der zu mappenden Objekte überschreitet PDO-Länge
0x06040043	Generelle Parameterinkompatibilität
0x06040047	Generelle interne Inkompatibilität im Gerät
0x06060000	Zugriffsfehler wegen Hardwareausfall
0x06070010	Datentyp nicht korrekt, Länge der Serviceparameter nicht korrekt
0x06070012	Datentyp nicht korrekt, Serviceparameter zu lang
0x06070013	Datentyp nicht korrekt, Serviceparameter zu kurz
0x06090011	Subindex existiert nicht
0x06090030	Wertebereich der Parameter überschritten (nur für Schreibzugriff)
0x06090031	Zu schreibender Parameterwert ist zu hoch
0x06090032	Zu schreibender Parameterwert ist zu niedrig
0x06090036	Maximumwert ist kleiner als Minimumwert
0x08000000	Genereller Fehler
0x08000020	Die Daten können entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0x08000021	Die Daten können wegen lokaler Kontrollen entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0x08000022	Die Daten können wegen aktuellem Modulstatus entweder nicht transferiert oder nicht in der SPS gespeichert werden.
0x08000023	Dynamische Objektverzeichnisgenerierung fehlgeschlagen oder kein Objektverzeichnis gefunden (z.B. Objektverzeichnis wird aus Datei generiert und ein Dateifehler ist aufgetreten).

**RETVAL** Wird die Funktion fehlerfrei ausgeführt, enthält der Rückgabewert die gültige Länge der Antwortdaten: 1: Byte, 2: Wort, 4: Doppelwort  
Tritt während der Bearbeitung der Funktion ein Fehler auf, enthält der Rückgabewert einen der nachfolgend aufgeführten Fehlercodes.

Code	Bedeutung
0xF021	Ungültige Slave-Adresse (Aufrufparameter gleich 0 oder größer 127)
0xF022	Ungültiger Transfertyp (Wert ungleich 60h, 61h)
0xF023	Ungültige Datenlänge (der Datenpuffer ist zu klein, beim SDO-Lesezugriff sollte dieser mindestens 4 Byte groß sein, beim SDO-Schreibzugriff sollte dieser 1Byte, 2Byte oder 4 Byte groß sein)
0xF024	Der SFC wird nicht unterstützt
0xF025	Schreibpuffer im CANopen-Master ist voll, Service kann zur Zeit nicht bearbeitet werden.
0xF026	Lesebuffer im CANopen-Master ist voll, Service kann zur Zeit nicht bearbeitet werden.
0xF027	Der SDO-Lese- oder Schreibzugriff wurde fehlerhaft beantwortet, siehe CANopen Error Codes.
0xF028	SDO-Timeout (es wurde kein CANopen-Teilnehmer mit der Node-Id gefunden)

**BUSY** Solange Busy = 1 ist der aktuelle Auftrag ist noch nicht beendet.

**DATABUFFER** Datenbereich, über den der SFC kommuniziert. Geben Sie hier einen ANY-Pointer vom Typ Byte an.  
SDO-Lesezugriff: Zielbereich für die gelesenen Nutzdaten.  
SDO-Schreibzugriff: Quellbereich für die zu schreibenden Nutzdaten.



### Hinweis!

Sofern eine SDO-Anforderung fehlerfrei abgearbeitet wurde, enthält *RETVAL* die Länge der gültigen Antwortdaten in (1, 2 oder 4 Byte) und *CANOPENERROR* den Wert 0.

## IM 208CAN - CANopen-Master - Objekt-Verzeichnis

### Struktur

Im CANopen-Objektverzeichnis werden alle für das Gerät 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 – 0x1018 allgemeine kommunikationsspezifische Parameter (z.B. der Gerätename)

0x1400 – 0x1427 Kommunikationsparameter (z.B. Identifier) der Receive-PDOs

0x1600 – 0x1627 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 – 0x1827 Kommunikations- und Mappingparameter der Transmit-PDOs  
0x1A00 – 0x1A27

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

Hier finden Sie die herstellerspezifischen Einträge. Der CAN-Master von VIPA besitzt keine herstellerspezifischen Einträge.

### 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
1000h	Device type
1001h	Error register
1005h	COB-ID SYNC
1006h	Communication Cycle Period
1007h	Synchronous Window Length
1008h	Manufacturer Hardware Version
1009h	Hardware Version
100Ah	Software Version
100Ch	Guard Time
100Dh	Life Time Factor
1016h	Consumer Heartbeat Time
1017h	Producer Heartbeat Time
1018h	Identity Object
1400h bis 1427h	Receive PDO Communication Parameter
1600h bis 1627h	Receive PDO Mapping Parameter
1800h bis 1827h	Transmit PDO Communication Parameter
1A00h bis 1A27h	Transmit PDO Mapping Parameter
1F22h	Concise DCF
1F25h	Post Configuration
1F80h	NMT StartUp
1F81h	Slave Assignment
1F82h	Request NMT
1F83h	Request Guarding
6000h	Digital-Input-8-Bit Array (see DS 401)
6100h	Digital-Input-16-Bit Array (see DS 401)
6120h	Digital-Input-32Bit Array (see DS 401)
6200h	Digital-Output-8-Bit Array (see DS 401)
6300h	Digital-Output-16-Bit Array (see DS 401)
6320h	Digital-Output-32-Bit Array (see DS 401)
A040h	Dynamic Unsigned8 Input
A100h	Dynamic Unsigned16 Input
A200h	Dynamic Unsigned32 Input
A4400h	Dynamic Unsigned64 Input
A4C0h	Dynamic Unsigned8 Output
A580h	Dynamic Unsigned16 Output
A680h	Dynamic Unsigned32 Output
A8C0h	Dynamic Unsigned64 Output

**Device Type**

Index	Sub-index	Name	Type	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)	405dec=0x0195

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

z=1 → digitale Eingänge

y=1 → digitale Ausgänge

x=1 → analoge Eingänge

w=1 → analoge Ausgänge

**Error register**

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

Bit 7							Bit 0
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)

**SYNC identifier**

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

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

**Device name**

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

VIPA Master / Slave 208-1CA00

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

**Hardware version**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1009	0	Manufacturer Hardware version	Visible string	ro	N	1.00	Hardware version number of bus coupler

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

**Software version**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100A	0	Manufacturer Software version	Visible string	ro	N	1.xx	Software version number CANopen software

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

**Guard time**

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

**Life time factor**

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

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).

**Consumer Heartbeat Time**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1016	0	Consumer heartbeat time	Unsigned8	ro	N	0x05	Number of entries
	1...127		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	Type	Attr.	Map.	Default value	Meaning
0x1017	0	Producer heartbeat time	Unsigned16	rw	N	0x0000	Defines the cycle time of heartbeat in ms

### Identity Object

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1018	0	Identity Object	Unsigned8	ro	N	0x04	Contains general Informations about the device (number of entries)
	1	Vendor ID	Unsigned32	ro	N	0xAFFFEAFFE	Vendor ID
	2	Product Code	Unsigned32	ro	N	0x2081CA00	Product Code
	3	Revision Number	Unsigned32	ro	N		Revision Number
	4	Serial Number	Unsigned32	ro	N		Serial Number

### Communication parameter RxPDO

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1400 ... 0x1427	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, Subindex 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 (0) oder nicht (1), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1).

Der Subindex 2 enthält die Übertragungsart.

**Mapping RxPDO**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1600 ...	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the first receive PDO; subindex 0: number of mapped objects
0x1627	1	1st mapped object	Unsigned32	rw	N	0x62000108	(2 byte index, 1 byte subindex, 1 byte bit-width)
	2	2nd mapped object	Unsigned32	rw	N	0x62000208	(2 byte index, 1 byte subindex, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8th mapped	Unsigned32	rw	N	0x62000808	(2 byte index, 1 byte subindex, 1 byte bit-width)

Die Empfangs-PDOs erhalten automatisch über den Koppler ein Default-Mapping abhängig von den angeschlossenen Modulen.

**Communication parameter TxPDO1**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1800 ...	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter of the first transmit PDO, subindex 0: number of following parameters
0x1827	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 (0) oder nicht (1), 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.

**Mapping TxPDO1**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1A00 ...	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the first transmit PDO;
0x1A27	1	1st mapped object	Unsigned32	rw	N	0x60000108	subindex 0: number of mapped objects (2 byte index, 1 byte subindex, 1 byte bit-width)
	2	2nd mapped object	Unsigned32	rw	N	0x60000208	(2 byte index, 1 byte subindex, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8th mapped object	Unsigned32	rw	N	0x60000808	(2 byte index, 1 byte subindex, 1 byte bit-width)

Die Sende-PDOs erhalten automatisch über den Koppler ein Default-Mapping, abhängig von den angeschlossenen Modulen.

**Concise DCF**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F22	Array	Concise DCF	Domain	rw	N		

Dieses Objekt ist für den Configuration Manager erforderlich. Das Concise-DCF ist eine Kurzfassung des DCF (**D**evice **C**onfiguration **F**ile).

**Post Configuration**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F25	Array	ConfigureSlave	Unsigned32	rw	N	0x00000000	

Der Configuration Manager kann über diesen Eintrag angewiesen werden, eine gespeicherte Konfiguration in das Netz zu übertragen.

Die Konfiguration kann zu jeder Zeit über Index 0x1F25 für einen bestimmten Knoten ausgelöst werden.

Subindex 0 hat den Wert 128.

Subindex x (mit x = 1..127): Löst Rekonfiguration für Knoten mit der Node ID x aus.

Subindex 128: Rekonfiguration aller Knoten.

Soll z.B. für den Knoten 2 die Konfiguration ausgelöst werden und sind für diesen Knoten Konfigurationsdaten vorhanden, so ist der Wert 0x666E6F63 (ASCII = "conf") auf das Objekt 1F25h Subindex 2 zu schreiben.

**NMT Start-up**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F80	0x00	NMTStartup	Unsigned32	rw	N	0x00000000	

Hier geben Sie an, ob das Gerät der NMT-Master ist.

Bit	Meaning
Bit 0	0 : Device is NOT the NMT Master. All other bits have to be ignored. The objects of the Network List have to be ignored. 1 : Device is the NMT Master.
Bit 1	0 : Start only explicitly assigned slaves. 1 : After boot-up perform the service NMT Start Remote Node All Nodes
Bit 2..31	Reserved by CiA, always 0

**Slave Assignment**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F81	0x00	SlaveAssignment	Unsigned32	rw	N	0x00000000	

Hier erfolgt ein Eintrag der Knoten, die vom Master überwacht, kontrolliert und gesteuert werden sollen. Für jeden zugeordneten Knoten ist hier ein Eintrag vorzunehmen.

Subindex 0 hat den Wert 127. Jeder andere Subindex korrespondiert mit der Node-ID des Knotens.

Byte	Bit	Meaning
Byte 0	Bit 0	0: Node with this ID is not a slave 1: Node with this ID is a slave. After configuration (with Configuration Manager) the Node will be set to state Operational.
	Bit 1	0: On Error Control Event or other detection of a booting slave inform the application. 1: On Error Control Event or other detection of a booting slave inform the application and automatically start Error Control service.
	Bit 2	0: On Error Control Event or other detection of a booting slave do NOT automatically configure and start the slave. 1: On Error Control Event or other detection of a booting slave do start the process Start Boot Slave.
	Bit 3..7	Reserved by CiA, always 0
Byte 1		8 Bit Value for the RetryFactor
Byte 2,3		16 Bit Value for the GuardTime

**Request NMT**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F82	0x00	RequestNMT	Unsigned32	rw	N	0x00000000	

Ist ein vollständig autonomer Start des Stacks nicht gewünscht, so können die Funktionalitäten:

- Statusumschaltung
- Starten des Guardings
- Konfiguration über CMT

auch für jeden Knoten einzeln auf Anfrage durchgeführt werden. Die Anfrage erfolgt immer über Objekte im Objektverzeichnis.

Die Umschaltung des Kommunikationsstatus aller im Netz vorhandenen Knoten (einschließlich des lokalen Slaves) wird dabei über den Eintrag 1F82h im lokalen Objektverzeichnis bewerkstelligt:

Subindex 0 hat den Wert 128.

Subindex x (with x=1..127): Löst NMT-Service für Knoten mit der Node ID x aus.

Subindex 128: Löst NMT-Service für alle Knoten aus.

Beim Schreiben wird der gewünschte Status als Wert angegeben

State	Value
Prepared	4
Operational	5
ResetNode	6
ResetCommunication	7
PreOperational	127

**Request Guarding**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1F83	0x00	RequestGuarding	Unsigned32	rw	N	0x00000000	

Subindex 0 hat den Wert 128.

Subindex x (with x=1..127): Löst Guarding für den Slave mit Node ID x aus.

Value	Write Access	Read Access
1	Start Guarding	Slave actually is guarded
0	Stop Guarding	Slave actually is not guarded

Subindex 128: Request Start/Stop Guarding für alle Knoten.

**8bit Digital inputs**

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

**16bit Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6100	0x00	16bit digital input block	Unsigned8	ro	N	depending on the fitted components	Number of available digital 16bit input blocks
	0x01	1st input block	Unsigned16	ro	N		1st digital input block
	...	...	...	...	...	...	...
	0x20	32nd input block	Unsigned16	ro	N		32nd digital input block

**32bit Digital inputs**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x6120	0x00	32bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 32bit input blocks
	0x01	1st input block	Unsigned32	ro	N		1st digital input block
	...	...	...	...	...	...	...
	0x10	16th input block	Unsigned32	ro	N		16th digital input block

**8bit Digital outputs**

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

**16bit Digital outputs**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x6300	0x00	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16bit output blocks
	0x01	1st output block	Unsigned16	rw	N		1st digital output block
	...	...	...	...	...	...	...
	0x20	32nd output block	Unsigned16	rw	N		32nd digital output block

**32bit Digital outputs**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x6320	0x00	32bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 32-bit output blocks
	0x01	1st output block	Unsigned32	rw	N		1st digital output block
	...	...	...	...	...	...	...
	0x10	16th output block	Unsigned32	rw	N		16th digital output block

**8bit Network input variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA040	0x00	8bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1st input block	Unsigned8	ro	Y		1st digital input block
	...	...	...	...	...	...	...
	0x140	320th input block	Unsigned8	ro	Y		320th digital input block

**16bit Network input variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA100	0x00	16bit digital input block	Unsigned8	ro	N	depending on the fitted components	Number of available digital 16bit input blocks
	0x01	1st input block	Unsigned16	ro	N		1st digital input block
	...	...	...	...	...	...	...
	0xA0	160th input block	Unsigned16	ro	N		160th digital input block

**32bit Network input variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA200	0x00	32bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 32bit input blocks
	0x01	1st input block	Unsigned32	ro	N		1st digital input block
	...	...	...	...	...	...	...
	0x50	80th input block	Unsigned32	ro	N		80th digital input block

**64bit Network input variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA440	0x00	64bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 64bit input blocks
	0x01	1st input block	Unsigned32	ro	N		1st digital input block
	...	...	...	...	...	...	...
	0x28	40th input block	Unsigned32	ro	N		40th digital input block

**8bit Network output variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA400	0x00	8bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8bit output blocks
	0x01	1st output block	Unsigned8	rw	Y		1st digital output block
	...	...	...	...	...	...	...
	0x140	320th output block	Unsigned8	rw	Y		320th digital output block

**16bit Network output variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA580	0x00	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16bit output blocks
	0x01	1st output block	Unsigned16	rw	N		1st digital output block
	...	...	...	...	...	...	...
	0xA0	160th output block	Unsigned16	rw	N		160th digital output block

**32bit Network output variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA680	0x00	32bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 32-bit output blocks
	0x01	1st output block	Unsigned32	rw	N		1st digital output block
	...	...	...	...	...	...	...
	0x50	80th output block	Unsigned32	rw	N		80th digital output block

**64bit Network output variables**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0xA8C0	0x00	64bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 64bit output blocks
	0x01	1st output block	Unsigned32	rw	N		1st digital output block
	...	...	...	...	...	...	...
	0x50	40th output block	Unsigned32	rw	N		40th digital output block

## IM 253CAN - CANopen-Slave - Aufbau

### Eigenschaften

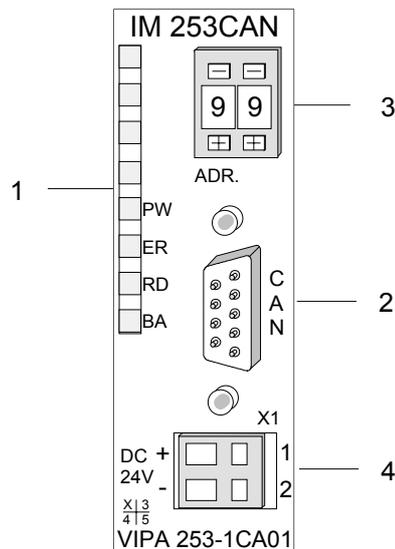
- 10 Rx und 10 Tx PDO
- 2 SDOs
- Unterstützung aller Übertragungsraten
- PDO-Linking
- PDO-Mapping

### Einschränkungen 253-1CA30 - ECO

Der IM 253-1CA30 - ECO ist funktional identisch mit dem IM 253-1CA01 und hat folgende Einschränkungen:

