



Handbücher/Manuals



**VIPA**  
**Gesellschaft für Visualisierung**  
**und Prozessautomatisierung mbH**

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

# Handbuch

## VIPA System 100V

### SM-CAN

Best.-Nr.: VIPA HB100D\_SM-CAN  
Rev. 06/14





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

© Copyright 2006 VIPA, Gesellschaft für Visualisierung und Prozessautomatisierung mbH  
Ohmstraße 4, D-91074 Herzogenaurach,  
Tel.: +49 (91 32) 744 -0  
Fax.: +49 (91 32) 744-144  
EMail: info@vipa.de  
<http://www.vipa.de>

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

Alle Rechte vorbehalten

### **Haftungsausschluss**

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

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

Für Verbesserungsvorschläge sind wir dankbar.

### **Warenzeichen**

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

SIMATIC, STEP und S7-300

sind eingetragene Warenzeichen der Siemens AG.

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

## Über dieses Handbuch

Das Handbuch beschreibt die bei VIPA erhältliche dezentrale Blockperipherie CAN-Bus des System 100V. Hier finden Sie neben einer Produktübersicht eine detaillierte Beschreibung der einzelnen Komponenten.

Sie erhalten Informationen für den Anschluss und die Handhabung der dezentralen Blockperipherie in einem CAN-System.

### Überblick

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

Im Rahmen dieser Einleitung erfolgt die Vorstellung des System 100V von VIPA als zentrales bzw. dezentrales Automatisierungssystem.

Neben einer Systemübersicht finden Sie hier allgemeine Hinweise zum System 100V wie Maße, Montage und Betriebsbedingungen.

#### **Teil 2: Dezentrale Blockperipherie CAN-Bus**

Dieses Kapitel beschreibt die Dezentrale Blockperipherie mit integriertem CAN-Bus-Koppler. Es liefert auch, in Form eines Schnelleinstiegs, alle Informationen, die für den Einsatz unter CANopen erforderlich sind.

## 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 System 100V .....	1-3
Allgemeine Beschreibung System 100V .....	1-4
Einbaumaße .....	1-5
<b>Teil 2 Dezentrale Blockperipherie CAN-Bus CANopen</b> .....	<b>2-1</b>
Grundlagen von CANopen .....	2-2
Systemübersicht.....	2-5
Aufbau.....	2-6
Montage und Verdrahtung.....	2-12
Anschlussbilder .....	2-13
Einsatz unter CANopen.....	2-14
Schnelleinstieg .....	2-15
Einstellung von Baudrate und Modul-ID .....	2-18
Telegrammaufbau .....	2-19
PDO - Prozessdaten-Objekt.....	2-21
SDO - Servicedaten-Objekt.....	2-24
Objekt-Verzeichnis .....	2-26
Emergency Object.....	2-40
NMT - Netzwerk Management.....	2-41
Technische Daten .....	2-43
<b>Anhang</b> .....	<b>A-1</b>
Index .....	A-1



## Benutzerhinweise

<b>Zielsetzung und Inhalt</b>	Dieses Handbuch beschreibt das System 100V. Beschrieben wird Aufbau, Projektierung und Anwendung des System 100V.
<b>Zielgruppe</b>	Das Handbuch ist geschrieben für Anwender mit Grundkenntnissen in der Automatisierungstechnik und in der SPS-Programmierung.
<b>Aufbau des Handbuchs</b>	Das Handbuch ist in Kapitel gegliedert. Jedes Kapitel beschreibt eine abgeschlossene Thematik.
<b>Orientierung im Dokument</b>	Als Orientierungshilfe stehen im Handbuch zur Verfügung: <ul style="list-style-type: none"><li>• Gesamt-Inhaltsverzeichnis am Anfang des Handbuchs</li><li>• Übersicht der beschriebenen Themen am Anfang jedes Kapitels</li><li>• Stichwortverzeichnis (Index) am Ende des Handbuchs</li></ul>
<b>Verfügbarkeit</b>	Das Handbuch ist verfügbar in: <ul style="list-style-type: none"><li>• gedruckter Form auf Papier</li><li>• in elektronischer Form als PDF-Datei (Adobe Acrobat Reader)</li></ul>
<b>Piktogramme Signalwörter</b>	Besonders wichtige Textteile sind mit folgenden Piktogrammen und Signalworten ausgezeichnet:

**Gefahr!**

Unmittelbar drohende oder mögliche Gefahr.  
Personenschäden sind möglich.

**Achtung!**

Bei Nichtbefolgen sind Sachschäden möglich.

**Hinweis!**

Zusätzliche Informationen und nützliche Tips.

## Sicherheitshinweise

### Bestimmungsgemäße Verwendung

Das System 100V ist konstruiert und gefertigt für

- 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 System 100V 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 der Module nationale Vorschriften beachten!**

## Teil 1 Grundlagen

### Überblick

Im Rahmen dieser Einleitung folgen Hinweise im Umgang und Informationen über Einsatzbereiche und Anwendung des System 100V.

Nachfolgend sind beschrieben:

- Sicherheitshinweise für den Benutzer
- Systemübersicht
- Montage und Umgebungsbedingungen

### Inhalt

Thema	Seite
<b>TEIL 1 Grundlagen</b> .....	<b>1-1</b>
Sicherheitshinweise für den Benutzer .....	1-2
Übersicht System 100V .....	1-3
Allgemeine Beschreibung System 100V.....	1-4
Einbaumaße.....	1-5

## Sicherheitshinweise für den Benutzer

### Handhabung elektrostatisch gefährdeter Baugruppen

VIPA-Baugruppen sind mit hochintegrierten Bauelementen in MOS-Technik bestückt. Diese Bauelemente sind hoch empfindlich gegenüber Überspannungen, die z.B. bei elektrostatischer Entladung entstehen. Zur Kennzeichnung dieser gefährdeten Baugruppen wird nachfolgendes Symbol verwendet:



Das Symbol befindet sich auf Baugruppen, Baugruppenträgern oder auf Verpackungen und weist so auf elektrostatisch gefährdete Baugruppen hin. Elektrostatisch gefährdete Baugruppen können durch Energien und Spannungen zerstört werden, die weit unterhalb der Wahrnehmungsgrenze des Menschen liegen. Hantiert eine Person, die nicht elektrisch entladen ist, mit elektrostatisch gefährdeten Baugruppen, können 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 System 100V

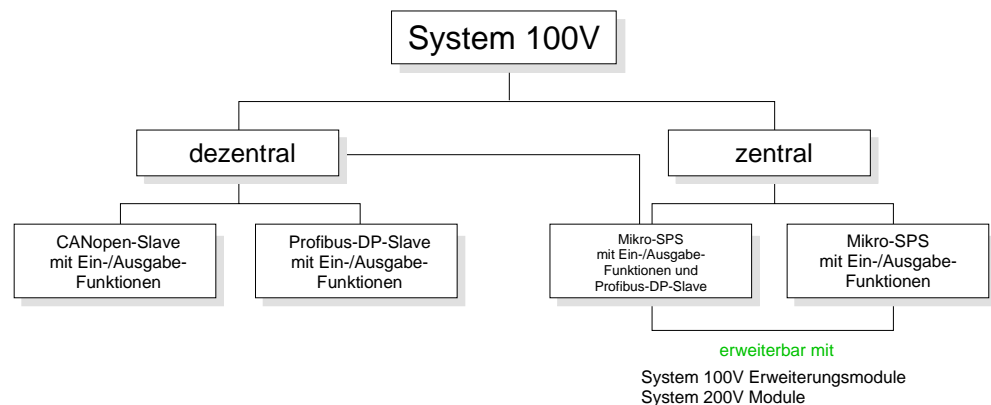
### Allgemeines

Das System 100V von VIPA ist ein kompaktes zentral wie dezentral einsetzbares Automatisierungssystem von VIPA. Das System ist ausgelegt für Anwendungen im unteren und mittleren Leistungsbereich.

Bei einem System 100V Modul sind CPU bzw. Buskoppler oder CPU und Buskoppler zusammen mit Ein-/Ausgabefunktionen in einem Gehäuse integriert.

System 100V Module werden direkt auf eine 35 mm Normprofilschiene montiert.

Sie können bei der Mikro-SPS die Anzahl der E/As mit bis zu 4 Erweiterungsmodulen vergrößern bzw. über Busverbinder System 200V Module ankoppeln. Die nachfolgende Abbildung soll Ihnen den Leistungsumfang des System 100V vermitteln:



### Zentrales System

Das zentrale System besteht aus einer CPU und integrierten E/A-Funktionen. Die CPU ist befehlskompatibel zur S7-300 von Siemens und kann mit den S7-Programmierertools von Siemens und VIPA via MPI programmiert und projektiert werden.

Über Busverbinder können Sie bis zu 4 Module aus der System 200V Familie ankoppeln bzw. die Anzahl der E/As durch Anschluss von bis zu 4 System 100V Erweiterungsmodulen vergrößern.

Die CPUs sind in verschiedenen Varianten verfügbar.

### Zentrales System mit DP-Slave

Dieses System besitzt neben CPU und integrierten E/A-Funktionen zusätzlich einen Profibus-DP-Slave, der sich in den CPU-Adressbereich einblendet.

### Dezentrales System

Beim dezentralen System ist an Stelle der CPU ein Profibus-DP-Slave bzw. ein CANopen-Slave mit E/A-Funktionen integriert. Das System ist nicht erweiterungsfähig.

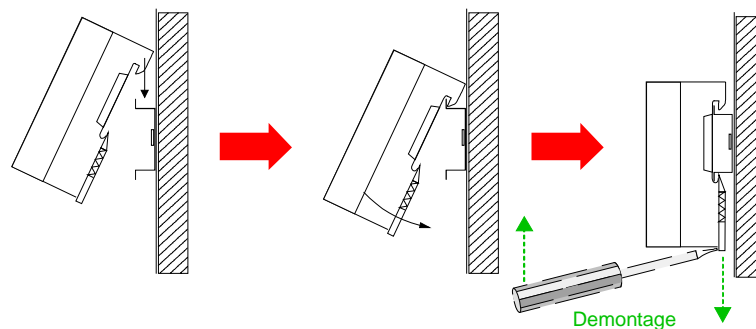
## Allgemeine Beschreibung System 100V

### Aufbau/Maße

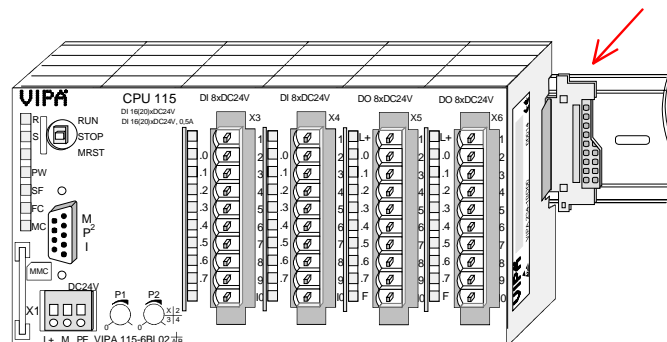
- Normprofil-Hutschiene 35mm
- Maße Grundgehäuse:  
4fach breit: (BxHxT) in mm: 101,6x76x48 in Zoll: 4x3x1,9  
6fach breit: (BxHxT) in mm: 152,4x76x48 in Zoll: 6x3x1,9

### Montage

Die Montage eines System 100V Moduls erfolgt durch Aufschnappen auf eine Normprofil-Hutschiene.



Jedem Erweiterungsmodul liegt ein 1fach Busverbinder bei. Bei Einsatz von Erweiterungsmodulen ist vor der Montage auf der rechten Seite von hinten der 1fach Busverbinder anzustecken.



### Betriebssicherheit

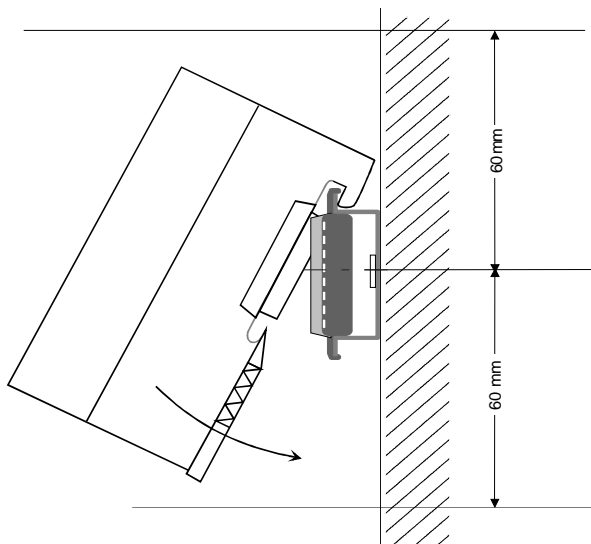
- Anschluss über Federzugklemmen, Aderquerschnitt 0,08...2,5mm<sup>2</sup>
- Vollisolierung der Verdrahtung bei Modulwechsel
- EMV-Festigkeit ESD/Burst gemäß IEC 61000-4-2 / IEC 61000-4-4 (bis Stufe 3)
- Schockfestigkeit gemäß IEC 60068-2-6 / IEC 60068-2-27 (1G/12G)

### Umgebungsbedingungen

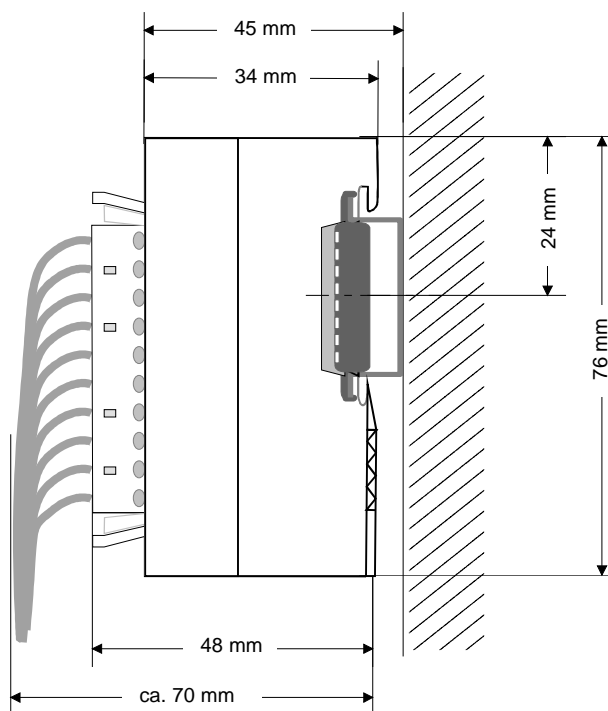
- Betriebstemperatur: 0 ... +60°C
- Lagertemperatur: -25 ... +70°C
- Relative Feuchte: 5 ... 95% ohne Betauung
- Lüfterloser Betrieb