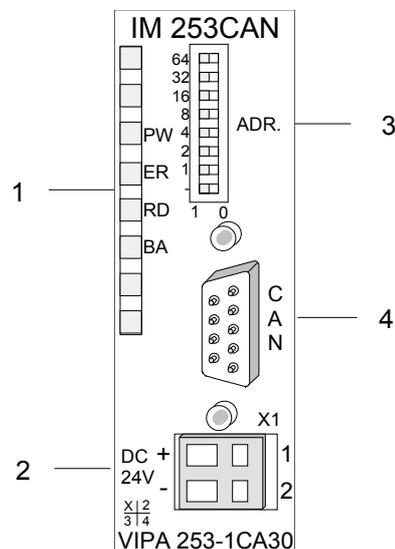
- CANopen-Slave für max. 8 Peripherie-Module
- Integriertes DC 24V-Netzteil zur Versorgung der Peripherie-Module mit max. 0,8A.
- Vorgabe der CAN-Bus-Adresse über DIP-Schalter

### Frontansicht 253-1CA01



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

### Frontansicht 253-1CA30 - ECO



- [1] LED Statusanzeigen
- [2] Anschluss für DC 24V Spannungsversorgung
- [3] Adress- bzw. Baudraten-einsteller (DIP-Schalter)
- [4] CAN-Bus-Stecker

## 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	Grün	Signalisiert eine anliegende Betriebsspannung.
ER	Rot	Blinkt wenn in der CAN-Kommunikation die Fehlerzähler übergelaufen sind. (z.B. kein weiterer CAN-Teilnehmer am Bus oder falsche CAN-Transferrate)
RD	Grün	Leuchtet bei Fehler in der Rückwandbusübertragung. Blinkt mit 1Hz bei positivem Selbsttest und erfolgreicher Initialisierung
BA	Gelb	Leuchtet bei Datenübertragung über den VBUS. 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

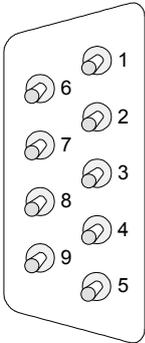
 PW ein      Fehler in der CAN-Baudrateneinstellung  
 ER blinkt 10Hz  
 RD blinkt 10Hz  
 BA blinkt 10Hz

 PW ein      Modul-ID-Einstellung aktiviert  
 ER aus  
 RD blinkt 1Hz  
 BA aus

**9poliger SubD-Stecker**

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

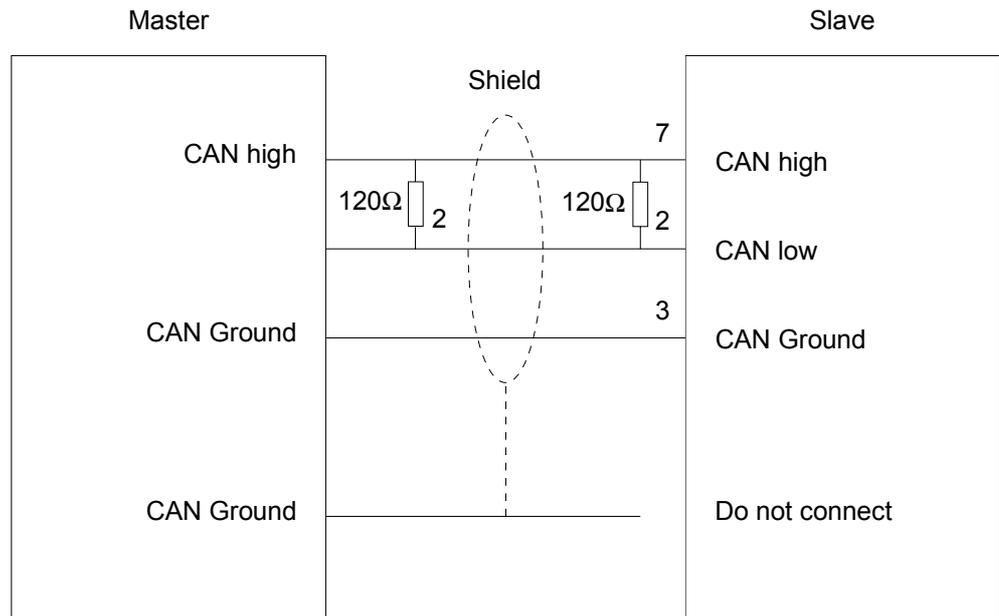
Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:



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

**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.



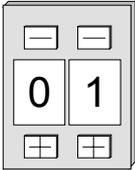
**Hinweis!**

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

### Adresseinsteller für Baudrate und Modul-ID

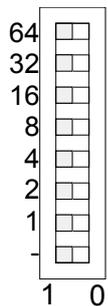
Über diesen Adresseinsteller stellen Sie die CAN-Baudrate sowie die Modul-ID ein. Jede Modul-ID darf nur einmal am Bus vergeben sein.

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



### Adress-Schalter IM 253-1CA31 - ECO

Im Gegensatz zu dem oben beschriebenen Codiertaster besitzt der IM 253-1CA30 - ECO zur Adresseinstellung einen DIL-Schalter.



### Spannungs- versorgung

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. Beim IM 253-1CA30 - ECO ist der Strom auf max. 0,8A begrenzt.

Das Netzteil ist gegen Verpolung und Überstrom geschützt.

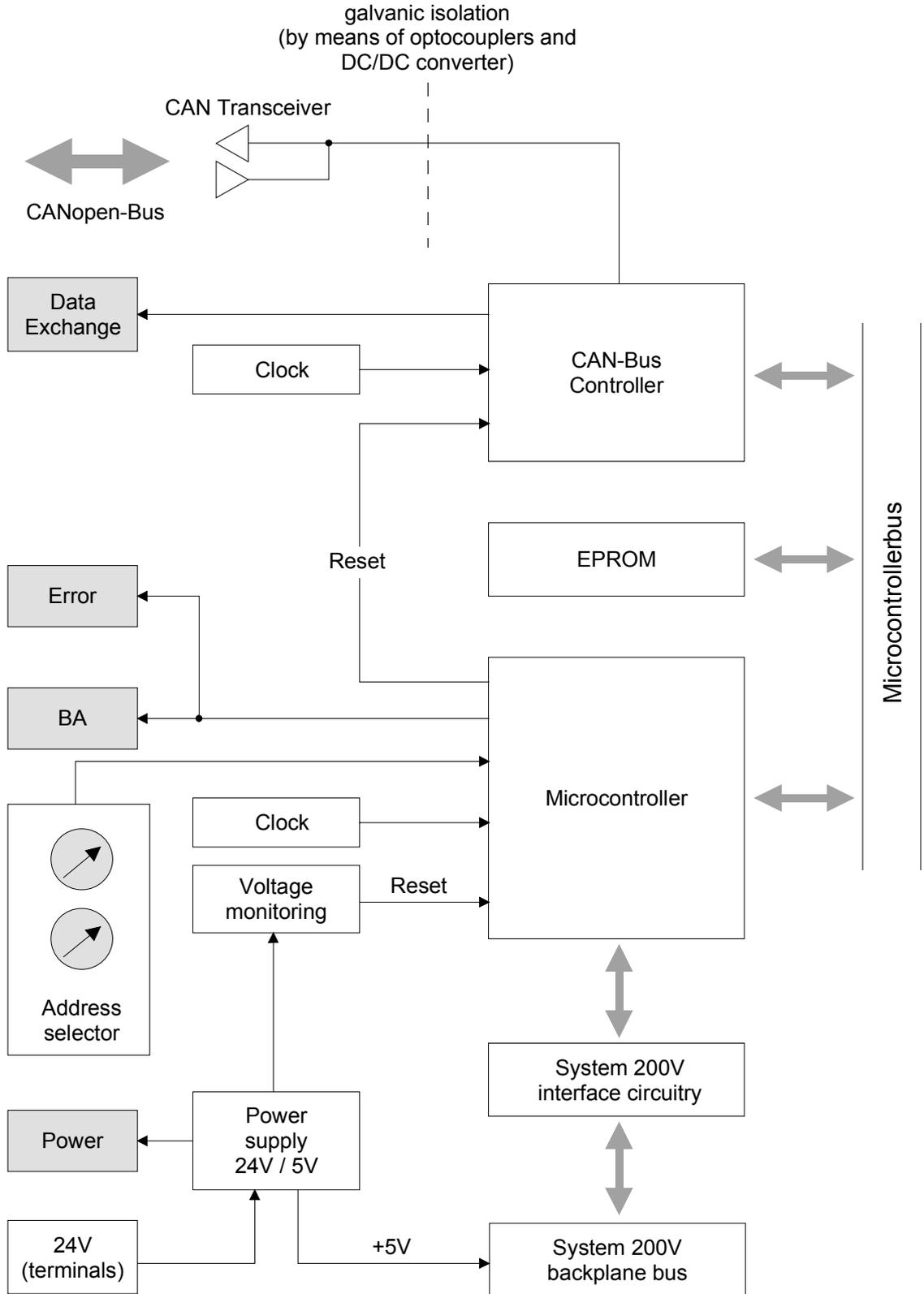
CAN-Bus und Rückwandbus sind galvanisch voneinander getrennt.



### Achtung!

Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung!

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

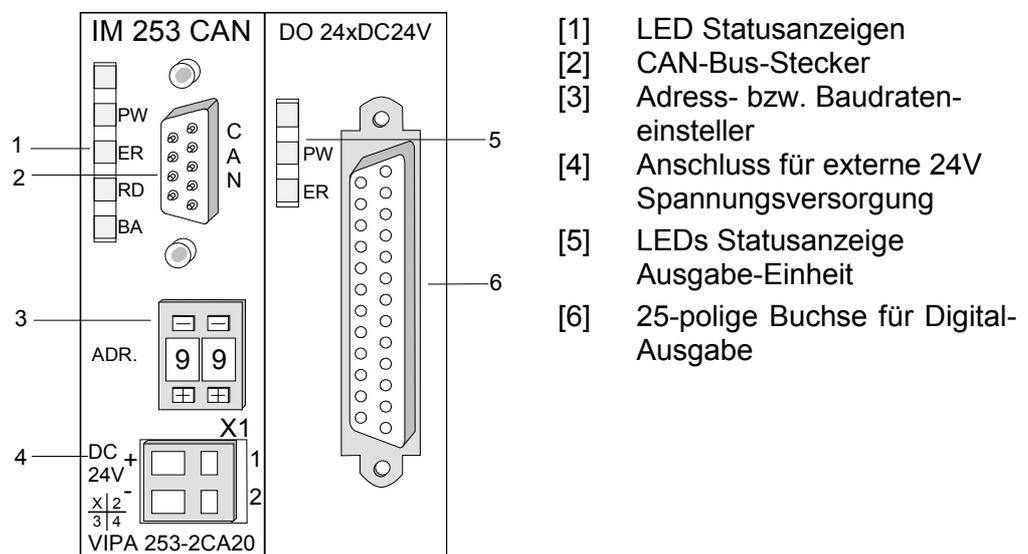


## IM 253CAN - CANopen-Slave DO 24xDC 24V - Aufbau

### 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	Grün	Signalisiert eine anliegende Betriebsspannung.
ER	Rot	Blinkt wenn in der CAN-Kommunikation die Fehlerzähler übergelaufen sind. (z.B. kein weiterer CAN-Teilnehmer am Bus oder falsche CAN-Transferrate).
RD	Grün	Leuchtet bei Fehler in der Rückwandbusübertragung. Blinkt mit 1Hz bei positivem Selbsttest und erfolgreicher Initialisierung
BA	Gelb	Leuchtet bei Datenübertragung über den VBUS. 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**

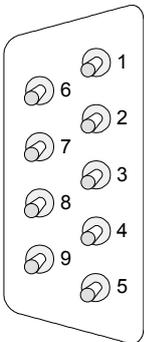
Auf dem digitalen Ausgabe-Teil befinden sich 2 LEDs, die folgende Funktion haben:

Bezeichnung	Farbe	Bedeutung
PW	Grün	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.

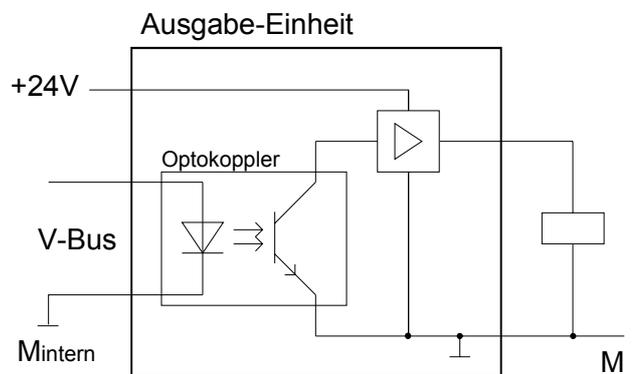
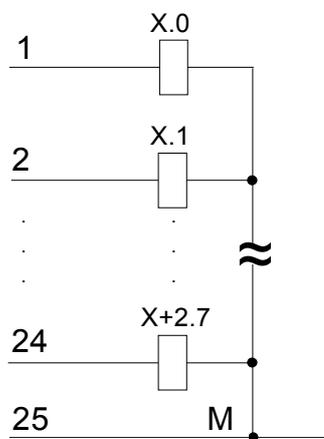
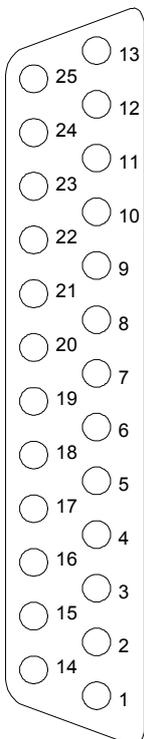
Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:



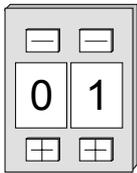
Pin	Belegung
1	nicht belegt
2	CAN low
3	CAN Ground
4	nicht belegt
5	nicht belegt
6	nicht belegt
7	CAN high
8	nicht belegt
9	nicht belegt

**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**

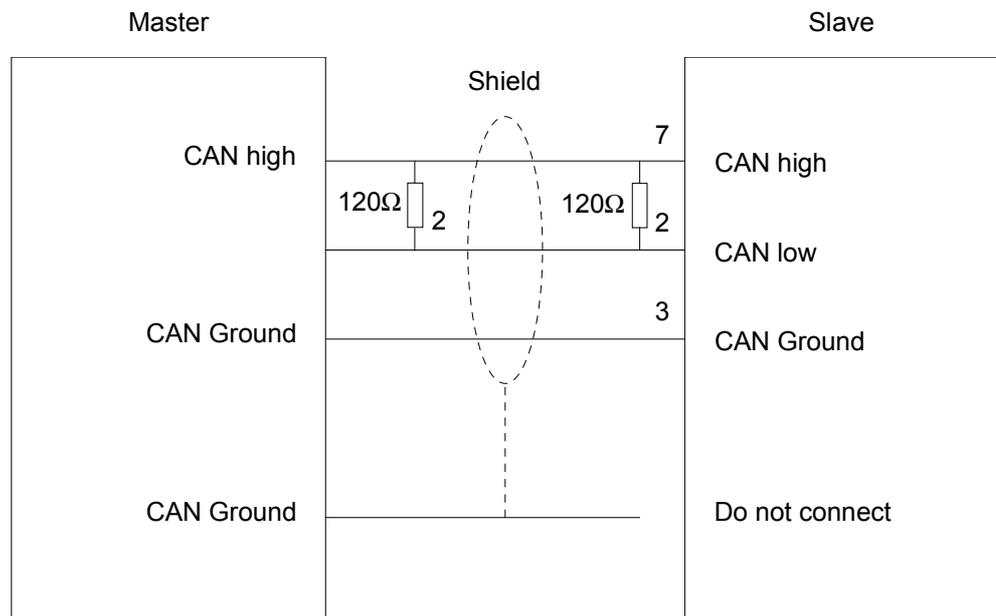
Der CAN-Bus-Koppler besitzt ein eingebautes Netzteil. Das Netzteil ist mit 24V Gleichspannung zu versorgen. Über die Versorgungsspannung werden neben der Buskopplerelektronik auch die angeschlossenen Module über den Rückwandbus versorgt. Bitte beachten Sie, dass das integrierte Netzteil den Rückwandbus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

## IM 253CAN - CANopen-Slave - 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 2</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 befindet sich nach dem Boot-Up im Objektverzeichnis in der Variablen 0x1014 der 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	Info4
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.

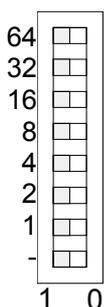
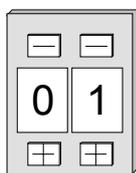
## IM 253CAN - CANopen-Slave - Baudrate und Modul-ID

### 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.

## IM 253CAN - CANopen-Slave - 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

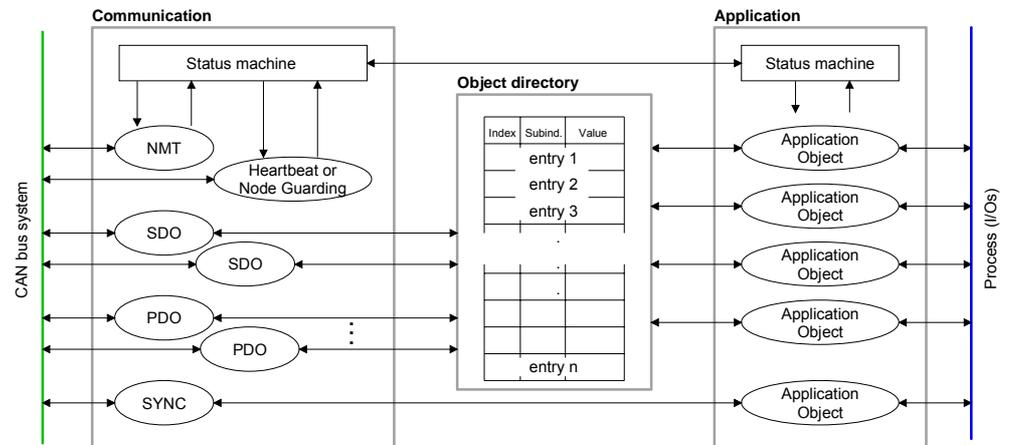


**Hinweis!**

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.