# Einbaumaße

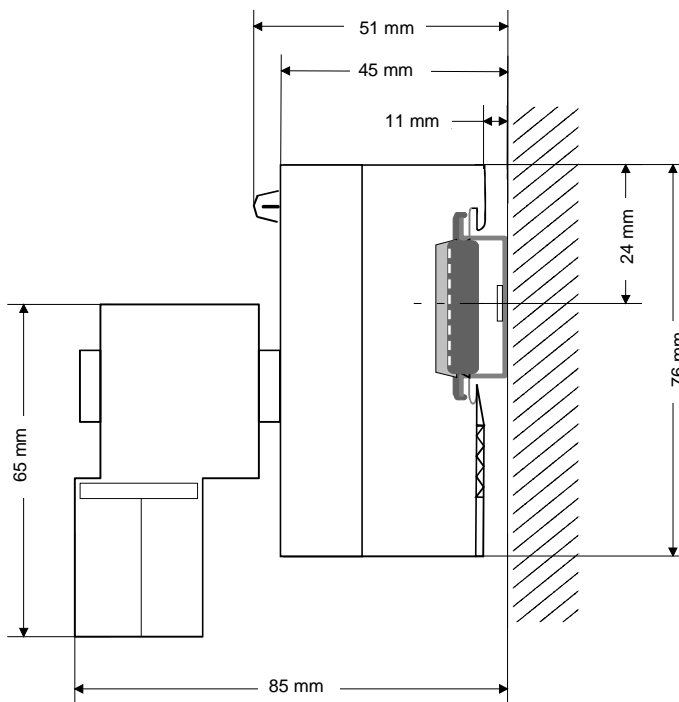
## Montagemaße



## Maße montiert und verdrahtet



**CPU 11x mit  
EasyConn von  
VIPA**



## Teil 2 Dezentrale Blockperipherie CAN-Bus CANopen

### Überblick

Die dezentrale Blockperipherie besteht aus einem CANopen-Slave und Ein-/Ausgabe-Komponenten. Sie ist in verschiedenen Ausführungen erhältlich, auf die im Folgenden näher eingegangen werden soll.

In diesem Kapitel sind alle Informationen zusammengefasst, die für Aufbau, Projektierung und Betrieb dieser System 100V Komponenten erforderlich sind.

Des Weiteren enthält das Kapitel neben einem Schnelleinstieg in die Projektierung für "Experten" eine Einführung in die Telegrammstruktur und die Funktionscodes von CANopen. Anschließend wird der Ablauf des Prozess- und Servicedatenaustausches beschrieben, gefolgt vom Objektverzeichnis, den Parameterdaten und Informationen zu Fehlerkontrolle und Wartung.

Nachfolgend sind beschrieben:

- Allgemeines zum CAN-Bus
- Aufbau und Montage
- Anschlussbilder
- Objektverzeichnis und -beschreibung
- Parameterdaten und Fehlerkontrolle

### Inhalt

Thema	Seite
<b>Teil 2 Dezentrale Blockperipherie CAN-Bus CANopen .....</b>	<b>2-1</b>
Grundlagen von CANopen .....	2-2
Systemübersicht.....	2-5
Aufbau.....	2-6
Montage und Verdrahtung.....	2-12
Anschlussbilder .....	2-13
Einsatz unter CANopen.....	2-14
Schnelleinstieg .....	2-15
Einstellung von Baudrate und Modul-ID .....	2-18
Telegrammaufbau .....	2-19
PDO - Prozessdaten-Objekt.....	2-21
SDO - Servicedaten-Objekt.....	2-24
Objekt-Verzeichnis .....	2-26
Emergency Object.....	2-40
NMT - Netzwerk Management.....	2-41
Technische Daten .....	2-43

## Grundlagen von CANopen

**UL-Zertifizierung** Die in diesem Kapitel aufgeführten Module haben die UL-Zulassung:



UL-Recognition-Mark  
Underwriters Laboratories (UL)  
Standard UL 508, File Nr.: E234291

### 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 sind beim CAN-Bus auch verschiedene Schicht-7-Anwenderprofile unter dem CAL-Schicht-7-Protokoll definiert (CAL=CAN application layer).

Ein solches Anwenderprofil ist CANopen, dessen Standardisierung der CiA CAN in Automation 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 der digitalen und analogen Ein-/Ausgabe-Module standardisiert.

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.

### PDO und SDO

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 25m und bei 50kbaud 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, mit einem  $120\Omega$  Abschlusswiderstand am Ende, konfiguriert sein.

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

Alle Teilnehmer im Netz kommunizieren mit der gleichen Baudrate.

Die Busstruktur 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 (zufälligem) 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.

**Parametrierung  
und Konfiguration**

Zur Parametrierung und Konfiguration des CANopen-Busses liefern die Hersteller des CANopen-Masters ein Software-Konfigurationstool. Mit diesem Tool können Sie auf das Objektverzeichnis Ihres CAN-Slaves zugreifen und somit Ihren Koppler parametrieren.

Die systemspezifischen Eigenschaften Ihres CANopen-Kopplers werden in Form einer EDS-Datei (electronic data sheet) mitgeliefert.

- Einschränkungen**
- max. 128 Teilnehmer pro Netz - max. 64 Slaves pro Segment
  - max. Netzausdehnung bei 1Mbaud 25m, bei 50kbaud 1000m (baudratenabhängig)
  - max. Baudrate 1Mbaud

**Buslänge**

Die maximale Buslänge wird bei CAN vorwiegend durch die Signallaufzeit beschränkt. Das Multi-Master Buszugriffsverfahren (Arbitrierung) erfordert, dass die Signale quasi gleichzeitig (vor der Abtastung innerhalb einer Bitzeit) an allen Knoten anliegen. Da die Signallaufzeit in den CAN-Anschaltungen (Transceiver, Optokoppler, CAN-Controller) nahezu konstant sind, muss die Leitungslänge an die Baudrate angepasst werden:

Baudrate	Buslänge
1MBit/s	25m
800MBit/s	50m
500kBit/s	100m
250kBit/s	250m
125kBit/s	500m
(50kBit/s)	1000m
(20kBit/s)	2500m

**Diagnose über Emergency-Object**

Zur Diagnose dient das "Emergency"-Object, das mit einer hohen Priorität versehen ist und Informationen über das Gerät und das Netz liefert.

**Adressierung**

Jeder Busteilnehmer identifiziert sich mit einer Adresse. Diese Adresse darf nur einmal in diesem Bussystem vergeben sein und kann zwischen 01... 99 liegen.

An den CAN-Bus-Kopplern stellen Sie die Adresse über den Adress-Einsteller ein.

**EDS-Datei**

Zur Konfiguration einer Slave-Anschaltung in Ihrem eigenen Projektierool bekommen sie die Leistungsmerkmale der Komponenten in Form einer EDS-Datei (**e**lectronic **d**ata **s**heet) mitgeliefert.

Die EDS-Dateien lauten:

**VIPA\_153\_4CF00.EDS**  
**VIPA\_153\_4CH00.EDS**  
**VIPA\_153\_6CH00.EDS**  
**VIPA\_153\_6CL10.EDS**


Installieren Sie diese EDS-Dateien in Ihrem Projektierool! Nähere Hinweise zur Installation der EDS-Dateien finden Sie im Handbuch zu Ihrem Projektierool.

## Systemübersicht

### Dezentrale Blockperipherie

Das System 100V ist ein universelles Bindeglied zwischen einem Feldbus und der Sensor/Aktor-Ebene.

Eine System 100V-Einheit besteht aus einem CANopen-Bus-Koppler und einer Kombination aus Ein-/Ausgabekanälen und Erweiterungsklemmen.

alte Best.-Nr.	neue Best.-Nr.	Modulbreite	Anzahl der Eingänge DC 24V	Anzahl der Ausgänge DC 24V, 1A	Anzahl Relais Ausgänge DC 30V/AC 230V, 5A	Eingangsdaten	Ausgangsdaten	Anzahl Klemmen	Stromaufnahme Baugruppe
									
	Digitale Ein-/Ausgabe								
123-4CF00	153-4CF00	4-fach	insg. 8		-	1Byte	1Byte	2x11	55mA
123-4CH00	153-4CH00	4-fach	8 ... 12	4 ... 8	-	2Byte	1Byte	-	55mA
123-6CH00	153-6CH00	6-fach	8 ... 12	4 ... 8	-	2Byte	1Byte	4x11	55mA
123-6CL10	153-6CL10	6-fach	24	8	-	3Byte	1Byte	-	55mA

### Einbindung in CANopen

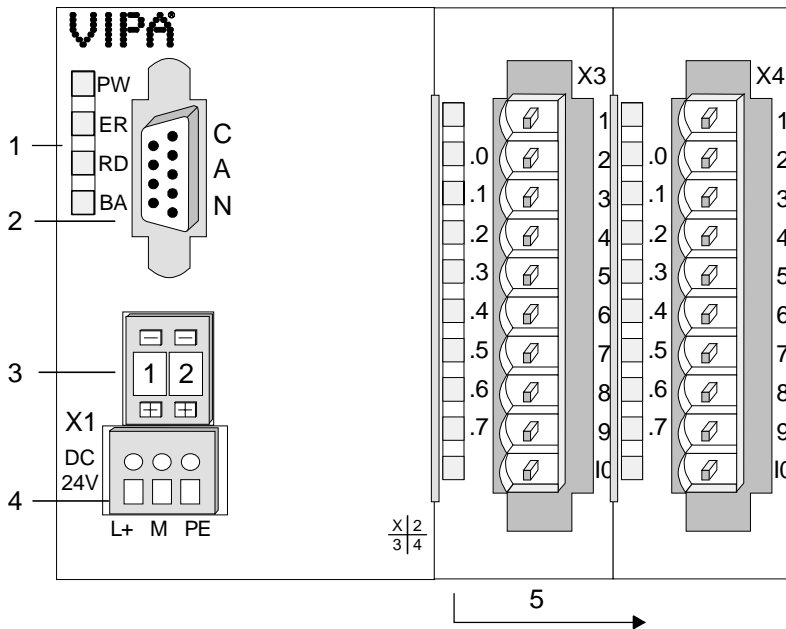
Über den integrierten CANopen-Koppler findet Ihr System 100V Anschluss an CAN.

### CANopen System-eigenschaften

- je 1 PDO für Senden und Empfangen (PDO Linking)
- 2 SDOs als Server
- Emergency Object
- NMT
- Node Guarding
- Heartbeat

# Aufbau

## Frontansicht



- [1] LED Statusanzeigen
- [2] CANopen-Schnittstelle
- [3] Adresseinsteller
- [4] Spannungsversorgung DC24V
- [5] Ein-/Ausgabe-Peripherie

## Komponenten

















### LEDs

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

Bezeichnung	Farbe	Bedeutung
PW	gelb	Signalisiert eine anliegende Betriebsspannung
ER	rot	Leuchtet bei Fehler
RD	grün	Blinkt mit 1Hz bei positivem Selbsttest und erfolgreicher Initialisierung
BA	gelb	Aus bei positivem Selbsttest und erfolgreicher Initialisierung 1Hz Blinken im Zustand "Pre-Operational". Leuchtet im Zustand "Operational". 10Hz Blinken im Zustand "Prepared".

**Statusanzeige durch LED-Kombination**

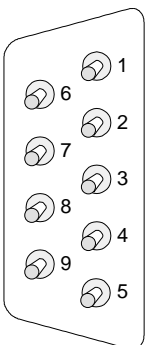
Durch Kombination der LEDs werden verschiedene Zustände angezeigt:

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

**9-poliger SubD-Stecker**

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

Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:



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

**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.

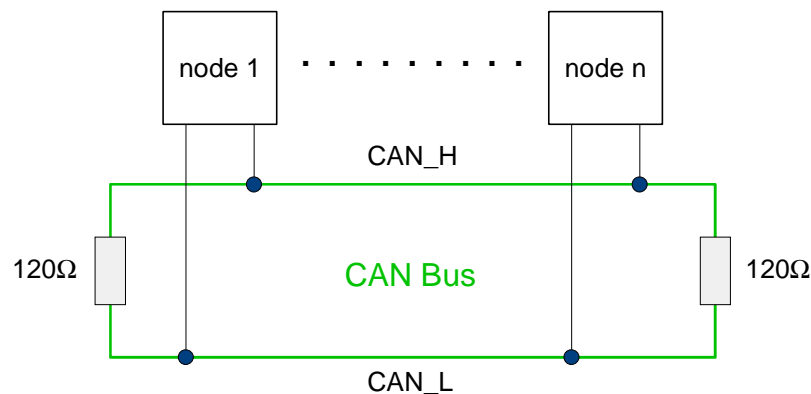
**Spannungsversorgung**

Der CAN-Bus-Koppler besitzt ein eingebautes Netzteil. Das Netzteil ist mit 24V Gleichspannung zu versorgen.

Das Netzteil ist gegen Verpolung und Überstrom geschützt.

**Verkabelung unter CAN-Bus**

CAN ist ein 2-Draht Bussystem, an dem alle Teilnehmer parallel angeschlossen werden. Der Bus muss an jedem Ende mit einem Abschlusswiderstand von 120 (bzw. 121) $\Omega$  terminiert werden, um Reflexionen zu vermeiden. Dies ist auch bei sehr kurzen Leitungslängen erforderlich.



Da die CAN-Signale als Differenzpegel auf dem Bus dargestellt werden, ist die CAN-Leitung vergleichsweise unempfindlich gegen eingepreßte Störungen (EMV). Da jeweils beide Leitungen betroffen sind, verändert die Störung den Differenzpegel kaum.

**Hinweis!**

An den Leitungsenden muss das Buskabel immer mit einem Abschlusswiderstand von 120 $\Omega$  terminiert werden, um Reflexionen und damit Übertragungsprobleme zu vermeiden!

**Kabel**

Für die CAN-Verdrahtung wird die Verwendung von paarig verdrehten, geschirmten Kabeln (2x2) mit einem Wellenwiderstand von 108...132 $\Omega$  empfohlen.

Wenn das Bezugspotential der CAN-Transceiver (CAN-Ground) nicht verbunden werden soll, kann auf das zweite Adernpaar verzichtet werden. Dies ist nur bei kleinen Netzausdehnungen mit gemeinsamer Speisung aller Teilnehmer empfehlenswert.

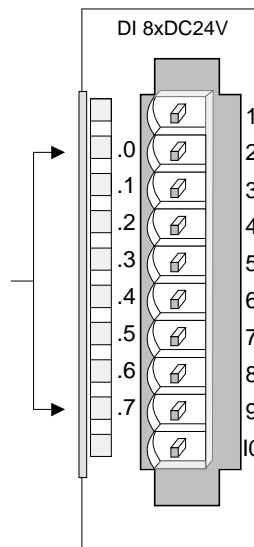
**Eingabe-Teil**

Der digitale Eingabe-Teil eines System 100V-Moduls erfasst die binären Steuersignale aus der Prozessebene und transportiert sie zum übergeordneten CANopen-Slave.

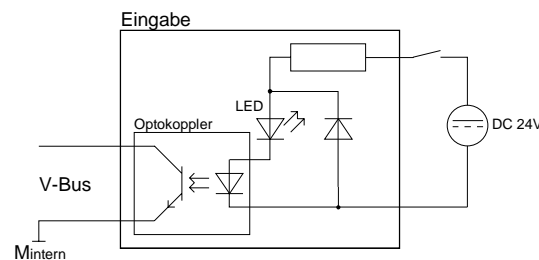
Jeder Eingabe-Kanal zeigt seinen Zustand über eine grüne LED, mit einer Verzögerungszeit von max. 3ms, an. Die Nenn Eingangsspannung beträgt DC 24V. Hierbei bewirken 0...5V den Signalzustand "0" und 15...28,8V den Signalzustand "1". Näheres zum Anschluss des Eingabeteils finden Sie unter "Anschlussbilder".