## IM 253CAN - CANopen-Slave - PDO

### 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...240). 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 niederprioreres 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.

## IM 253CAN - CANopen-Slave - SDO

### 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)

## IM 253CAN - CANopen-Slave - 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 0x1A00 – 0x1A0F	Kommunikations- und Mappingparameter der 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	Type	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 → digitale Eingänge

y=1 → digitale Ausgänge

x=1 → analoge Eingänge

w=1 → analoge Ausgänge

## Error register

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

Bit7							Bit0
ManSpec	reserved	reserved	Comm.	reserved	reserved	reserved	Generic

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

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

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

### Device name

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

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

VIPA IM 253 1CA30 = VIPA CANopen-Slave IM 253-1CA30 - ECO

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

**Hardware version**

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

VIPA IM 253 1CA01 = 1.00

VIPA IM 253 1CA30 = 1.00

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

**Software version**

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

VIPA IM 253 1CA01 = 3.xx

VIPA IM 253 1CA30 = 3.xx

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

**Node number**

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

### Life time factor

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

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

### Save parameters

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

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

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

### Consumer Heartbeat Time

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1016	0	Consumer heartbeat time	Unsigned8	ro	N	0x05	Number of entries
	1		Unsigned32	rw	N	0x00000000	Consumer 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	Type	Attr.	Map.	Default value	Meaning
0x1017	0	Producer heartbeat time	Unsigned16	rw	N	0x0000	Defines the cycle time of heartbeat in ms

### Identity Object

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

\*) Default value Product Code: bei 253-1CA01: 0x2531CA01  
bei 253-1CA30: 0x2531CA30

### Modular Devices

Index	Sub-index	Name	Type	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**

Modultyp	Typkennung	Eingabe-Byte	Ausgabe-Byte
DI 8	9FC1h	1	-
DI 8 - Alarm	1FC1h	1	-
DI 16	9FC2h	2	-
DI 16 / 1C	08C0h	6	6
DI 32	9FC3h	4	-
DO 8	AFC8h	-	1
DO 16	AFD0h	-	2
DO 32	AFD8h	-	4
DIO 8	BFC9h	1	1
DIO 16	BFD2h	2	2
AI2	15C3h	4	-
AI4	15C4h	8	-
AI4 - fast	11C4h	8	-
AI8	15C5h	16	-
AO2	25D8h	-	4
AO4	25E0h	-	8
AO8	25E8h	-	16
AI2 / AO2	45DBh	4	4
AI4 / AO2	45DCh	8	4
SM 238	45DCh	8	4
	38C4h	16	16
CP 240	1CC1h	16	16
FM 250	B5F4h	10	10
FM 250-SSI	B5DBh	4	4
FM 253, FM 254	18CBh	16	16

**Error Behavior**

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

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	Type	Attr.	Map.	Default value	Meaning
0x1400	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000200 + NODE_ID	COB-ID RxPDO1
	2	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 (0) oder nicht (1), 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	Type	Attr.	Map.	Default value	Meaning
0x1401	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC0000300 + NODE_ID	COB-ID RxPDO2
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

---

**Communication  
parameter RxPDO3**

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

---

**Communication  
parameter RxPDO4**

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

---

**Communication  
parameter RxPDO5**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1404	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	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	Type	Attr.	Map.	Default value	Meaning
0x1405	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	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	Type	Attr.	Map.	Default value	Meaning
0x1406	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	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	Type	Attr.	Map.	Default value	Meaning
0x1407	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	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	Type	Attr.	Map.	Default value	Meaning
0x1408	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	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	Type	Attr.	Map.	Default value	Meaning
0x1409	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC00007C0 + NODE_ID	COB-ID RxPD10
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Mapping RxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1600	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the first receive PDO; sub-index 0: number of mapped objects
	1	1st mapped object	Unsigned32	rw	N	0x62000108	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2nd mapped object	Unsigned32	rw	N	0x62000208	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8th 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	Type	Attr.	Map.	Default value	Meaning
0x1601	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the second receive PDO; sub-index 0: number of mapped objects
	1	1st mapped object	Unsigned32	rw	N	0x64110110	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2nd mapped object	Unsigned32	rw	N	0x64110210	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8th mapped	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)

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	Type	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
	1	1st mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2nd mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8th mapped	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)

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

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

### Communication parameter TxPDO3

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1802	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 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	Type	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	Type	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	Type	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	Type	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	Type	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	Type	Attr.	Map.	Default value	Meaning
0x1808	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 9th transmit PDO. COB-ID TxPDO9
	1	COB-ID	Unsigned32	rw	N	0x800004C0 + NODE_ID	
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO10

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1809	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 10th transmit PDO. COB-ID TxPDO10
	1	COB-ID	Unsigned32	rw	N	0x800006C0 + NODE_ID	
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Mapping TxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1A00	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the first transmit PDO; sub-index 0: number of mapped objects
	1	1st mapped object	Unsigned32	rw	N	0x60000108	
	2	2nd mapped object	Unsigned32	rw	N	0x60000208	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8th 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	Type	Attr.	Map.	Default value	Meaning
0x1A01	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the second transmit PDO; sub-index 0: number of mapped objects
	1	1st mapped object	Unsigned32	rw	N	0x64010110	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2nd mapped object	Unsigned32	rw	N	0x64010210	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8th 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	Type	Attr.	Map.	Default value	Meaning
0x1A02 - 0x1A09	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the 3rd to 10th transmit PDO; sub-index 0: number of mapped objects
	1	1st mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2nd mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8th 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	Type	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	Type	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	Type	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	Type	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	Type	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 0 to 3	Unsigned32	rw	N	depending on the components fitted	Parameter bytes 0 to 3
	2	Prm 4 to 7	Unsigned32	rw	N	depending on the components fitted	Parameter bytes 4 to 7
	3	Prm 8 to 11	Unsigned32	rw	N	depending on the components fitted	Parameter bytes 8 to 11
	4	Prm 12 to 15	Unsigned32	rw	N	depending on the components fitted	Parameter bytes 12 to 15

Über die Indizes 0x3001 bis 0x3010 können die Analogmodule, Zähler und Kommunikationsmodule parametrierbar 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
FM 254	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
    