**Statusanzeige  
Steckerbelegung****LED Beschreibung**

.0... .7 LEDs (grün)  
E+0.0 bis E+0.7  
ab ca. 15V wird das  
Signal "1" erkannt und die  
entsprechende LED  
angesteuert

**Pin Belegung**

Pin	Belegung
1	nicht belegt
2	Eingang E+0.0
3	Eingang E+0.1
4	Eingang E+0.2
5	Eingang E+0.3
6	Eingang E+0.4
7	Eingang E+0.5
8	Eingang E+0.6
9	Eingang E+0.7
10	Masse

**Prinzipschaltbild  
Eingabe-Teil**

**Ausgabe-Teil**

Der digitale Ausgabe-Teil erfasst die binären Steuersignale aus dem übergeordneten CANopen-Slave und transportiert sie über die Ausgänge an die Prozessebene. Der Ausgabe-Teil ist zusätzlich mit DC 24V über den Frontstecker zu versorgen (siehe Anschlussbilder). Die anliegende Versorgungsspannung wird über die gelbe LED (L+) angezeigt.

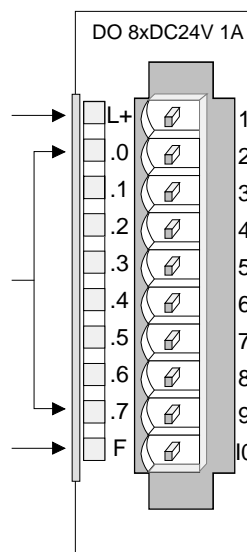
Jeder digitale Ausgabe-Kanal zeigt seinen Zustand über eine grüne LED an. Bei aktiviertem Ausgang leuchtet die zugehörige LED.

Bei Überlast, Überhitzung oder Kurzschluss leuchtet die mit "F" bezeichnete Fehler-LED rot.

**Statusanzeige  
Steckerbelegung**

**LED Beschreibung**

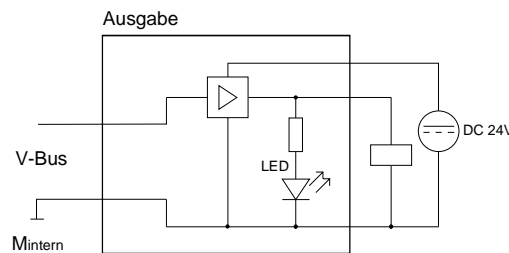
- L+ LED (gelb)  
Versorgungsspannung liegt an
- .0... .7 LEDs (grün)  
A+0.0 bis A+0.7  
sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert
- F LED (rot)  
Fehler bei Überlast, Überhitzung oder Kurzschluss



**Pin Belegung**

- 1 Versorgungsspg. DC 24V
- 2 Ausgang A+0.0
- 3 Ausgang A+0.1
- 4 Ausgang A+0.2
- 5 Ausgang A+0.3
- 6 Ausgang A+0.4
- 7 Ausgang A+0.5
- 8 Ausgang A+0.6
- 9 Ausgang A+0.7
- 10 Versorgungsspg. Masse

**Prinzipschaltbild  
Ausgabe-Teil**



## Ein-/Ausgabe-Teil

Der Ein-/Ausgabe-Teil besteht aus 4 E/A-Kanälen, die als Ein- bzw. Ausgabe-Kanal eingesetzt werden können und 4 Ausgabe-Kanäle. Jeder der 4 E/A-Kanäle besitzt eine Diagnosefunktion, d.h. sobald ein Ausgang aktiv ist, wird der zugehörige Eingang auf "1" gesetzt. Bei einem Kurzschluss an der Last, wird der Eingang auf "0" gezogen und durch Auswertung des Eingangs kann der Fehler erkannt werden.

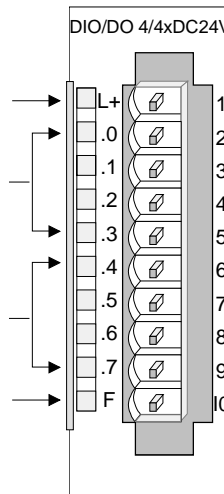
Der Ein-/Ausgabe-Teil ist mit DC 24V über den Frontstecker zu versorgen (siehe Anschlussbilder). Die anliegende Versorgungsspannung wird über die gelbe LED (L+) angezeigt.

Jeder digitale Eingabe-/Ausgabe Kanal zeigt seinen Zustand über eine grüne LED an. Bei aktiviertem Ein-/Ausgang leuchtet die zugehörige LED.

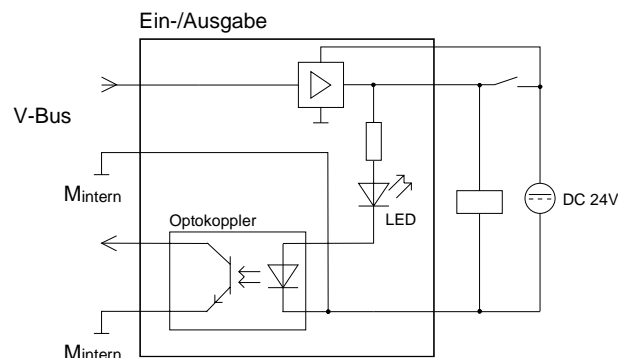
Bei Überlast, Überhitzung oder Kurzschluss leuchtet die mit "F" bezeichnete Fehler-LED rot.

## Statusanzeige Steckerbelegung

LED	Beschreibung	Pin	Belegung
L+	LED (gelb) Versorgungsspannung liegt an	1	Versorgungsspg. DC 24V
.0... .3	LEDs (grün) E/A+0.0 bis E/A+0.3 sobald E/A=1 wird die entsprechende LED angesteuert	2 3 4 5	Ein-/Ausgang E/A+0.0 Ein-/Ausgang E/A+0.1 Ein-/Ausgang E/A+0.2 Ein-/Ausgang E/A+0.3
.4... .7	LEDs (grün) A+0.4 bis A+0.7 sobald ein Ausgang aktiv ist, wird die entsprechende LED angesteuert	6 7 8 9	Ausgang A+0.4 Ausgang A+0.5 Ausgang A+0.6 Ausgang A+0.7
F	LED (rot) Fehler bei Überlast, Überhitzung oder Kurzschluss	10	Versorgungsspg. Masse



## Prinzipschaltbild Ein-/Ausgabe-Teil

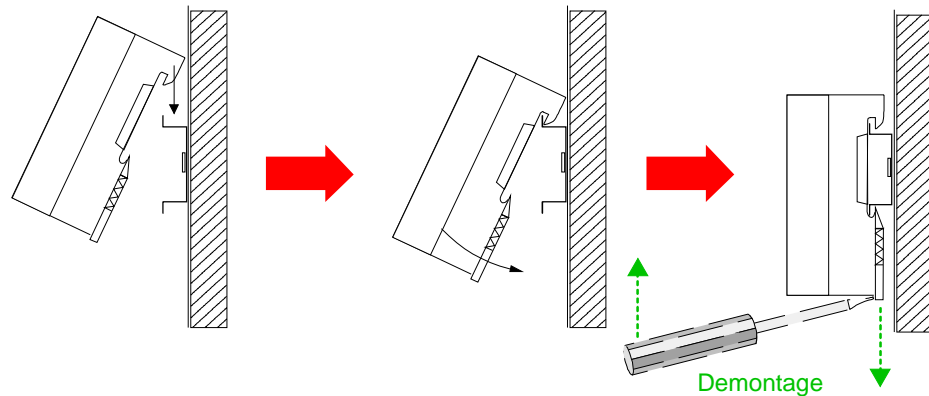


## Montage und Verdrahtung

### Montage/ Demontage

System 100V-Module werden auf 35mm Standard-Normprofilschienen montiert.