```



## 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	1st mapped object	Unsigned32	rw	N		
	...	...	...	...	...		
	0x40	8th 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	8bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1st input block	Unsigned8	ro	Y		
	...	...	...	...	...		
	0x48	27th input block	Unsigned8	ro	Y		

### 8bit Polarity Digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6002	0x00	8bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1st input block	Unsigned8	rw	N	0x00	1st polarity digital input block
	...	...	...	...	...	...	...
	0x48	72nd input block	Unsigned8	rw	N	0x00	72nd 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	16bit digital input block	Unsigned8	ro	N	depending on the fitted components	Number of available digital 16bit input blocks
	0x01	1st input block	Unsigned16	ro	N		1st digital input block
	...	...	...	...	...	...	...
	0x24	36th input block	Unsigned16	ro	N		36th digital input block

**16bit Polarity  
Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6102	0x00	16bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 16bit input blocks
	0x01	1st input block	Unsigned16	rw	N	0x0000	1st polarity digital input block
	...	...	...	...	...	...	...
	0x24	36th input block	Unsigned16	rw	N	0x0000	36th 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	32bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 32bit input blocks
	0x01	1st input block	Unsigned32	ro	N		1st digital input block
	...	...	...	...	...	...	...
	0x12	18th input block	Unsigned32	ro	N		18th digital input block

### 32bit Polarity Digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6122	0x00	8bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 32bit input blocks
	0x01	1st input block	Unsigned32	rw	N	0x00000000	1st polarity digital input block
	...	...	...	...	...	...	...
	0x12	18th input block	Unsigned32	rw	N	0x00000000	18th 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	8bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8bit output blocks
	0x01	1st output block	Unsigned8	rw	Y		1st digital output block
	...	...	...	...	...	...	...
	0x48	72nd output block	Unsigned8	rw	Y		72nd digital output block

### 8bit Change Polarity Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6202	0x00	8bit digital output block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 8bit output blocks
	0x01	1st output block	Unsigned8	rw	N	0x00	1st polarity digital output block
	...	...	...	...	...	...	...
	0x48	72nd output block	Unsigned8	rw	N	0x00	72nd 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	8bit digital output block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 8bit output blocks
	0x01	1st output block	Unsigned8	rw	N	0xFF	1st error mode digital output block
	...	...	...	...	...	...	...
	0x48	72nd output block	Unsigned8	rw	N	0xFF	72nd 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	8bit digital output block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 8bit output blocks
	0x01	1st output block	Unsigned8	rw	N	0x00	1st error value digital output block
	...	...	...	...	...	...	...
	0x48	72nd output block	Unsigned8	rw	N	0x00	72nd error value digital output block

Vorausgesetzt der Error Mode ist aktiviert, wird im Fehlerfall der hier vorgegebene Wert übernommen.

### 16bit Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6300	0x00	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16bit output blocks
	0x01	1st output block	Unsigned16	rw	N		1st digital output block
	...	...	...	...	...	...	...
	0x24	36th output block	Unsigned16	rw	N		36th digital output block

### 16bit Change Polarity Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6302	0x00	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16bit output blocks
	0x01	1st output block	Unsigned16	rw	N	0x0000	1st polarity digital output block
	...	...	...	...	...	...	...
	0x24	36th output block	Unsigned16	rw	N	0x0000	36th 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	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16-bit output blocks
	0x01	1st output block	Unsigned16	rw	N	0xFFFF	1st error mode digital output block
	...	...	...	...	...	...	...
	0x24	36th output block	Unsigned16	rw	N	0xFFFF	36th error mode digital output block

Dieses Objekt zeigt an, ob ein Ausgang im Falle eines internen Gerätefehlers einen vordefinierten Fehlerwert annimmt (s. Objekt 6307).

1 = Ausgangswert nimmt vordefinierten Wert aus Objekt 6307

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	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 16bit output blocks
	0x01	1st output block	Unsigned16	rw	N	0x0000	1st error value digital output block
	...	...	...	...	...	...	...
	0x24	36th output block	Unsigned16	rw	N	0x0000	36th error value digital output block

Vorausgesetzt der entsprechende ErrorMode ist aktiviert, setzen Gerätefehler den Ausgang auf den Wert, der durch dieses Objekt definiert wird.

### 32bit Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6320	0x00	32bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 32bit output blocks
	0x01	1st output block	Unsigned32	rw	N		1st digital output block
	...	...	...	...	...	...	...
	0x12	18th output block	Unsigned32	rw	N		18th digital output block

### 32bit Change Polarity Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6322	0x00	32bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 32bit output blocks
	0x01	1st output block	Unsigned32	rw	N	0x00000000	1st polarity digital output block
	...	...	...	...	...	...	...
	0x12	18th output block	Unsigned32	rw	N	0x00000000	18th 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	32bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 32bit output blocks
	0x01	1st output block	Unsigned32	rw	N	0xFFFFFFFF	1st error mode digital output block
	...	...	...	...	...	...	...
	0x48	18th output block	Unsigned32	rw	N	0xFFFFFFFF	18th error mode digital output block

Dieses Objekt zeigt an, ob ein Ausgang auf einen vordefinierten Fehlerwert gesetzt wird (s. Objekt 6307), falls ein interner Gerätefehler auftritt.

1 = Ausgangswert übernimmt den in Objekt 6307 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	32bit digital input block	Unsigned8	ro	N	depending on the components fitted	Number of available digital 32bit output blocks
	0x01	1st output block	Unsigned32	rw	N		1st error value digital output block
	...	...	...	...	...		...
	0x12	18th output block	Unsigned32	rw	N		18th error value digital output block

Vorausgesetzt der entsprechende ErrorMode ist aktiviert, setzen Gerätefehler den Ausgang auf den Wert, der durch dieses Objekt definiert wird.

### Analog inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6401	0x00	2byte input block	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	1st input channel	Unsigned16	ro	Y		1st analog input channel
	...	...	...	...	...		...
	0x24	24th input channel	Unsigned16	ro	Y		24th analog input channel

### Analog outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6411	0x00	2byte output block	Unsigned8	ro	N	depending on the components fitted	Number of available analog outputs
	0x01	1st output channel	Unsigned16	ro	Y		1st analog output channel
	...	...	...	...	...		...
	0x24	24th output channel	Unsigned16	ro	Y		24th 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 1st input channel	Unsigned8	rw	N	0x07	Input interrupt trigger for 1st analog input channel
	...	...	...	...	...	...	...
	0x24	Trigger 24th input channel	Unsigned8	rw	N	0x07	Input interrupt trigger for 24th analog input channel

Dieses Objekt legt fest, welches Ereignis einen Interrupt eines bestimmten Kanals auslösen soll. Die gesetzten Bits der untenstehenden Liste verweisen auf den Interrupt Trigger.

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

**Analog Input  
Interrupt Source**

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

Dieses Objekt legt fest, welcher Kanal den Interrupt verursacht hat. Gesetzte Bits verweisen auf die Nummer des Kanals, der den Interrupt verursacht hat. Die Bits werden automatisch zurückgesetzt, nachdem sie von einem SDO gelesen oder durch ein PDO versandt wurden.

- 1 = Interrupt verursacht
- 0 = kein Interrupt 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:

- Inhibit-Zeit (Objekt 0x1800, Subindex 3)
- Grenzwertüberwachung (Objekte 0x6424 + 0x6425)
- 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 1st input channel	Unsigned32	rw	N	0x00000000	Upper limit value for 1st analog input channel
	...	...	...	...	...	...	...
	0x24	Upper limit 24th input channel	Unsigned32	rw	N	0x00000000	Upper limit value for 24th 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 1st input channel	Unsigned32	rw	N	0x00000000	Lower limit value for 1st analog input channel
	...	...	...	...	...	...	...
	0x24	Lower limit 24th input channel	Unsigned32	rw	N	0x00000000	Lower limit value for 24th 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 1st input channel	Unsigned32	rw	N	0x00000002	Delta value for 1st analog input channel
	...	...	...	...	...	...	...
	0x24	Delta value 24th input channel	Unsigned32	rw	N	0x00000002	Delta value for 24th 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	1st analog output block	Unsigned8	rw	N	0xFF	1st error mode analog output block
	...	...	...	...	...	...	...
	0x24	36th analog output block	Unsigned8	rw	N	0xFF	36th 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 = auf Fehlerwert 0x6444 setzen

### Analog Output Error Value

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6444	0x00	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available analog output blocks
	0x01	1st analog block	Unsigned16	rw	N	0x0000	1st analog output block
	...	...	...	...	...	...	...
	0x24	36th analog block	Unsigned16	rw	N	0x0000	36th 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 an file error)

## IM 253CAN - CANopen-Slave - 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

## IM 253CAN - CANopen-Slave - NMT - Netzwerk Management

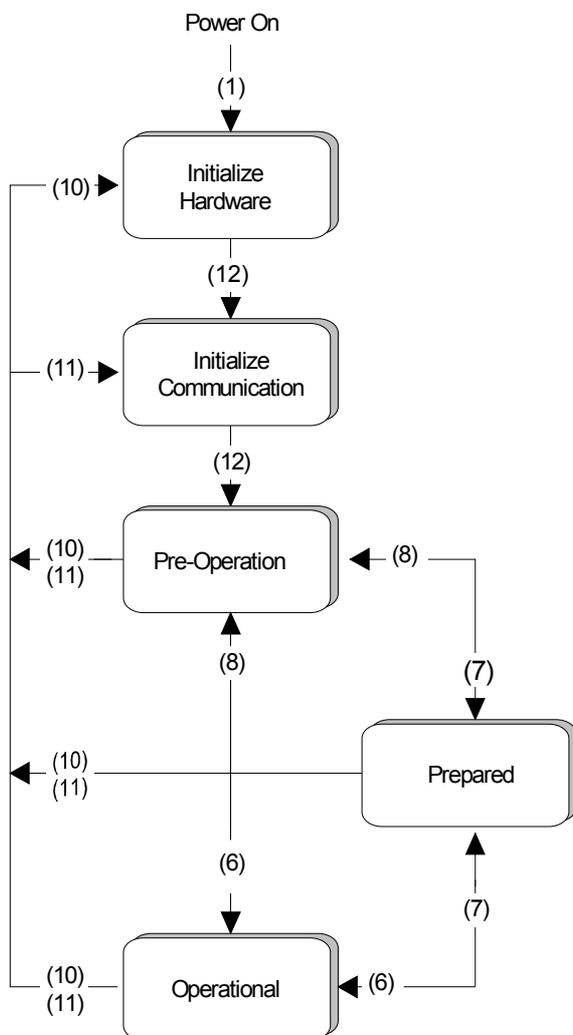
Das Netzwerkmanagement (NMT) spezifiziert globale Dienste für Netzwerküberwachung und -Management. Dazu gehört neben dem An- und Abmelden einzelner Teilnehmer auch die Überwachung der Teilnehmer während des Betriebs- und die Behandlung von Ausnahmezuständen.

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-Master IM 208 CAN

Elektrische Daten	VIPA 208-1CA00
Spannungsversorgung	über Rückwandbus
Stromaufnahme (Nennwert)	1A
Potenzialtrennung	≥ AC 500V
Statusanzeige	über LEDs auf der Frontseite
Anschlüsse/Schnittstellen	9poliger SubD-Stecker      CANopen-Ankopplung
CAN-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)
Kombination mit Peripheriemodulen	
max. Anzahl Slaves	125
max. Anzahl TxPDOs	40
max. Anzahl RxPDOs	40
max. Anzahl Eingangs-Byte	384
max. Anzahl Ausgangs-Byte	384
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x78
Gewicht	110g

**CANopen-Koppler  
IM 253CAN**

Elektrische Daten	VIPA 253-1CA01	VIPA 253-1CA30 - ECO
Spannungsversorgung	DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil	
Stromaufnahme (Leerlauf)	50mA	50mA
Stromaufnahme (Nennwert)	max. 0,8A	max. 0,3A
Ausgangsstrom Rückwandbus	max. 3,5A	max. 0,8A
Verlustleistung	2W	1,5W
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.	Je Koppler max. 8 E/A-Module frei kombinierbar
max. Teilnehmeranzahl	127 Stationen (je nach Masteranschaltung)	
Kombination mit Peripheriemodulen		
max. Modulanzahl	32 (abhängig von der Stromaufnahme)	8
max. Eingänge/Ausgänge	je 80Byte (80Byte = 10 PDOs à 8Byte)	
Maße und Gewicht		
Abmessungen (BxHxT) in mm	25,4x76x78	
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. 5A
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 (20,4...28,8V) intern über CAN-Koppler versorgt
Ausgangsstrom je Kanal	1A (Summenstrom max. 4A)
Statusanzeige	Power (PW) Sicherung intakt, Error (ER) Kurzschluss, Überlast
Programmierdaten	
Ausgabedaten	3Byte
Maße und Gewicht	
Abmessungen (BxHxT) in mm	50,8x76x78
Gewicht	150g

## Teil 6 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 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
- Technische Daten

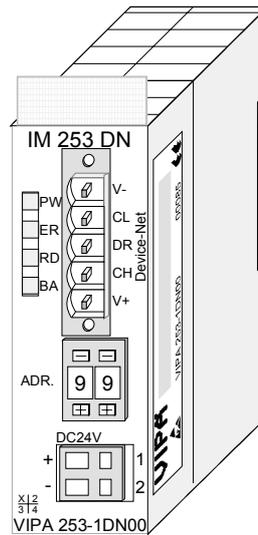
### Inhalt

Thema	Seite
<b>Teil 6 DeviceNet</b> .....	<b>6-1</b>
Systemübersicht.....	6-2
Grundlagen .....	6-3
IM 253DN - DeviceNet-Koppler - Aufbau.....	6-5
Projektierung unter Einsatz des DeviceNet-Managers .....	6-8
Einstellung von Baudrate und DeviceNet-Adresse .....	6-9
Test am DeviceNet-Bus .....	6-10
Module im DeviceNet-Manager parametrieren .....	6-11
I/O-Adressierung des DeviceNet-Scanners .....	6-16
Diagnose.....	6-17
Technische Daten .....	6-22

## Systemübersicht

Mit dem DeviceNet-Koppler von VIPA können Sie bis zu 32 Module Ihrer System 200V Peripherie an DeviceNet ankopplern.

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.

<b>Übertragungs- medium</b>	<p>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.</p> <p>Die Stichleitungen können bis zu 6m lang sein, wobei der Gesamtumfang aller Stichleitungen von der Baudrate abhängt.</p> <p>Netzknoten können ohne Unterbrechung des Netzwerks entfernt oder hinzugefügt werden. Es wird automatisch erkannt, ob ein Teilnehmer ausgefallen oder neu am Netz ist.</p> <p>DeviceNet verwendet als Übertragungsmedium eine abgeschirmte Fünfdrahtleitung.</p> <p>DeviceNet arbeitet mit Spannungsdifferenzen und ist daher unempfindlicher gegenüber Störeinflüssen als eine Spannungs- oder Stromschnittstelle.</p> <p>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.</p>
<b>Buszugriffs- verfahren</b>	<p>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).</p> <p>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.</p> <p>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.</p>
<b>Adressierung</b>	<p>Alle Teilnehmer am Bus müssen eindeutig über ein ID-Adresse identifizierbar sein. Jedes DeviceNet-Gerät besitzt eine Möglichkeit zur Adresseinstellung.</p>
<b>EDS-Datei</b>	<p>Zur Konfiguration einer Slave-Anschaltung in Ihrem eigenen Projektierool bekommen Sie die Leistungsmerkmale der DeviceNet-Geräte in Form einer EDS-Datei (<b>E</b>lectronic <b>D</b>ata <b>S</b>heet) mitgeliefert.</p>

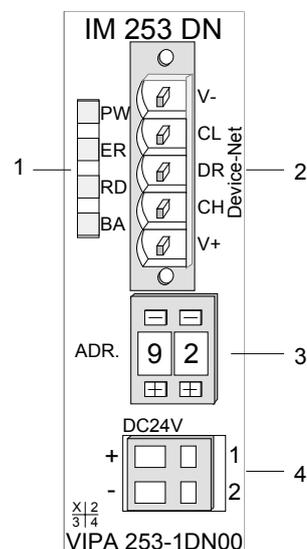
## IM 253DN - DeviceNet-Koppler - Aufbau

### Eigenschaften

Der DeviceNet-Koppler IM 253DN ermöglicht die einfache Anbindung von dezentralen Peripheriemodulen über das DeviceNet-Protokoll.

- 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

### Frontansicht 253-1DN00



- [1] LED Statusanzeige
- [2] Anschluss DeviceNet
- [3] Adresseinsteller
- [4] Anschluss 24V  
Versorgungsspannung

## Komponenten

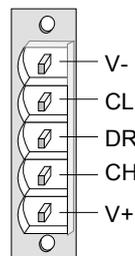
### LEDs

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	grün	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.

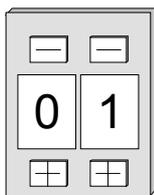


- [V-] GND Betriebsspannung
- [CL] CAN low
- [DR] DRAIN
- [CH] CAN HIGH
- [V+] DC 24V Betriebsspannung

### 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.

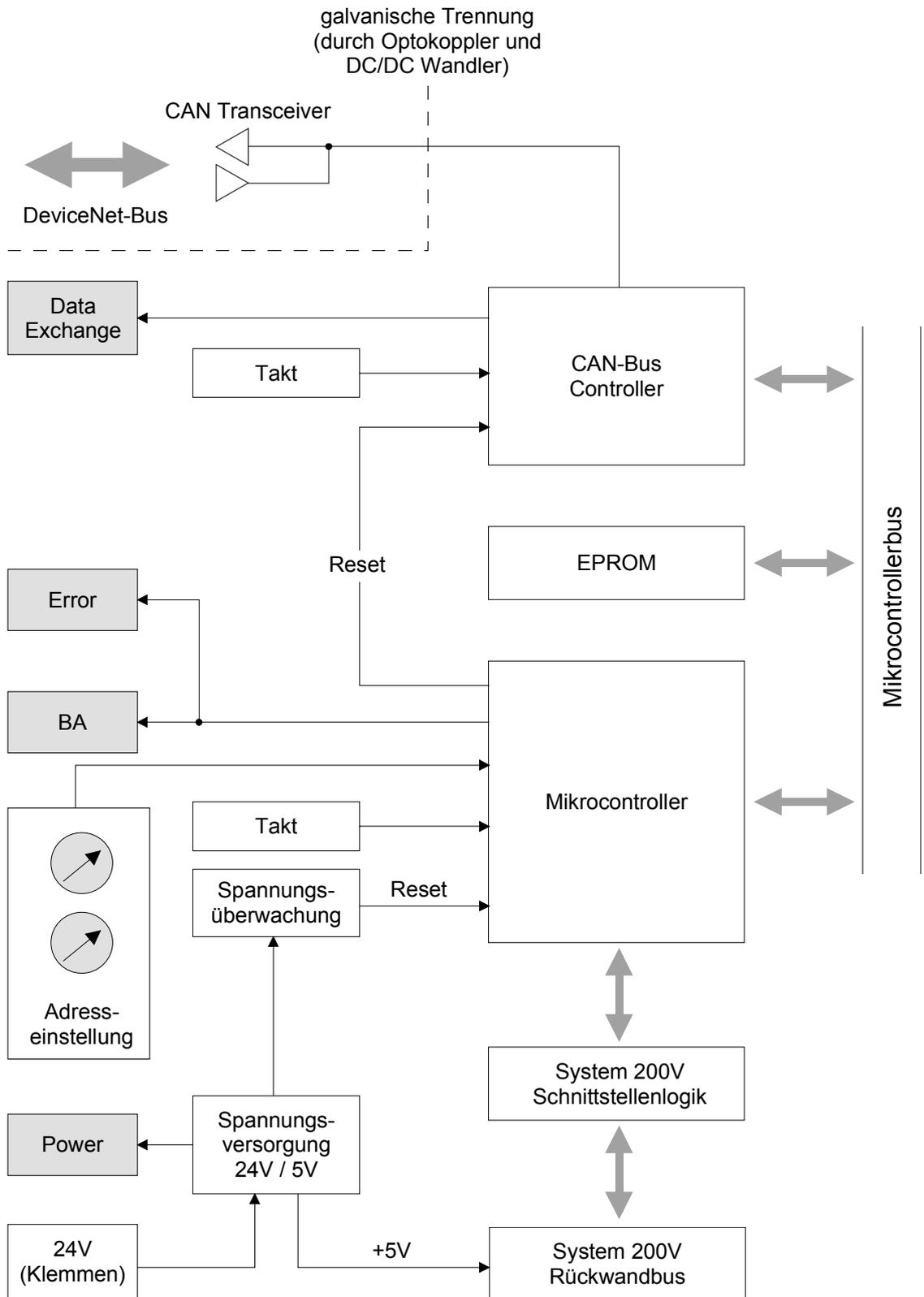
DeviceNet und Rückwandbus sind galvanisch voneinander getrennt.



#### 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

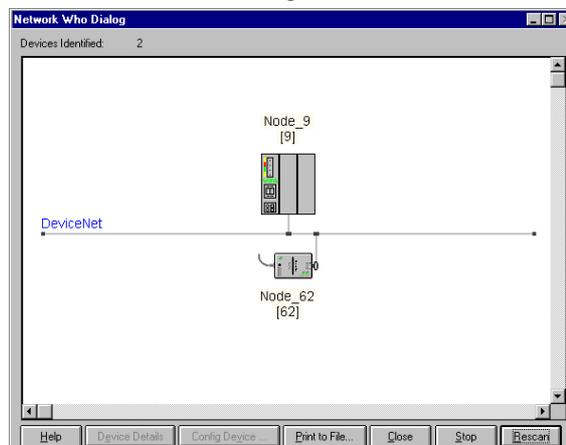
Bei einer falschen oder bereits vorhandenen Adresse leuchtet nach Power On die ER-LED (rot).

## 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

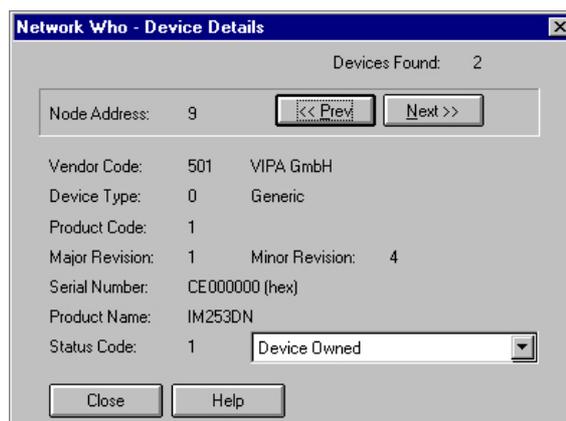
*Es öffnet sich das folgende Netzwerkfenster:*



### 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:

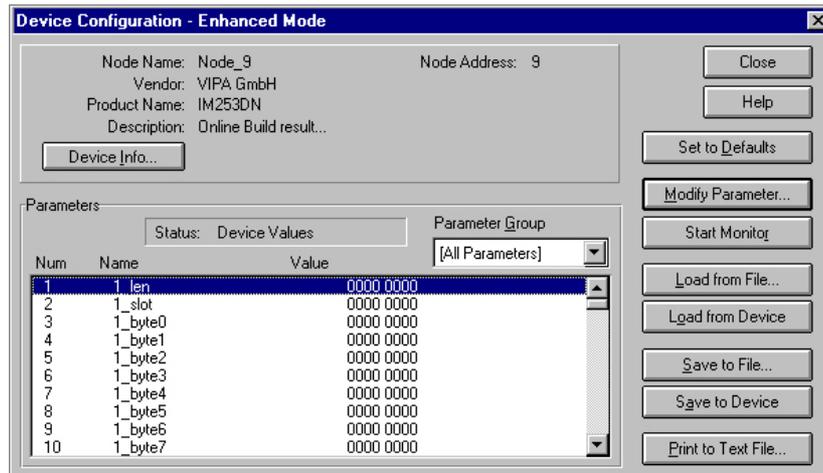
- In DeviceNet werden Parameterdaten in Form von Gruppen verwaltet.
- Maximal kann jeder DeviceNet-Koppler 144Byte Parameterdaten verarbeiten und speichern.
- Die 144Byte sind aufgeteilt in 8 Gruppen zu je 18Byte.
- Jede Gruppe darf die Parameterdaten für 1 Modul beinhalten.
- Die Gruppen sind durch 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

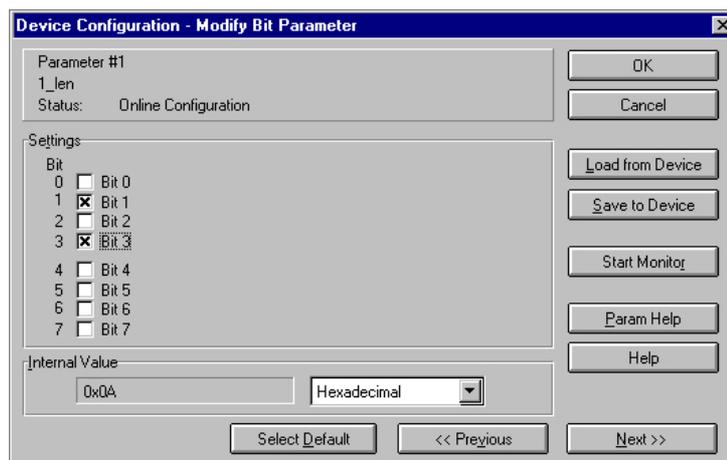
Voraussetzung: Ihr IM 253DeviceNet-Koppler befindet sich aktiv am Bus.

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:*



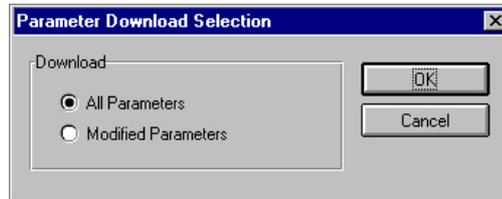
- Suchen Sie in der Parameterliste eine freie Gruppe (Value=0000 0000). Durch Einstellung von "All Parameters" im Auswahlfeld *Parameter Group* können Sie alle 8 Gruppen in der Parameterliste ausgeben.
- Doppelklicken Sie auf den "Len"-Parameter.  
*Es öffnet sich das folgende Dialogfenster:*



- 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.

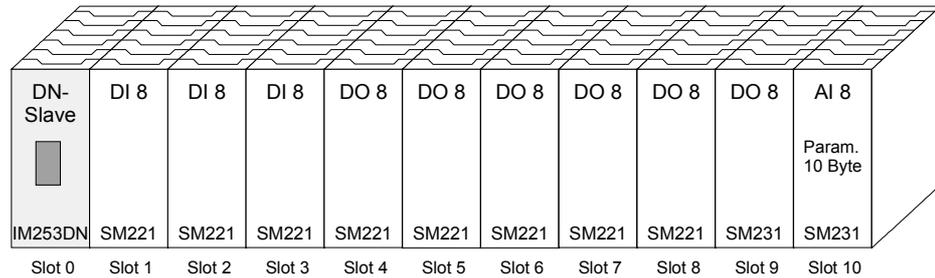
- Während der Datenübertragung erhalten Sie als Status-Text die Meldung "Status: downloading". Sobald die Übertragung beendet ist, wechselt der Status-Text in "Status: Device Values"
- Bei Abfrage der "Device Details" sieht man nun, dass der Status zusätzlich das Bit CONFIGURED enthält.



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:



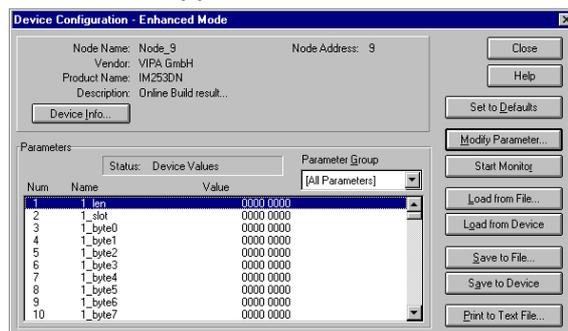
Das Beispiel zeigt einen DeviceNet-Koppler mit 10 Modulen, wobei die Module auf den Steckplätzen 1 bis 9 nicht parametrierbar sind.

Nachfolgend ist die Parametrierung des Analog-Moduls auf Steckplatz 10 beschrieben:

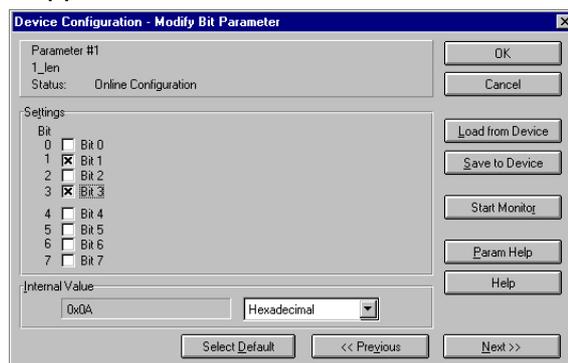
Voraussetzung: - Das Beispiel ist aufgebaut und aktiv am Bus.

- 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 5 ... 0: 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 parametrisierten 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 checked="" 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 checked="" type="checkbox"/> PW ein <input checked="" type="checkbox"/> ER ein <input checked="" type="checkbox"/> RD blinkt <input checked="" 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 checked="" type="checkbox"/> PW ein <input type="checkbox"/> ER aus <input checked="" type="checkbox"/> RD ein <input checked="" 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 checked="" 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. <b>Hinweis!</b> 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 1Sekunde 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 checked="" 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 checked="" type="checkbox"/> BA ein	

**Parametrier-Modus**

**POWER ON im Parametrier-Modus**

LED	Bedeutung
<input checked="" 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 checked="" 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 checked="" 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 checked="" type="checkbox"/> BA ein	

## 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 Busses 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, 500kBaude bis 500m max. 64
Kombination mit Peripheriemodulen	
Modulanzahl Eingänge Ausgänge	max. 32 max. 256Byte max. 256Byte
Mechanische Daten	
Abmessungen (BxHxT) Gewicht	25,4x76x78mm 80g

## Teil 7 SERCOS - Ersatzteil

### 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 7 SERCOS - Ersatzteil</b> .....	<b>7-1</b>
Systemübersicht.....	7-2
Grundlagen .....	7-3
IM 253Sercos - SERCOS-Koppler - Aufbau .....	7-5
Grundparametrierung über Adresseinsteller.....	7-8
SERCOS Identifier .....	7-10
Beispiel zur automatischen ID-Zuweisung.....	7-13
Technische Daten .....	7-22



### Hinweis!

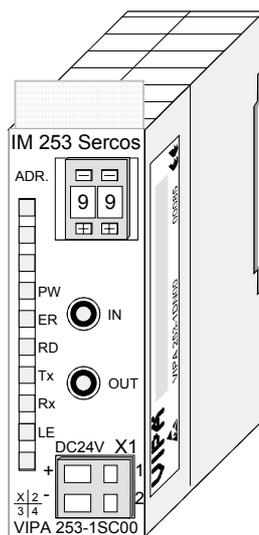
Für den Einsatz des in diesem Kapitel beschriebenen SERCOS-Kopplers werden fundierte Kenntnisse im Umgang mit SERCOS vorausgesetzt.

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 ankopeln.  
 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**

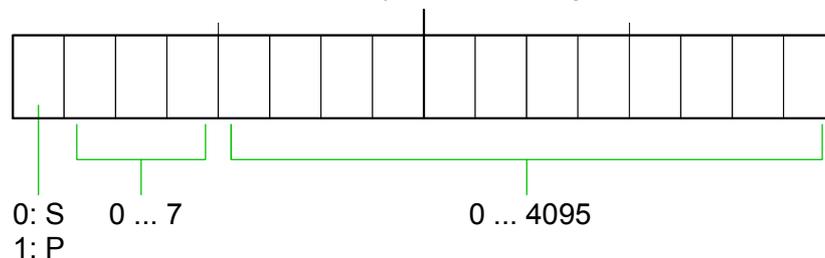
Alle Teilnehmer am Bus müssen eindeutig über ein Adresse identifizierbar sein. Jedes SERCOS-Gerät besitzt eine Möglichkeit zur Adresseinstellung.

**ID-Nummer für Datenaustausch**

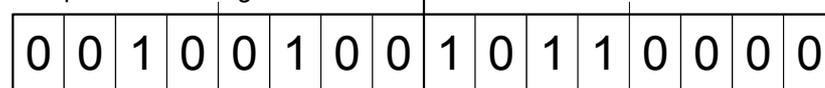
Die Adressierung der Daten beim bedarfsgesteuerten Datenaustausch und die Definition der Echtzeitdaten erfolgt bei SERCOS mittels Ident-Nummern. Für die ID-Nummern ist die ein Zahlenbereich von  $2^{16}$  festgelegt, der sich in zwei Bereiche aufteilt:

- 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

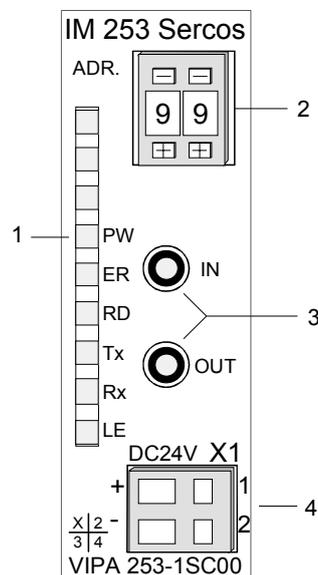
### Eigenschaften

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

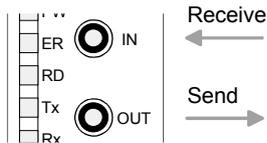
## Komponenten

### 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)

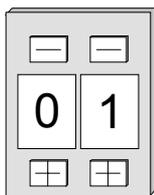
### LWL-Anschluss SERCOS



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

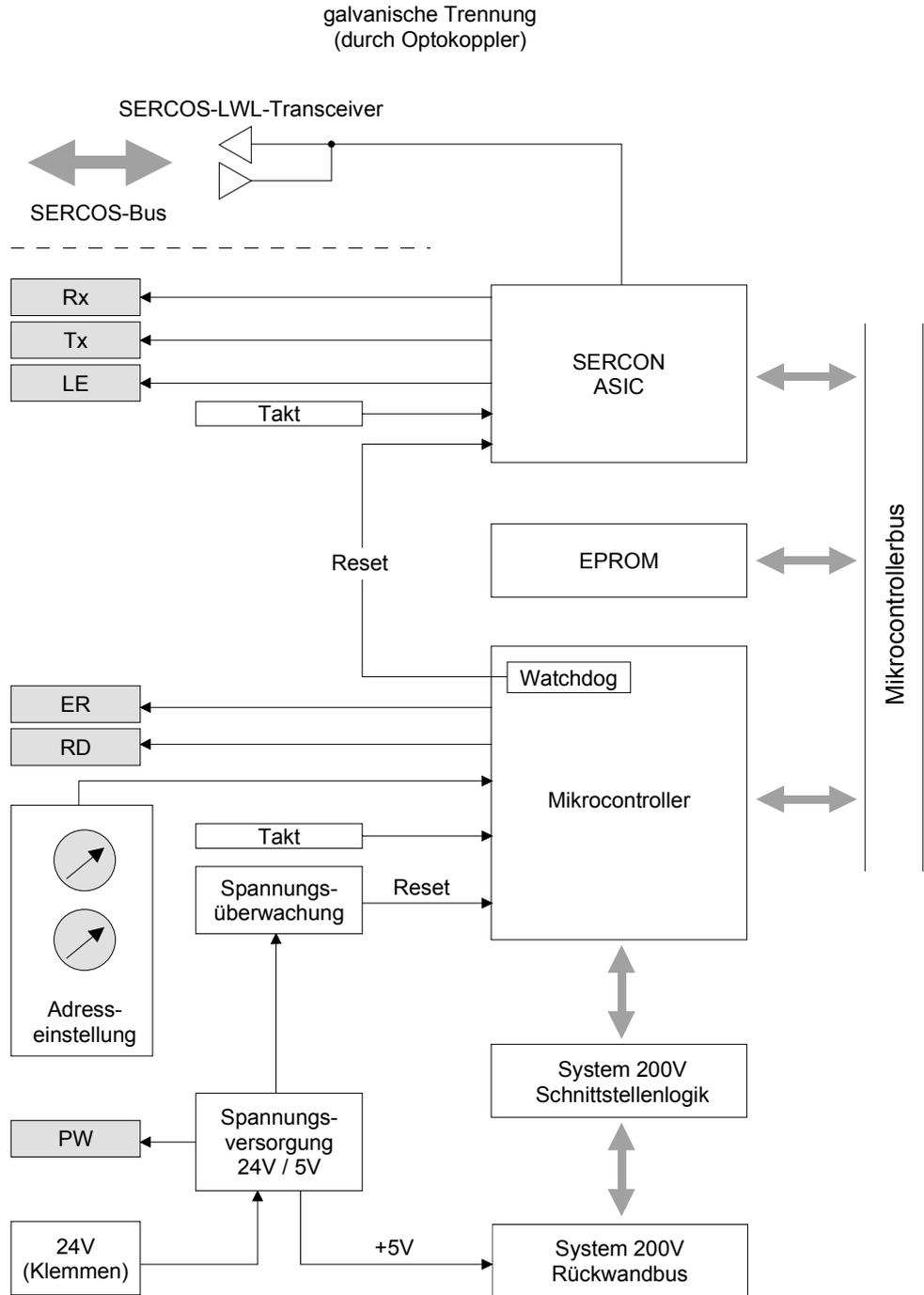
Der SERCOS-Koppler besitzt ein eingebautes Netzteil. Das Netzteil ist gegen Verpolung und Überstrom geschützt.

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 lesenden 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)

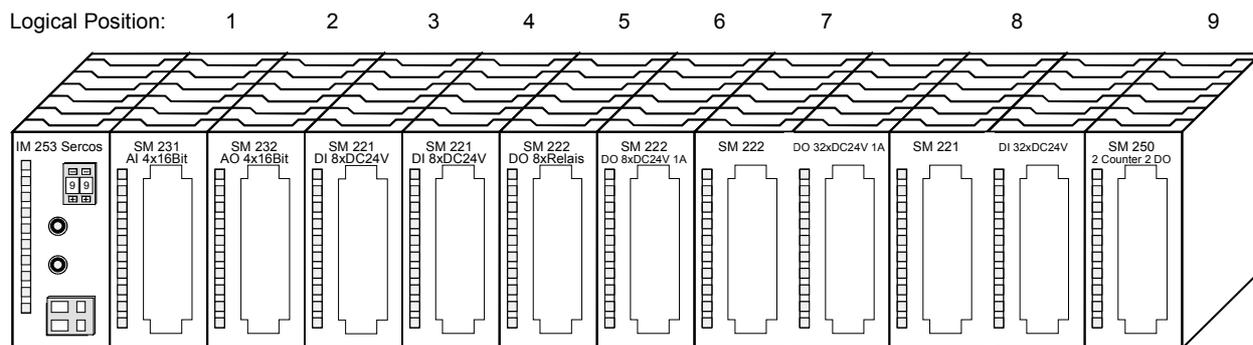
Die Längenangabe und eine Beschreibung der zu übergebenden Parameter finden Sie in den entsprechenden Kapiteln zu den Modulen in diesem Handbuch.

## 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 232-1BD51 (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	10dez
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 (232-1BD51) 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	25,4x76x78mm 75g

## Teil 8 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 8 Ethernet-Koppler .....</b>	<b>8-1</b>
Systemübersicht.....	8-2
Grundlagen Ethernet.....	8-3
Planung eines Netzwerks .....	8-7
IM 253NET - Ethernet-Koppler - Aufbau.....	8-9
Zugriffsmöglichkeiten auf den Ethernet-Koppler.....	8-11
Prinzip der automatischen Adressierung .....	8-14
Projektierung unter WinNCS .....	8-15
Diagnose und Test mittels Internet Browser.....	8-16
ModbusTCP .....	8-20
Modbus-Funktionscodes .....	8-21
Siemens S5 Header Protokoll .....	8-26
Prinzip der Alarmbearbeitung .....	8-28
Programmierbeispiel .....	8-32
Technische Daten .....	8-33

## 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.

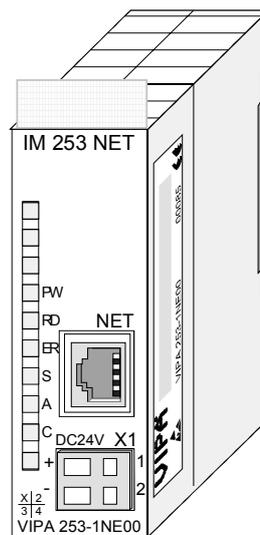
Slave-Systeme hingegen sind "Datensammler", die dem anfragenden Master die E/A-Daten der angesteckten Module zur Verfügung stellen.

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.

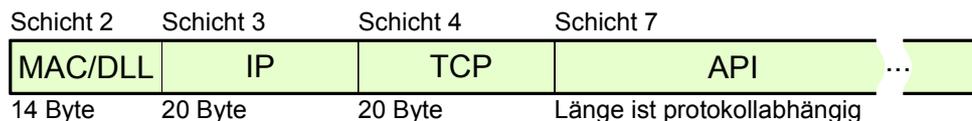
Für Test und Diagnose stellt der Ethernet-Slave einen Web-Server zur Verfügung, der lesenden und schreibenden Zugriff auf die E/A-Peripherie sowie die Parametrierung von Modulen erlaubt.

## 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**

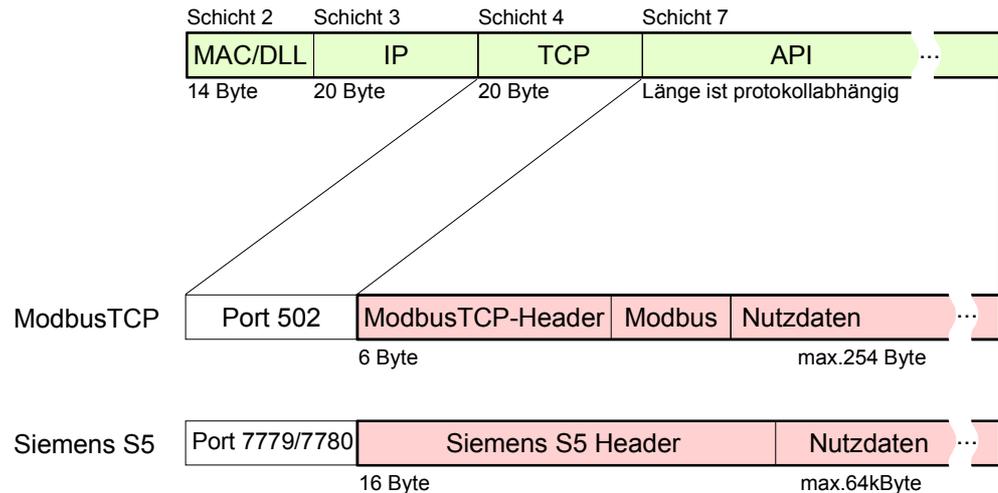
API steht für **A**pplication **P**rogramming **I**nterface. API erfüllt die Vorgaben für den Application Layer (Schicht 7).

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

### Allgemeines

Das Hauptkennzeichen einer Busstruktur ist, dass nur ein einziger physikalischer Übertragungsweg existiert. Als physikalisches Übertragungsmedium wird dabei verwendet:

- ein oder mehrere elektrische Leitungen (verdrillte Leitung)
- Koaxialkabel (Triaxialkabel)
- Lichtwellenleiter

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.

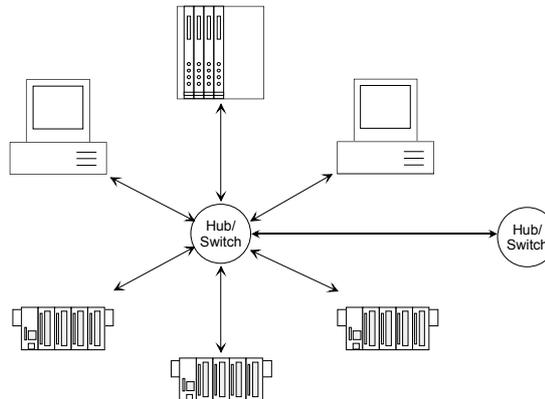
### Normen und Richtlinien

Folgende Normen und Richtlinien im Zusammenhang mit Netzwerktechnologien sind von internationalen und nationalen Gremien festgelegt worden:

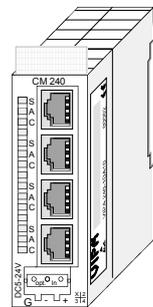
ANSI	American National Standards Institute Hier werden zur Zeit in der ANSI X3T9.5 Vereinbarungen für LANs mit hohen Übertragungsgeschwindigkeiten (100 MB/s) auf Glasfaserbasis formuliert. (FDDI) Fibre Distributed Data Interface.
CCITT	Committee Consultative Internationale de Telephone et Telegraph. 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 Commission. 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 E/A-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.
- Umfangreiche Alarmbearbeitung
- Web-Server für Test und Diagnose integriert
- RJ45-Buchse 100BaseTX, 10BaseT
- Automatische Polaritäts- und Geschwindigkeitserkennung (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

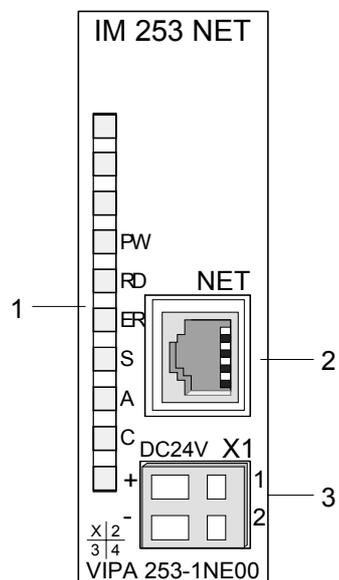


### Achtung!

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 253NET



- [1] LED Statusanzeigen
- [2] RJ45-Buchse für Twisted Pair Anschluss
- [3] Anschluss für DC 24V Spannungsversorgung

**Komponenten**

**LEDs**

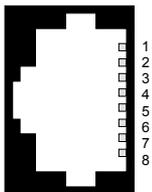
Der Ethernet-Koppler besitzt verschiedene LEDs, die der Diagnose dienen und den eigenen Betriebszustand anzeigen. Die Verwendung und die jeweiligen Farben dieser LEDs finden Sie in der nachfolgenden Tabelle.

Bez.	Farbe	Bedeutung
PW	Grün	Power: Signalisiert eine 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	-

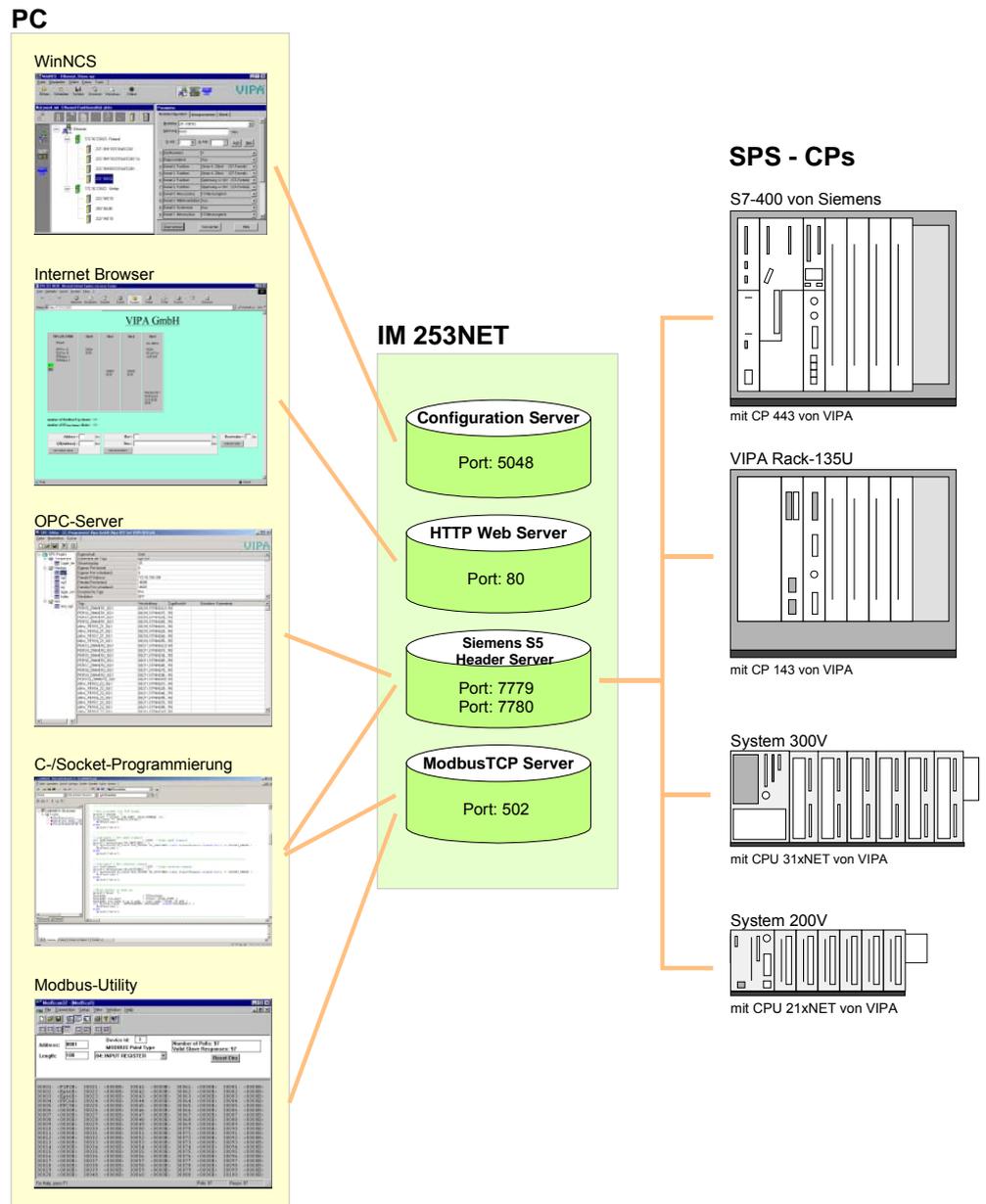
**Spannungsversorgung**

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 dieses 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 anzusteuern, 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.

Zusätzlich wird zur Alarmbearbeitung hinter den 256Byte großen E/A-Daten das "Alarm information image" mit einer Größe von 520Byte abgelegt.

### 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.



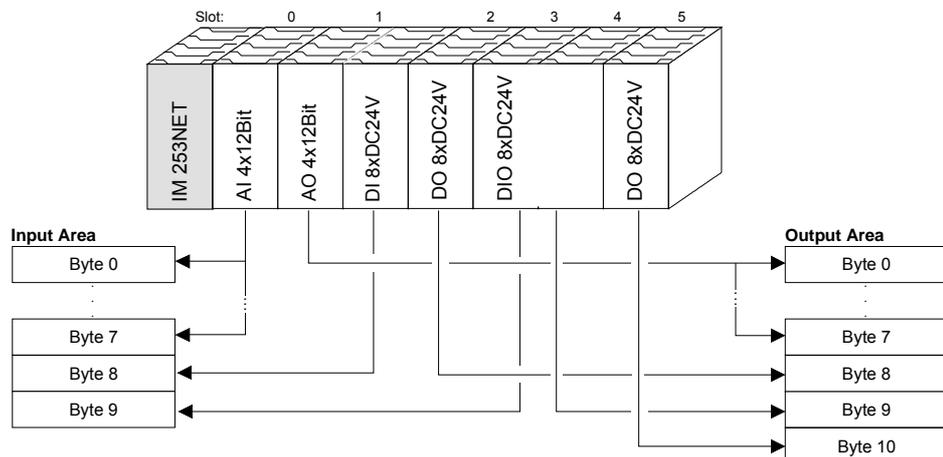
### Hinweis!

Eine 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 VIPA\_ETH200V.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: VIPA\_ETH200V.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
- Parametrierung und Diagnosedaten 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	Prm(len10)= 00 00 2d 2d 28 28 00 00 00 00
<input type="checkbox"/> RDY <input type="checkbox"/> ERR					Diag= 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Informationen über angebundene Clients

Number of Modbus/TCP clients-<2>: [172.16.131.31] [172.16.131.55]

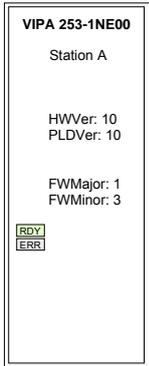
Number of S5 from Siemens clients: <1>: [172.16.131.10]

Elemente für den aktiven Zugriff auf den Ethernet-Koppler

Password = <input type="text"/> Address = <input type="text"/> dec QB[Address] = <input type="text"/> hex <input type="button" value="set output value"/>	Password = <input type="text"/> Slot = <input type="text"/> dec Prm = <input type="text"/> hex <input type="button" value="set parameters"/>	Password = <input type="text"/> Resetvalue = <input type="text"/> dec <input type="button" value="reboot node"/>	Password = <input type="text"/> Timeout = <input type="text"/> msec <input type="button" value="set timeout"/>	Password = <input type="text"/> Slot = <input type="text"/> dec <input type="button" value="confirm alarm"/>
--	---	--	--	--

[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.

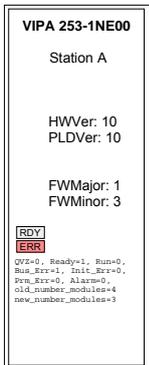
*PLDVer:* Das PLD (**P**rogrammable **L**ogik **D**evice) ist ein programmierbarer Logik-Baustein, der die Kommunikation zwischen Rückwandbus und Prozessor steuert.

*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 Hauptversions-Nummer.

*RDY, ERR:* Zustandsanzeige der LEDs RD und ER

*rdy* (Kleinbuchstaben): LED blinkt / *RDY* (Großbuchstaben): LED leuchtet  
Solange der Ethernet-Koppler fehlerfrei kommuniziert, bleibt die Zustandsanzeige wie oben gezeigt. Im Fehlerfall erscheint unterhalb von ERR beispielsweise folgende Meldung:

**Fehleranzeige**



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

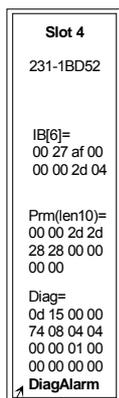
**Bereich Module  
Slot 0 ... 31**

In diesem Bereich werden alle Informationen zur Ein-/Ausgabe-Peripherie dargestellt wie Modulname, Ein-/Ausgabe-Belegung, falls vorhanden Parameterbytes und Diagnosedaten.

*Konfiguration (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 Eingangs-Bereich 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).

Alarm liegt an:  
DiagAlarm oder ProcAla

Die mit *Prm()* = Parameterbytes beinhalten folgende Informationen:

- Die Länge des Parameterblocks steht in Klammern mit einem vorangestellten *len*.
- Die Byte-Inhalte zeigen die Parameterbytes des entsprechenden Moduls. *DIAG* = zeigt 16Byte Diagnosedaten für die Alarmbearbeitung.

**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 <> 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. Jedes Steuerelement ist passwortgeschützt. Verwenden Sie das für Ihren Koppler projektierte Passwort (default = "00000000"). Folgende 5 Steuerelemente stehen zur Verfügung:

- Ausgänge steuern
- Modul parametrieren
- Reset des Ethernet-Kopplers ausführen
- Timeout konfigurieren
- Alarm quittieren

Password =	<input type="text"/>	
Address =	<input type="text"/>	dec
QB[Address] =	<input type="text"/>	hex
set output value		

*Ausgänge steuern*

Mit diesem Steuerelementen können Sie einen gewünschte Ausgabeadressbereich 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 Address 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
    
```