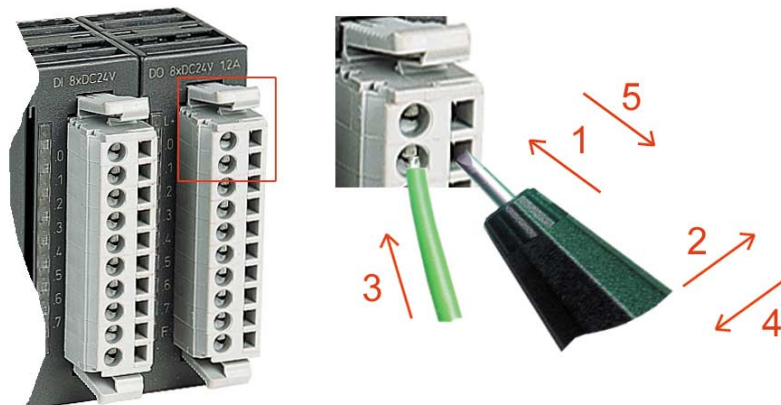
Zur Demontage ist die Verriegelung mit einem Schraubendreher nach unten zu ziehen und das Modul von der Hutschiene abzuheben.



### Verdrahtung

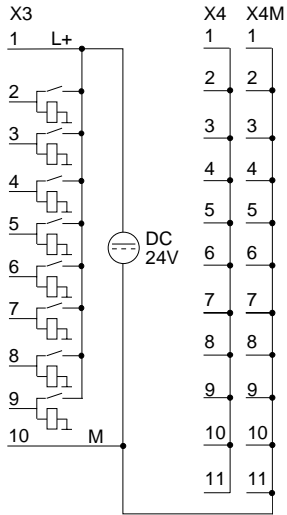
Drücken Sie mit einem passenden Schraubendreher die Federklemme in der rechteckigen Öffnung nach hinten und führen Sie durch die runde Öffnung den Draht ein.

Durch Herausziehen des Schraubendrehers wird der Draht sicher gehalten.

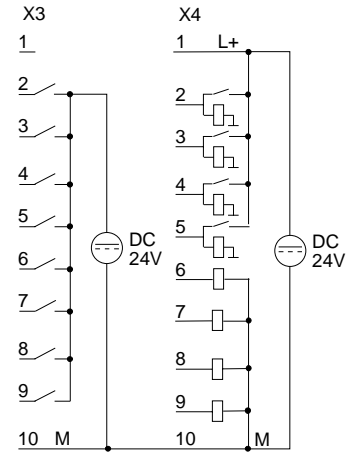


# Anschlussbilder

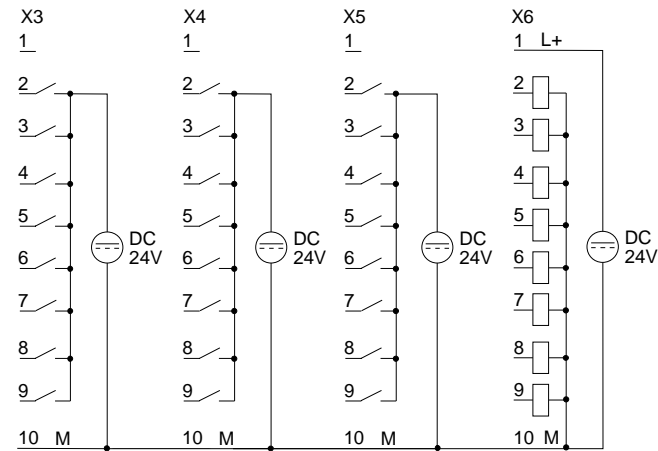
VIPA 153-4CF00



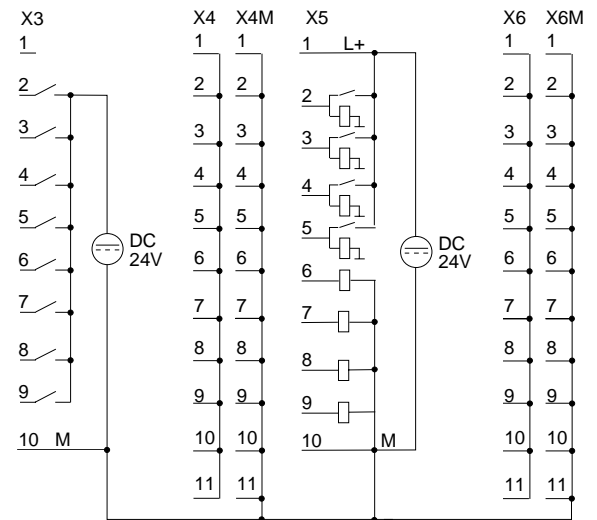
VIPA 153-4CH00



VIPA 153-6CL10



VIPA 153-6CH00



## Einsatz unter CANopen

### Übersicht

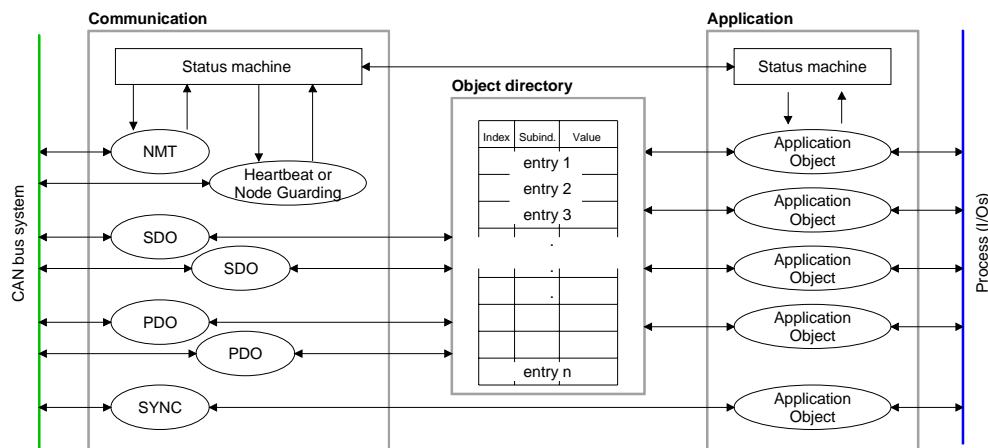
Der Datenaustausch bei CANopen-Geräten erfolgt über Datenobjekte. Im CANopen-Kommunikationsprofil sind zwei Standard-Objektarten (PDO und SDO) sowie einige Spezialobjekte (für Netzwerkmanagement usw.) definiert.

Das System 100V für CAN unterstützt folgende Objekte:

- 1 Sende-PDO, 1 Empfangs-PDO, PDO Linking
- 2 Standard-SDO (Server)
- 1 Emergency Object
- 1 Synchronisations-Objekt (SYNC ohne Timestamp)
- Node Guarding / Heartbeat
- NMT

### Struktur des Gerätemodells

Ein CANopen Gerät kann wie folgt strukturiert werden:



#### Kommunikation

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

#### Applikation

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.

#### Objektverzeichnis

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

## 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 100V 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 100V werden aus den Knotenadressen (1...63) abgeleitet:

Datenart	Default CAN-Identifizier
digitale Eingänge 1 ... 24Bit	0x180 (=384) + Knotenadresse
digitale Ausgänge 1 ... 8Bit	0x200 (=512) + Knotenadresse

### Digitale Ein-/Ausgänge

Die CAN-Nachrichten mit digitalen Eingangsdaten stellen sich wie folgt dar:

Identifizier 0x180+Knotenadresse + bis zu 3Byte Nutzdaten (je nach Modul)

Identifizier 11Bit	DI 0 8Bit	DI 1 8Bit	DI 2 8Bit
--------------------	-----------	-----------	-----------

Die CAN-Nachrichten mit digitalen Ausgangsdaten stellen sich wie folgt dar:

Identifizier 0x200+Knotenadresse + bis zu 1Byte Nutzdaten

Identifizier 11Bit	DO 0 8Bit
--------------------	-----------

**Node Guarding**

Da das System 100V 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 11-Bit Identifier:

Identifier 0x700+Knotenadresse

<b>Identifier</b> 11Bit
-------------------------

Der System 100V Knoten antwortet mit einem Telegramm, das ein Statusbyte enthält:

Identifier 0x700+Knotenadresse + Statusbyte

<b>Identifier</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 100V 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-Identifier (0x700+Modul-Id) übertragen:

Identifier 0x700+Knotenadresse + Statusbyte

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

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

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

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

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

Identifizier 0x80 + Knotenadresse + 8NutzdatenByte

Identifizier 11 Bit	EC0	EC1	EReg	Inf0	Inf1	Inf2	Inf3	Inf4
---------------------	-----	-----	------	------	------	------	------	------

Error Code	Bedeutung	Info 0	Info 1	Info 2	Info 3	Info 4
0x0000	Reset Emergency					
0x1000	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



### Hinweis!

Mit den beschriebenen Telegrammen sind Sie nun in der Lage, Ihr System 100V zu starten und zu stoppen, Eingänge zu lesen, Ausgänge zu schreiben und die Baugruppen zu überwachen.