Password =	<input type="text"/>	
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 mus 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.



Password =	<input type="text"/>
Resetvalue =	<input type="text"/> dec
<input type="button" value="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, 3 oder 4. Andere Werte werden ignoriert!

- Resetvalue= 1 Reboot des Kopplers (Defaulteinstellung)
- Resetvalue= 2 Löschen aller Modul-Konfigurationen (Modulnamen) und Reboot des Kopplers
- Resetvalue= 3 Löschen aller Modul-Parameter (nicht Konfiguration) und Reboot des Kopplers
- Resetvalue= 4 Reset Passwort (auf Default-Wert "00000000")

**Passwort  
Rücksetzen**

Das Rücksetzen des Passwortes auf den Default-Wert "00000000" ist über folgende Vorgehensweise möglich:

- Schalten Sie die Spannungsversorgung Ihres Ethernet-Kopplers aus und ziehen Sie diesen vom Rückwand-Bus ab.
- Schalten Sie die Spannungsversorgung des Kopplers wieder ein.
- Starten Sie Ihren Web-Browser und rufen Sie über die IP-Adresse die Web-Seite des Ethernet-Kopplers auf.
- Geben Sie unter dem Parameter "reboot node" das Passwort "00000000" ein.
- Setzen Sie "Resetvalue =" auf 4 ein und klicken Sie auf [reboot node].  
→ Der Ethernet-Koppler bootet neu und das Passwort wird auf den Default-Wert "00000000" zurückgesetzt.

Password =	<input type="text"/>
Timeout =	<input type="text"/> msec
<input type="button" value="set timeout"/>	

*Timeout konfigurieren*

Der Koppler verfügt über ein Verbindungs-Timeout. Wird der Wert 0 übergeben, so ist diese Funktion deaktiviert. (Im Bild des Ethernet-Kopplers steht "Timout: off").



**Hinweis!**

Wählen Sie "Timout: off", wenn Sie per Internet Browser Ausgänge steuern möchten, da sonst nach Ablauf des Timeouts alle Ausgänge in den sicheren Zustand 0 gebracht werden. Bei Timeout-Zeiten > 0msec muss eine IO-Verbindung schneller als der Zeitwert lesen / schreiben aufgebaut werden. Ist dies nicht der Fall, so werden die Verbindungen abgebaut und die Ausgänge auf den sicheren Zustand 0 gesetzt. Die RD LED blinkt und auf der Web-Site ist "rdy" in Kleinbuchstaben zu sehen.

Password =	<input type="text"/>
Slot =	<input type="text"/> dec
<input type="button" value="confirm alarm"/>	

*Alarm quittieren*

Mit "confirm alarm" können Sie den Alarm eines Moduls quittieren. Durch Vorgabe des Steckplatzes des gewünschten Moduls wird mit [confirm alarm] das Alarm-Statusbit des Moduls zurückgesetzt.

## 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*

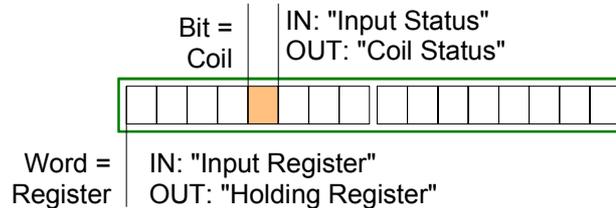
Byte	Name	Beschreibung
0	Transaction identifier (High-Byte)	wird von Server zurückgesendet (beliebig)
1	Transaction identifier (Low-Byte)	wird von Server zurückgesendet (beliebig)
2	Protocol identifier (High-Byte)	immer 0
3	Protocol identifier (Low-Byte)	immer 0
4	Length field (High-Byte)	immer 0 da 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 hoch zählen lassen und somit eine 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.

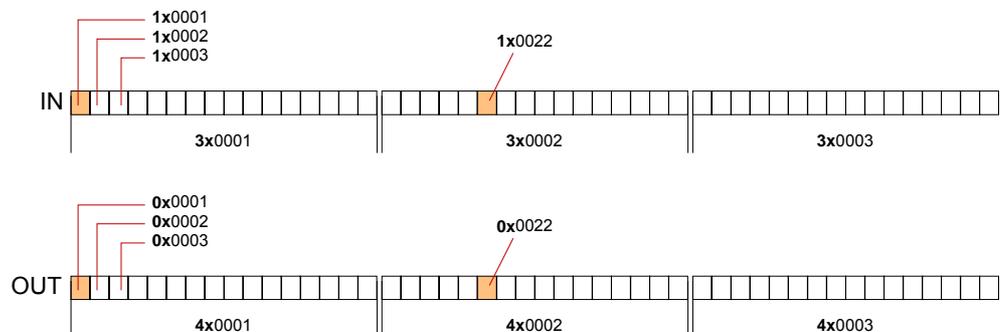
### Bereichsdefinitionen

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 Master-Ausgabe  
Zugriff über Funktions-Code 01h, 05h, 0Fh
- 1x: Bit-Bereich für Master-Eingabe  
Zugriff über Funktions-Code 02h
- 3x: Wortbereich-Bereich für Master-Eingabe  
Zugriff über Funktions-Code 04h, 17h
- 4x: Wortbereich-Bereich für Master-Ausgabe  
Zugriff über Funktions-Code 03h, 06h, 10h, 17h



Eine Beschreibung der Funktions-Codes finden Sie auf den Folgeseiten.

**Übersicht**

Mit folgende Funktionscodes können Sie von einem Modbus-Master auf einen Slave zugreifen. Die Beschreibung erfolgt immer aus Sicht des Masters:

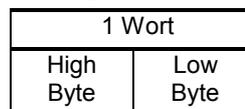
Code	Befehl	Beschreibung
01h	Read n Bits	n Bit lesen von Master-Ausgabe-Bereich 0x
02h	Read n Bits	n Bit lesen von Master-Eingabe-Bereich 1x
03h	Read n Words	n Worte lesen von Master-Ausgabe-Bereich 4x
04h	Read n Words	n Worte lesen von Master-Eingabe-Bereich 3x
05h	Write one Bit	1 Bit schreiben in Master-Ausgabe-Bereich 0x
06h	Write one Word	1 Wort schreiben in Master-Ausgabe-Bereich 4x
0Fh	Write n Bits	n Bit schreiben in Master-Ausgabe-Bereich 0x
10h	Write n Words	n Worte schreiben in Master-Ausgabe-Bereich 4x
17h	Write n Words and Read m Words	n Worte schreiben in Master-Ausgabe-Bereich 4x und in der Antwort kommen m gelesene Worte des Master-Eingabe-Bereiches 3x

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**      Diese Funktion ermöglicht das wortweise Lesen aus einem Koppler.  
**03h, 04h**

Kommandotelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse Wort	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. 126Worte

max. 255Byte

**Write a Bit**      Mit dieser Funktion können Sie ein Bit in Ihrem Koppler ändern. Eine Zustandsänderung erfolgt unter "Zustand Bit" mit folgenden Werten:  
**05h**

"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   6								
	1Byte	1Byte	1Wort	1Wort	1Byte	1Byte	1Byte	1Byte
						max. 248Byte		
	max. 255Byte							

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								
	1Byte	1Byte	1Wort	1Wort	1Byte	1Wort	1Wort	1Wort
						max. 124Worte		
	max. 255Byte							

Antworttelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Adresse 1. Wort	Anzahl der Worte
x x 0 0 0 6				
	1Byte	1Byte	1Wort	1Wort

**Write n Words und Read m Words 17h** Über diese Funktion können Sie mit einem Request n Worte schreiben und m Worte lesen.

Kommandotelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Read Adresse	Read Anzahl der Worte	Write Adresse	Write Anzahl der Worte	Write Anzahl der Bytes	Write Daten 1. Wort	Write Daten 2. Wort	...
x x 0 0 0										
	1Byte	1Byte	1Wort	1Wort	1Wort	1Wort	1Byte	1Wort	1Wort	1Wort
								max. 122Worte		
	max. 255Byte									

Antworttelegramm

ModbusTCP-Header	Slave-Adresse	Funktions-Code	Read Anzahl der Bytes	Read Daten 1. Wort	Read Daten 2. Wort	...
x x 0 0 0						
	1Byte	1Byte	1Byte	1Wort	1Wort	
					max. 126Worte	
	max. 255Byte					

## 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 folgender Form abgelegt:	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		irrelevant	irrelevant	irrelevant
Anfangsadresse Bedeutung  erlaubter Bereich:	MB0: Länge In-Bereich MB1: 00 MB2: Länge Out-Bereich MB3: 00	EB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.  0...255	AB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.  0...255	PB-Nr., ab der die Daten entnommen bzw. eingeschrieben werden.  0...65535
Anzahl Bedeutung  erlaubter Bereich:		Länge des Quell-/Ziel-datenblocks in Bytes.  1...256	Länge des Quell-/Ziel-datenblocks in Bytes.  1...256	Länge des Quell-/Ziel-datenblocks in Bytes.  1...256



## Prinzip der Alarmbearbeitung

### Übersicht

Viele nichtdigitale Module wie Analog-, Funktions- oder Master-Module aus dem System 200V können - wenn parametrierbar - im Fehlerfall Alarmdaten (alarmdata) liefern.

Sobald ein oder mehrere Module einen Alarm melden, werden die Alarmdaten des entsprechenden Steckplatzes vom Ethernet-Koppler empfangen und quittiert. Dieser setzt daraufhin in seinem internen *Alarm Information Image* (Alarmabbild) ein dem Steckplatz zugeordnetes Bit und legt die bis 16Byte langen Diagnosedaten ab.

Im System 200V wird Diagnosealarm und Prozessalarm unterschieden. Hierbei ist zu beachten, dass ein Modul zu einem Zeitpunkt immer nur einen der beiden Alarmarten auslösen kann. Zur Unterscheidung gibt es im Diagnoseabbild je ein 32Bit breites Feld (Bit 0 = Steckplatz 0 bis Bit 31 = Steckplatz 31) für Prozessalarm und Diagnosealarm. Danach folgen je Steckplatz 16Byte Alarmdaten.

Zur Quittierung können Sie auf Diagnose- und Prozessalarmstatus auch schreibend zugreifen. Auf die 16Byte Alarmdaten haben Sie nur lesenden Zugriff.

### Alarm Information Image

Das *Alarm Information Image* mit einer Größe von 520Byte liegt hinter den 256Byte E/A-Daten und hat folgenden Aufbau:

32Bit Prozessalarmstatus (Little Endian Format): Bit 0 ... Bit 31 entspricht Steckplatz 0 ... 31
32Bit Diagnosealarmstatus (Little Endian Format): Bit 0 ... Bit 31 entspricht Steckplatz 0 ... 31
16Byte Alarmdaten von Steckplatz 0
16Byte Alarmdaten von Steckplatz 1
...
16Byte Alarmdaten von Steckplatz 31

### Diagnose ausgeben

#### Web-Server

Alle alarmfähigen Module zeigen den Eintrag "Diag=" mit den aktuellen 16Byte Alarmdaten. Liegt ein neuer Alarm vor, so wird im Alarmstatus das entsprechende Bit gesetzt und auf der Website unter dem Diag-Bereich je nach Alarm-Typ entweder "DiagAlarm" für Diagnose-Alarm bzw. "ProcAlarm" Prozessalarm eingeblendet.

Slot 4
231-1BD52
IB[6]= 00 27 af 00 00 00 2d 04
Prm(len10)= 00 00 2d 2d 28 28 00 00 00 00
Diag= 0d 15 00 00 74 08 04 04 00 00 01 00 00 00 00 00
DiagAlarm

*ModbusTCP*

Lesen ab Register 3x0129:

Register	Inhalt
3x0129	Prozessalarmstatus: Byte 0, Byte 1
3x0130	Prozessalarmstatus: Byte 2, Byte 3
3x0131	Diagnosealarmstatus: Byte 0, Byte 1
3x0132	Diagnosealarmstatus: Byte 2, Byte 3
3x0133	Steckplatz 0: Alarmdaten 16Byte
3x0141	Steckplatz 1: Alarmdaten 16Byte
3x0149	Steckplatz 2: Alarmdaten 16Byte
3x0157	Steckplatz 3: Alarmdaten 16Byte
3x0165	Steckplatz 4: Alarmdaten 16Byte
3x0173	Steckplatz 5: Alarmdaten 16Byte
3x0181	Steckplatz 6: Alarmdaten 16Byte
3x0189	Steckplatz 7: Alarmdaten 16Byte
3x0197	Steckplatz 8: Alarmdaten 16Byte
3x0205	Steckplatz 9: Alarmdaten 16Byte
3x0213	Steckplatz 10: Alarmdaten 16Byte
3x0221	Steckplatz 11: Alarmdaten 16Byte
3x0229	Steckplatz 12: Alarmdaten 16Byte
3x0237	Steckplatz 13: Alarmdaten 16Byte
3x0245	Steckplatz 14: Alarmdaten 16Byte
3x0253	Steckplatz 15: Alarmdaten 16Byte
3x0261	Steckplatz 16: Alarmdaten 16Byte
3x0269	Steckplatz 17: Alarmdaten 16Byte
3x0277	Steckplatz 18: Alarmdaten 16Byte
3x0285	Steckplatz 19: Alarmdaten 16Byte
3x0293	Steckplatz 20: Alarmdaten 16Byte
3x0301	Steckplatz 21: Alarmdaten 16Byte
3x0309	Steckplatz 22: Alarmdaten 16Byte
3x0317	Steckplatz 23: Alarmdaten 16Byte
3x0325	Steckplatz 24: Alarmdaten 16Byte
3x0333	Steckplatz 25: Alarmdaten 16Byte
3x0341	Steckplatz 26: Alarmdaten 16Byte
3x0349	Steckplatz 27: Alarmdaten 16Byte
3x0357	Steckplatz 28: Alarmdaten 16Byte
3x0365	Steckplatz 29: Alarmdaten 16Byte
3x0373	Steckplatz 30: Alarmdaten 16Byte
3x0381	Steckplatz 31: Alarmdaten 16Byte

*Siemens S5 Header*

Lesen ab Peripheriebyte 256:

Byteadresse	Inhalt
256	Prozessalarmstatus: Byte 0, Byte 1
258	Prozessalarmstatus: Byte 2, Byte 3
260	Diagnosealarmstatus: Byte 0, Byte 1
262	Diagnosealarmstatus: Byte 2, Byte 3
264	Steckplatz 0: Alarmdaten 16Byte
280	Steckplatz 1: Alarmdaten 16Byte
296	Steckplatz 2: Alarmdaten 16Byte
312	Steckplatz 3: Alarmdaten 16Byte
328	Steckplatz 4: Alarmdaten 16Byte
344	Steckplatz 5: Alarmdaten 16Byte
360	Steckplatz 6: Alarmdaten 16Byte
376	Steckplatz 7: Alarmdaten 16Byte
392	Steckplatz 8: Alarmdaten 16Byte
408	Steckplatz 9: Alarmdaten 16Byte
424	Steckplatz 10: Alarmdaten 16Byte
440	Steckplatz 11: Alarmdaten 16Byte
456	Steckplatz 12: Alarmdaten 16Byte
472	Steckplatz 13: Alarmdaten 16Byte
488	Steckplatz 14: Alarmdaten 16Byte
504	Steckplatz 15: Alarmdaten 16Byte
520	Steckplatz 16: Alarmdaten 16Byte
536	Steckplatz 17: Alarmdaten 16Byte
552	Steckplatz 18: Alarmdaten 16Byte
568	Steckplatz 19: Alarmdaten 16Byte
584	Steckplatz 20: Alarmdaten 16Byte
600	Steckplatz 21: Alarmdaten 16Byte
616	Steckplatz 22: Alarmdaten 16Byte
632	Steckplatz 23: Alarmdaten 16Byte
648	Steckplatz 24: Alarmdaten 16Byte
664	Steckplatz 25: Alarmdaten 16Byte
680	Steckplatz 26: Alarmdaten 16Byte
696	Steckplatz 27: Alarmdaten 16Byte
712	Steckplatz 28: Alarmdaten 16Byte
728	Steckplatz 29: Alarmdaten 16Byte
744	Steckplatz 30: Alarmdaten 16Byte
760	Steckplatz 31: Alarmdaten 16Byte

**Diagnose  
quittieren**

*Web-Server*

Mit dem per WinNCS (Version > V320) wahlweise einblendbaren Feld "confirm alarm" können Sie einen Alarm eines Steckplatzes quittieren. Tragen Sie hierzu das Passwort des Ethernet-Kopplers ein, sowie den Steckplatz (0 ... 31) in dem das Alarmstatusbit gelöscht werden soll. Daraufhin betätigen Sie die Schaltfläche [confirm alarm] woraufhin die Website neu geladen wird und "DiagAlarm" bzw. "ProcAlarm" gelöscht sein sollte.

Password =	<input type="text"/>
Slot =	<input type="text"/> dec
confirm alarm	

*ModbusTCP*

Schreiben ab Register 4x0129:

Register	Inhalt
4x0129	Prozessalarmstatus: Byte 0, Byte 1
4x0130	Prozessalarmstatus: Byte 2, Byte 3
4x0131	Diagnosealarmstatus: Byte 0, Byte 1
4x0132	Diagnosealarmstatus: Byte 2, Byte 3

*Siemens S5 Header*

Schreiben ab Peripheriebyte 256:

Byteadresse	Inhalt
256	Prozessalarmstatus: Byte 0, Byte 1
258	Prozessalarmstatus: Byte 2, Byte 3
260	Diagnosealarmstatus: Byte 0, Byte 1
262	Diagnosealarmstatus: Byte 2, Byte 3

**Typische  
Anwendung**

Sie überwachen die beiden Alarmstatus-Doppelworte. Solange sie "0" sind liegt kein Alarm an. Sobald sie <> "0" sind liegen eines oder mehrere aktualisierte Alarmstatusfelder vor. Diese sind im Anwenderprogramm auszuwerten. Nach der Auswertung sollten Sie das Alarmstatusfeld auf "0" setzen, was einer Quittierung entspricht. Jetzt können weitere Alarme bearbeitet werden.

*Mehrere Alarme von unterschiedlichen Steckplätzen:*

Kamen von verschiedenen Steckplätzen gleichzeitig Alarme, so werden für jeden Steckplatz das Alarmstatusbit gesetzt und die Alarmdaten hinterlegt. Es kommt zu keinem Informationsverlust.

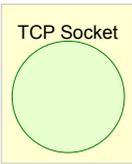
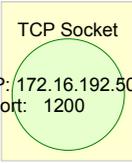
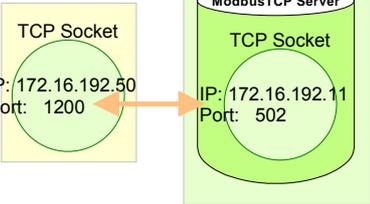
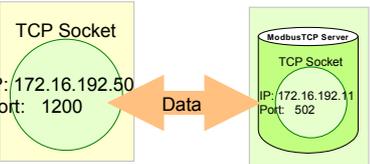
*Mehrere Alarme aus einem Steckplatz:*

Bei mehreren Alarmen von einem Steckplatz steht das entsprechende Alarmstatusbit auf "1". Es können die Alarmdaten des aktuellsten Alarms gelesen werden. Was vorher geschehen ist und wie viele Alarme auftraten kann nicht nachvollzogen werden!

# 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		
<b>1.</b> 		<b>zu 1.</b> Microsoft Socket System starten	<code>WSAStartup (wVersionRequested, &amp;wsaData);</code>
<b>2.</b> 		<b>zu 2.</b> Socket-Ressourcen für TCP reservieren	<code>m_lsock = socket (AF_INET, SOCK_STREAM, 0);</code>
<b>3.</b> 		<b>zu 3.</b> Socket an den lokalen PC anbinden	<code>SocketAddr.sin_port = htons( 0 );</code> <code>SocketAddr.sin_addr.S_un.S_addr = inet_addr( "0.0.0.0" );</code> <code>bind(m_lsock, (LPSOCKADDR) &amp;SocketAddr, sizeof(SocketAddr));</code>  Bei Aufruf von <code>bind</code> 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)
<b>4.</b> 		<b>zu 4.</b> Verbindung zu externem Gerät aufbauen	<code>SocketAddr.sin_port = htons (m_wPort);</code> <code>SocketAddr.sin_addr.S_un.S_addr = inet_addr(m_szIpAddress);</code> <code>connect(m_lsock, (LPSOCKADDR) &amp;SocketAddr, sizeof(SocketAddr));</code>
<b>5.</b> 		<b>zu 5.</b> Für schreibenden bzw. lesenden Zugriff sind je nach Protokoll entsprechende Telegramme aufzubauen und in <code>sndBuf</code> abzulegen. <code>sndBufLen</code> beinhaltet die Anzahl der zu sendenden Bytes.	
		<i>Lesender Zugriff</i> <code>sndBuf</code> senden (Request) <code>recv(m_lsock, (char *)rcvBuf, sizeof(rcvBuf), 0);</code> Telegramm in <code>rcvBuf</code> empfangen (Response+Daten)	
		<i>Schreibender Zugriff</i> <code>sndBuf</code> senden (Request+Daten) <code>recv(m_lsock, (char *)rcvBuf, sizeof(rcvBuf), 0);</code> Telegramm in <code>rcvBuf</code> empfangen (Response)	
<b>6.</b> 		<b>zu 6.</b> Socket wieder schließen	<code>closesocket(m_lsock);</code>

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	25,4x76x78
Gewicht	70g



## Teil 9 Buserweiterung IM 260 - IM 261

### Überblick

In diesem Kapitel wird die Buserweiterung beschrieben, die das Aufteilen einer System 200V Zeile auf bis zu 4 Zeilen ermöglicht. Hierbei darf die maximale Anzahl von 32 Modulen nicht überschritten werden.

Nachfolgend sind beschrieben:

- Einsatzbereich
- Vorgehensweise bei der Verkabelung
- LEDs
- Technische Daten

### Inhalt

Thema	Seite
<b>Teil 9 Buserweiterung IM 260 - IM 261 .....</b>	<b>9-1</b>
Einsatzbereich.....	9-2
Verkabelung.....	9-3
Statusanzeigen .....	9-4
Technische Daten .....	9-5

### Bestelldaten

Typ	Bestellnummer	Beschreibung
IM 260	VIPA 260-1AA00	Basisanschaltung IM 260
IM 261	VIPA 261-1CA00	Zeilenanschaltung IM 261
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

## Einsatzbereich

### Übersicht

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!