Nachfolgend sind alle Funktionen nochmals detailliert beschrieben.

## Einstellung von Baudrate und Modul-ID

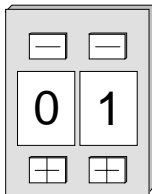
### Übersicht

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

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

### Einstellung der Baudrate über Adresseinsteller

- Stellen Sie am Adresseinsteller die Adresse 00 ein
- Schalten Sie die Spannungsversorgung für den CAN-Buskoppler 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"	1 MBaud	25 m
"01"	500 kBaud	100 m
"02"	250 kBaud	250 m
"03"	125 kBaud	500 m
"04"	100 kBaud	600 m
"05"	50 kBaud	1000 m
"06"	20 kBaud	2500 m
"07"	10 kBaud	5000 m
"08"	800 kBaud	50 m

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

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

### Einstellung der Modul-ID

- Stellen Sie die Modul-ID im Bereich 01...99 am Adresseinsteller ein. Jede Modul-ID darf nur einmal am Bus vergeben sein. Die Modul-ID muss vor dem Einschalten des Buskopplers eingestellt werden.  
Nach 5s werden die Einstellungen übernommen und der Buskoppler geht in den Normalbetrieb (Zustand "Pre-Operational").

### Einstellung der Baudrate über SDO-Write

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

## Telegrammaufbau

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

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

Der Unterschied zu einem Schicht-2-Telegramm besteht in einer zusätzlichen Unterteilung des 2Byte 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-Buskoppler unterstützt folgende Objekte:

- 1 Transmit-PDO
- 1 Receive-PDO
- 2 SDOs
- 1 Emergency Object
- 1 Netzwerkmanagement Objekt NMT
- Node Guarding
- Heartbeat

Jedem Objekt ist ein Funktionscode zugeordnet. Bitte entnehmen Sie den entsprechenden Funktionscode der nachfolgenden Tabelle!

**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	Netzwerkmanagem.
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
SDO2S2M	*	Master	Applikationsspez.	Konfigurationsdaten
SDO2M2S	*	Slave	Applikationsspez.	Konfigurationsdaten
Node Guarding	1110	Master, Slave (RTR)	CiA DS-301	Modulüberwachung
Heartbeat	1110	Master, Slave	Applikationsspez.	Modulüberwachung

\*) Den Funktionscode für SDO2 entnehmen Sie bitte dem Objektverzeichnis 0x1201.



### 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.

## PDO - Prozessdaten-Objekt

### PDO

Für den Prozessdatenaustausch stehen 2 **Prozessdaten-Objekte** (PDO) zur Verfügung. Jedes PDO besteht dabei aus maximal 8 Datenbytes. PDOs werden unbestätigt übertragen, da das CAN-Protokoll die Übertragung sicherstellt. Für Eingangsdaten steht 1 Transmit-PDO und für Ausgangsdaten 1 Receive-PDO zur Verfügung.

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 8 Byte aufgeteilt. Diese Segmente heißen Prozessdatenobjekte (PDOs). Die PDOs entsprechen jeweils einem CAN-Telegramm und werden über dessen spezifischen CAN-Identifizier zugeordnet und in ihrer Priorität bestimmt.

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. Für den Prozessdatenaustausch steht jeweils 1 RxPDO und 1 TxPDO zur Verfügung. Die Belegung dieser PDOs mit Ein- bzw. Ausgangsdaten erfolgt automatisch.

### PDO Mapping

Das PDO-Mapping ist fix und kann nicht geändert werden.

### PDO Identifizier COB-ID

Der wichtigste Kommunikationsparameter eines PDOs ist der CAN-Identifizier (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 Busmaster.

Im System 100V sind für Sende- und Empfangs-PDOs Default-Identifizier in Abhängigkeit von der Knotenadresse vorgesehen.

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

Der Transmissionstyp ist im Objektverzeichnis (Indizes 0x1400-0x1404 und 0x1800-0x1804, Subindex 0x02) fest auf asynchron, eventgesteuert (= 0xFF) eingestellt. Über den EVENT-Timer (Value \* 1ms) können die PDOs zyklisch übertragen werden.

Send:                   0x180 + Modul-ID: PDO1S2M Digital                   (nach DS-301)

Receive:               0x200 + Modul-ID: PDO1M2S Digital                   (nach DS-301)

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

**PDO  
Übertragungsart**

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

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

**Synchron**

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

Bei Übertragungsart 1-240 wird das PDO zyklisch gesendet bzw. erwartet: nach jedem "n-ten" SYNC ( $n=1\dots240$ ). Da die Übertragungsart nicht nur im Netz, sondern auch auf einem Koppler kombiniert werden darf, kann so z.B. ein schneller Zyklus für digitale Eingänge vereinbart werden ( $n=1$ ), während die Daten der Analogeingänge 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.

## SDO - Servicedaten-Objekt

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 8Bytes mit Protokollinformationen belegt. Für Zugriffe auf Objektverzeichniseinträge mit bis zu vier Bytes Länge genügt eine einzige CAN-Nachricht. Bei Datenlängen größer als 4Bytes 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-Identifier sind:

- Receive-SDO1: 0x600 + Modul-ID
- Transmit-SDO1: 0x580 + Modul-ID
- Receive-SDO2: 0x640 + Modul-ID
- Transmit-SDO2: 0x5C0 + 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**

0x05030000	Toggle Bit not alternated
0x05040000	SDO protocol timeout
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 a 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)

## Objekt-Verzeichnis

### Struktur

Im CANopen-Objektverzeichnis werden alle für den Buskoppler relevanten CANopen Objekte eingetragen.

Jeder Eintrag im Objektverzeichnis ist durch einen 16-Bit-Index gekennzeichnet.

Falls ein Objekt aus mehreren Komponenten besteht (z.B. Objekttyp Array oder Record), sind die Komponenten über einen 8-Bit-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, oder gelesen und geschrieben werden darf.

Das Objektverzeichnis ist in folgende 3 Bereiche aufgeteilt:

### Kommunikations-spezifischer Profilbereich (0x1000 – 0x1FFF)

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

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

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

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

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

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

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

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

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



### Hinweis!

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

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

**Objektverzeichnis  
Übersicht**

Index		Content of Object
0x1000		Device type
0x1001		Error register
0x1003		Error store
0x1004		Number of PDOs
0x1005		SYNC identifier
0x1006		SYNC interval
0x1008		Device name
0x1009		Hardware version
0x100A		Software version
0x100B		Node number
0x100C		Guard time
0x100D		Life time factor
0x100E		Node Guarding Identifier
0x1010	X	Save parameter
0x1011	X	Load parameter
0x1014		Emergency COB-ID
0x1016	X	Heartbeat consumer time
0x1017	X	Heartbeat producer time
0x1018		Device identification
0x1029		Error behavior
0x1400	X	Communication parameter for Receive-PDOs (RxPDO, Master to Slave)
0x1600		Mapping parameter for Receive-PDOs (RxPDO)
0x1800	X	Communication parameter for Transmit-PDOs (TxPDO, Slave to Master)
0x1A00		Mapping parameter for Transmit-PDOs (TxPDO)
0x2001		CAN-Baudrate
0x2100		Kill EEPROM
0x2400	X	PDO Control
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	X	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)

X = save into EEPROM

**Device Type**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1000	0	Device Type	Unsigned32	ro	N	0x00030191	Statement of device type

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

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

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

- z=1 bedeutet digitale Eingänge,
- y=1 digitale Ausgänge,
- x=1 analoge Eingänge,
- w=1 analoge Ausgänge.