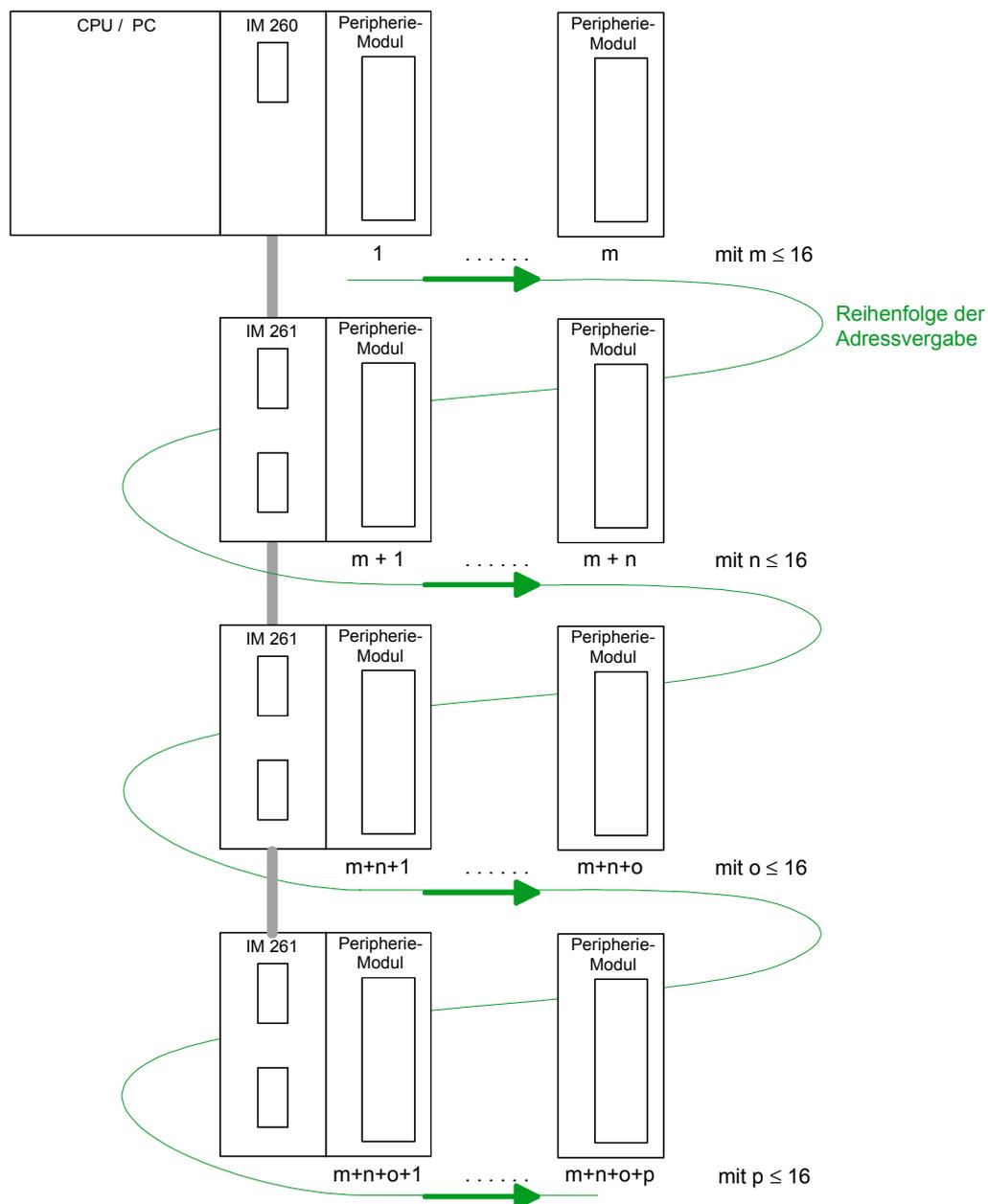
Für den Einsatz der Buserweiterung gibt es gewisse Regeln, die zu beachten sind:

- 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!

# Verkabelung

## Aufbau

In der nachfolgenden Abbildung ist der Aufbau einer Buserweiterung unter Beachtung der Aufbauregeln aufgeführt:



Es gilt:  $m + n + o + p \leq 32$



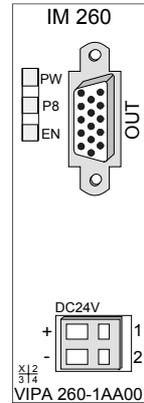
### Hinweis!

Die Buserweiterung darf nur bei Verwendung des PC 288 (VIPA 288-2BL10) oder einer CPU (auch Kombi-CPU's) 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

## Statusanzeigen

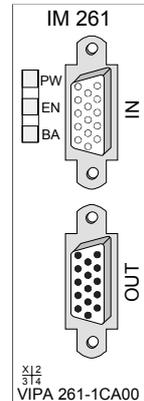
### Statusanzeige Basisanschaltung IM 260

LED	Farbe	Beschreibung
PW	grün	Versorgungsspannung liegt an
P8	gelb	Versorgungsspannung für nachfolgende Zeilen aktiv
EN	gelb	Rückwandbus-Kommunikation aktiv



### Statusanzeige Zeilenanschaltung IM 261

LED	Farbe	Beschreibung
PW	grün	Versorgungsspannung über IM 260 liegt an
EN	gelb	Rückwandbus-Kommunikation aktiv
BA	rot	Befehlsausgabesperre (BASP) aktiv



## Technische Daten

### IM 260 Basisanschaltung

Elektrische Daten	VIPA 260-1AA00
Spannungsversorgung	DC 24V (20,4...28,8) extern über Front
Stromaufnahme	1,9A
Stromaufnahme Rückwandbus	30mA
max. Kabellänge zw. 1. und letzter Zeile	2,5m
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x78
Gewicht	80g

### IM 261 Zeilenanschaltung

Elektrische Daten	VIPA 261-1CA00
Spannungsversorgung	über IM 260
Stromversorgung Rückwandbus	max. 1,5A pro Zeile (Summe max. 4A)
max. Kabellänge zw. 1. und letzter Zeile	-
Maße und Gewicht	
Abmessungen (BxHxT) in mm	25,4x76x78
Gewicht	50g



## Anhang

### A Index

<b>A</b>	
Abisolierlängen .....	3-81
Adresseinsteller	
CANopen-Slave .....	5-41, 5-46, 5-51
ECO .....	5-41
DeviceNet-Koppler.....	6-6
Profibus-DP-Slave .....	3-37, 3-59
ECO .....	3-59
SERCOS.....	7-6
<b>B</b>	
Baudrate	
CANopen-Slave .....	5-51
DeviceNet-Koppler.....	6-9
SERCOS.....	7-8
Betriebszustände	
CANopen-Master .....	5-16
Interbus.....	4-4
Profibus-Master .....	3-16
Buserweiterung .....	9-1
Hinweise .....	9-2
Technische Daten.....	9-5
Verkabelung .....	9-3
<b>C</b>	
CANopen .....	5-1
Buszugriff.....	5-5
Grundlagen .....	5-4
Master.....	5-6
Aufbau.....	5-6
Betriebsarten.....	5-16
Fehlermeldungen .....	5-22
Firmwareupdate .....	5-15
Gerätemodell .....	5-20
GSD einbinden.....	5-9
Objektverzeichnis.....	5-24
PDO .....	5-20
Projektierung.....	5-8
Schnelleinstieg .....	5-8
SDO .....	5-21
SFC 219 .....	5-21
Technische Daten .....	5-104
Telegrammaufbau.....	5-19
WinCoCT-Einsatz .....	5-10
Slave.....	5-38
Adresseinsteller .....	5-51
Aufbau .....	5-38
Baudrate .....	5-51
Blockschaltbild .....	5-42
DO 24xDC 24V .....	5-43
ECO .....	5-38, 5-41
Fehlermeldungen.....	5-49, 5-59
Funktionscodes.....	5-53
Heartbeat .....	5-103
Identifizier .....	5-47, 5-55
Kommunikationsarten.....	5-56
Modul-ID .....	5-51
Modultypen .....	5-70
NMT .....	5-102
Node Guarding .....	5-103
Objektverzeichnis .....	5-60
PDO .....	5-54
Linking .....	5-56
Schnelleinstieg.....	5-47
SDO .....	5-58
Statusanzeigen .....	5-39
Technische Daten.....	5-105
Telegrammaufbau .....	5-52
Verkabelung.....	5-40, 5-46
COB-ID .....	5-55
<b>D</b>	
DeviceNet .....	6-1
Adressierung .....	6-4
Buszugriff .....	6-4
Grundlagen.....	6-3
Koppler .....	6-5
Adresse.....	6-9
Anschluss .....	6-6
Aufbau .....	6-5
Baudrate .....	6-9
Beispiel .....	6-14
Blockschaltbild .....	6-7
Diagnose.....	6-17
Parametrierung .....	6-11
Projektierung.....	6-8

- Test ..... 6-10
- Manager ..... 6-8
- Scanner ..... 6-16
- Diagnosefunktionen
  - CANopen Salve ..... 5-101
  - DeviceNet ..... 6-17
  - DP-V0-Slave ..... 3-49
  - DP-V1-Slave ..... 3-68
  - Ethernet-Koppler ..... 8-16
  - DP-Zyklus ..... 3-9
- E**
  - EasyConn ..... 3-81
  - EDS-Datei ..... 6-4
  - Emergency Objekt ..... 5-49, 5-101
  - Ethernet ..... 8-1
    - Grundlagen ..... 8-3
    - Koppler ..... 8-9
      - Adressierung ..... 8-14
      - Alarmbearbeitung ..... 8-28
      - Aufbau ..... 8-9
      - Auslieferungszustand ..... 8-9
      - Diagnose ..... 8-16
      - GSD einbinden ..... 8-15
      - ModbusTCP ..... 8-20
        - Bereiche ..... 8-21
        - Funktionscodes ..... 8-21
      - OPC-Server ..... 8-12
      - ORG-Format ..... 8-26
      - Passwort ..... 8-19
      - Projektierung ..... 8-11, 8-15
      - Siemens S5 Header ..... 8-26
      - Socketprogrammierung ..... 8-32
      - SPS-Header ..... 8-27
      - Technische Daten ..... 8-33
      - Web-Server ..... 8-12, 8-16
      - Zugriffsmöglichkeiten ..... 8-11
    - ModbusTCP ..... 8-6
    - Netzwerkplanung ..... 8-7
    - Protokolle ..... 8-4
    - Siemens S5 Header ..... 8-6
  - F**
    - Fehlermeldungen
      - CANopen-Master: ..... 5-22
      - CANopen-Slave ..... 5-49, 5-59
      - DeviceNet-Koppler ..... 6-17
      - DP-V0-Slave ..... 3-52
      - DP-V1-Slave ..... 3-72
  - Interbus ..... 4-16
  - G**
    - Grundlagen
      - CANopen ..... 5-4
      - DeviceNet ..... 6-3
      - Ethernet ..... 8-3
      - Interbus ..... 4-3
      - Profibus-DP ..... 3-5
      - SERCOS ..... 7-3
  - H**
    - Heartbeat ..... 5-49, 5-103
    - Hub ..... 8-3
  - I**
    - ID-Code ..... 4-13
    - ID-Länge ..... 4-13
    - ID-Register ..... 4-3
    - Inbetriebnahme
      - CANopen-Master ..... 5-8
      - CANopen-Slave ..... 5-47
      - DeviceNet coupler ..... 6-8
      - Interbus ..... 4-15
      - Profibus ..... 3-89
    - Intel-Format ..... 3-64
    - Interbus ..... 4-1
      - Betriebsarten ..... 4-4
      - Datenkonsistenz ..... 4-14
      - Datenübertragung ..... 4-5
      - Grundlagen ..... 4-3
      - Koppler ..... 4-7
        - Anschluss ..... 4-10
        - Fehlermeldungen ..... 4-16
        - Inbetriebnahme ..... 4-15
        - Prozessabbild ..... 4-11
        - Technische Daten ..... 4-18
      - Master ..... 4-3
      - Master konfigurieren ..... 4-17
  - L**
    - LEDs
      - Buserweiterung ..... 9-4
      - CANopen-Master ..... 5-7
      - CANopen-Slave ..... 5-39, 5-43, 5-44
      - DeviceNet-Koppler ..... 6-6, 6-17
      - Ethernet-Koppler ..... 8-10
      - Interbus-Koppler ..... 4-7
      - Profibus-DP-Master ..... 3-14
      - Profibus-DP-Slave ..... 3-36, 3-58

- SERCOS-Koppler ..... 7-6
- Leitungsabschluss ..... 3-80
- Lichtwellenleiter LWL ..... 3-12, 3-82
  - Schnittstelle ..... 3-14
- M*
- min\_slave\_interval ..... 3-9
- MMC ..... 3-15
- ModbusTCP ..... 8-20
- Motorola-Format ..... 3-64
- N*
- NMT ..... 5-102
- Node Guarding ..... 5-48, 5-103
- Normdiagnose-Daten ..... 3-69
- P*
- Parametrierung
  - DeviceNet-Koppler ..... 6-11
  - Profibus-DP-V0-Slave ..... 3-48
  - Profibus-DP-V1-Slave ..... 3-63
  - SERCOS ..... 7-8
- PDO ..... 5-54
- Profibus-DP ..... 3-1
  - Adressierung ..... 3-12
  - Aufbaurichtlinien ..... 3-79
  - Beispiele ..... 3-91
  - Datenkonsistenz ..... 3-7
  - Datenverkehr ..... 3-7
  - Diagnose-LEDs ..... 3-90
  - DP-V0-Slave ..... 3-35
    - Anlauf ..... 3-47
    - Aufbau ..... 3-35
    - Blockschaltbild ..... 3-45
    - Datenübertragung ..... 3-8
    - Diagnose ..... 3-49
    - DO 24xDC 24V ..... 3-38
    - Fehlermeldungen ..... 3-52
    - Grundlagen ..... 3-5
    - GSD einbinden ..... 3-46
    - Parameter ..... 3-48
    - Projektierung ..... 3-46
    - redundant ..... 3-42
  - DP-V1-Slave ..... 3-56
    - Adressierung ..... 3-65
    - Alarm ..... 3-73
    - Anlauf ..... 3-62
    - Aufbau ..... 3-56
    - Blockschaltbild ..... 3-60
- Diagnose ..... 3-68
- ECO ..... 3-56, 3-59
- Fehlermeldungen ..... 3-72
- Firmware-Update ..... 3-76
- Grundlagen ..... 3-5, 3-10
- GSD einbinden ..... 3-61
- I&M-Daten ..... 3-77
- Modulkonfiguration ..... 3-67
- Parameter ..... 3-63
- Projektierung ..... 3-61
- Einschränkungen ..... 3-8
- Grundlagen ..... 3-5
- Inbetriebnahme ..... 3-89
- Lichtwellenleiter LWL ..... 3-82
- Master ..... 3-6, 3-13
  - 2bf-Datei ..... 3-22
  - Aufbau ..... 3-13
  - Betriebszustände ..... 3-16
  - Einsatz an CPU 21x ..... 3-17
  - Firmwareupdate ..... 3-33
  - Flash-ROM ..... 3-25
  - MSAC\_C1/2 ..... 3-6
  - Projektierung ..... 3-18
  - Slave-Betrieb ..... 3-28
  - Urlöschen ..... 3-32
  - WInNCS-Einsatz ..... 3-23
  - wld-Datei ..... 3-22
- Multi Master System ..... 3-86
- Netze ..... 3-85
- Optischer Profibus ..... 3-87
- Optoelektrischer Profibus ..... 3-88
- Redundanzstatus ..... 3-55
- Slave ..... 3-6
- Stecker ..... 3-81
- Technische Daten ..... 3-99
- Token-Passing-Verfahren ..... 3-7
- Übertragungsmedium ..... 3-12
- Übertragungsprotokoll ..... 3-7
- Protokolle ..... 8-4
- Prozessabbild
  - CANopen-Master ..... 5-17
  - DeviceNet-Koppler ..... 6-16
  - Ethernet-Koppler ..... 8-14
  - Interbus ..... 4-11
  - Profibus ..... 3-17
- S*
- Schieberegister ..... 4-3

Schnittstelle	
DeviceNet-Koppler.....	6-6
Interbus-Koppler .....	4-8
RJ45 Ethernet-Koppler .....	8-10
RS485 CANopen-Master .....	5-7
RS485 CANopen-Slave .....	5-40
RS485 Profibus-DP-Master ....	3-14
RS485 Profibus-DP-Slave .....	3-36
SERCOS-Koppler .....	7-6
SDO .....	5-58
Segmentlänge unter Profibus ....	3-79
SERCOS.....	7-1
Adressierung .....	7-4
Blockschaltbild .....	7-7
Grundlagen .....	7-3
Koppler .....	7-5
Anschluss.....	7-6
Aufbau.....	7-5
Baudrate.....	7-8
Beispiel.....	7-13
Identifizier .....	7-10
Parametrierung .....	7-8
Technische Daten .....	7-22
SIP-Tool.....	3-27
Spannungsversorgung	
CANopen-Master .....	5-7
CANopen-Slave .....	5-41
ECO .....	5-41
DeviceNet-Koppler.....	6-6
Ethernet-Koppler.....	8-10
Interbus-Koppler .....	4-8
Profibus-DP-Master .....	3-15
Profibus-DP-Slave .....	3-37
ECO .....	3-59
SERCOS.....	7-6
Switch .....	8-3
System 200V	
Aufbaurichtlinien .....	2-12
Betriebssicherheit .....	1-5
Busverbinder.....	2-2
Demontage .....	2-7
Dezentrales System.....	1-4
Einbaumaße .....	2-10
EMV .....	2-12
Grundregeln .....	2-13
Grundlagen.....	1-1
Komponenten .....	1-4
Montage .....	2-1, 2-5
Peripheriemodule .....	1-4
Projektierung .....	1-4
Schirmung von Leitungen.....	2-14
Sicherheitshinweise.....	1-2
Störeinträge.....	2-12
Tragschienen.....	2-2
Übersicht .....	1-3, 1-5
Umgebungsbedingungen .....	1-5
Verdrahtung.....	2-8
Zentrales System .....	1-4
Systemübersicht	
CANopen .....	5-2
DeviceNet.....	6-2
Ethernet.....	8-2
Interbus .....	4-2
Profibus-DP .....	3-2
SERCOS .....	7-2
<b>T</b>	
Technische Daten	
Buserweiterung .....	9-5
CANopen-Master.....	5-104
CANopen-Slave.....	5-105
DeviceNet-Koppler .....	6-22
Ethernet-Koppler .....	8-33
Interbus-Koppler.....	4-18
Profibus-DP-Master.....	3-99
Profibus-DP-Slave.....	3-101
SERCOS-Koppler.....	7-22
Twisted Pair.....	8-3
Einschränkungen.....	8-8
<b>V</b>	
V-Bus-Zyklus .....	3-9
<b>W</b>	
WinCoCT.....	5-10
WinNCS	
mit Ethernet-Koppler .....	8-15
unter Profibus-DP.....	3-23
<b>Z</b>	
Zyklische Datenübertragung.....	3-8

M.Stich