VIPA 153-xxxxx = 0x0003 0191

**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
<b>0x1003</b>	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
	1	Actual error	Unsigned32	ro	N		Last error state to have occurred
	...	...	...	...	...	...	...
	10		Unsigned32	ro	N		A maximum of 10 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 Object) und damit eine detaillierte Fehlerbeschreibung.

Neue Fehler werden jeweils unter 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 Power-On 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
<b>0x1004</b>	0	Number of PDOs supported	Unsigned32	ro	N	0x00010001	Number of PDOs supported
	1	Number of synchronous PDOs supported	Unsigned32	ro	N	0x00010001	Number of synchronous PDOs supported
	2	Number of asynchronous PDOs supported	Unsigned32	ro	N	0x00010001	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 (Rx)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 32-Bit 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 153-xxxxx = VIPA CANopen-Slave 153-4CF00

VIPA CANopen-Slave 153-4CH00

VIPA CANopen-Slave 153-6CH00

VIPA CANopen-Slave 153-6CL10

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

**Hardware version**

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

VIPA 153-xxxxx = 1.00

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

**Software version**

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

VIPA 153-xxxxx = 1.00

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

**Node number**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
<b>0x100B</b>	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
<b>0x100C</b>	0	Guard time [ms]	Unsigned16	rw	N	0x0000	Interval between two guard telegrams 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 06040043h.

### 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	0xAFFEAF00	Vendor ID
	2	Product Code	Unsigned32	ro	N	VIPA 153-4CF00 = 0x1534CF00 VIPA 153-4CH00 = 0x1534CA00 VIPA 153-6CL10 = 0x1536CB10 VIPA 153-6CH00 = 0x1536CA00	Product Code
	3	Revision Number	Unsigned32	ro	N		Revision Number
	4	Serial Number	Unsigned32	ro	N		Serial Number

### 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

### 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	0x40000200 + 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 32-Bit Wertes (Bits 0-10) enthalten den CAN-Identifizier, das MSBit (Bit 31) gibt Auskunft, ob das PDO aktiv ist (1) oder nicht (0), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1).

Der Subindex 2 enthält die Übertragungsart.

### Mapping RxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1600	0	Number of Elements	Unsigned8	ro	N	0x01	Mapping parameter of the first receive PDO; sub-index 0: number of mapped objects
	1	1st mapped object	Unsigned32	ro	N	0x62000108	(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, muss der Eintrag in Subindex 0 entsprechend angepasst werden.

### 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	0x00000180 + 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 100µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1ms]

Subindex 1 (COB-ID): Die unteren 11Bit des 32-Bit Wertes (Bits 0-10) enthalten den CAN-Identifizier, das MSBit (Bit 31) gibt Auskunft, ob das PDO aktiv ist (1) oder nicht (0), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1). Der Subindex 2 enthält die Übertragungsart, Subindex 3 die Wiederholungsverzögerung zwischen zwei gleichen PDOs. Wenn ein Event Timer mit einem Wert ungleich 0 existiert, wird nach Ablauf dieses Timers das PDO übertragen.

Existiert ein Inhibit Timer, wird das Ereignis um diese Zeit verzögert.

### Mapping TxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1A00	0	Number of Elements	Unsigned8	ro	N	VIPA 153-4CF00=0x01 VIPA 153-4CH00=0x02 VIPA 153-6CL10=0x03 VIPA 153-6CH00=0x02	Mapping parameter of the first transmit PDO; sub-index 0: number of mapped objects
	1	1st mapped object	Unsigned32	ro	N	0x60000108	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	2	2nd mapped object	Unsigned32	ro	N	0x60000208	(2 byte index, 1 byte sub-index, 1 byte bit-width)
	3	3rd mapped object	Unsigned32	ro	N	0x60000308	(2 byte index, 1 byte sub-index, 1 byte bit-width)

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.

## 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

## 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]

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.

**8bit Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
<b>0x6000</b>	0x00	8-bit digital input block	Unsigned8	ro	N	VIPA 153-4CF00=0x01 VIPA 153-4CH00=0x02 VIPA 153-6CL10=0x03 VIPA 153-6CH00=0x02	Number of available digital 8-bit input blocks
	0x01	1st input block	Unsigned8	ro	Y		1st digital input block
	0x02	2nd input block	Unsigned8	ro	Y		2nd digital input block
	0x03	3rd input block	Unsigned8	ro	Y		3rd digital input block

**8bit Polarity Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
<b>0x6002</b>	0x00	8-bit digital input block	Unsigned 8	ro	N	VIPA 153-4CF00=0x01 VIPA 153-4CH00=0x02 VIPA 153-6CL10=0x03 VIPA 153-6CH00=0x02	Number of available digital 8-bit input blocks
	0x01	1st input block	Unsigned 8	rw	N	0x00	1st polarity digital input block
	0x02	2nd input block	Unsigned 8	rw	N	0x00	2nd polarity digital input block
	0x03	3rd input block	Unsigned 8	rw	N	0x00	3rd polarity digital input block

Individuelle Invertierung der Eingangskanäle

1 = Eingang invertiert

0 = Eingang nicht invertiert

**8bit Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
<b>0x6200</b>	0x00	8-bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8-bit output blocks
	0x01	1st output block	Unsigned8	rw	Y		1st digital output block

### 8bit Change Polarity Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6202	0x00	8-bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8-bit output blocks
	0x01	1st output block	Unsigned8	rw	N	0x00	1st polarity digital output block

Individuelle Invertierung der Ausgangskanäle

1 = Ausgang invertiert

0 = Ausgang nicht invertiert

### 8bit Error Mode Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6206	0x00	8-bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8-bit output blocks
	0x01	1st output block	Unsigned8	rw	N	0xFF	1st error mode digital output block

Mit diesem Objekt können Sie bestimmen, ob im Fehlerfall ein Ausgabe-Kanal einen bestimmten Wert annimmt, den Sie im Objekt 0x6207 vorgeben.

1 = den Wert in Objekt 0x6207 übernehmen

0 = Ausgabewert im Fehlerfall fixieren

### 8bit Error Value Digital outputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6207	0x00	8-bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8-bit output blocks
	0x01	1st output block	Unsigned8	rw	N	0x00	1st error value digital output block

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

1 = Im Fehlerfall Ausgabewert auf 0 sobald Objekt 0x6206 aktiviert ist.

0 = Im Fehlerfall Ausgabewert auf 1 sobald Objekt 0x6206 aktiviert ist.

# 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



---

**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

### Dezentrale Block- peripherie CAN-Bus

Elektrische Daten				
Spannungsversorgung	DC 24V, über Front von ext. Netzteil			
Stromaufnahme	max. 55mA			
Statusanzeige	über LEDs auf der Frontseite			
Anschlüsse/Schnittstellen	9-polig SubD (Stecker) CAN-Bus-Ankopplung			
CAN-Bus Schnittstelle				
Ankopplung	9-polige 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			
max. Teilnehmeranzahl	127 Stationen (je nach Masteranschlaltung)			
digitale Ein-/Ausgabe	153-4CF00	153-4CH00	153-6CL10	153-6CH00
Anzahl der Eingänge	0...8	8...12	24	8...12
Anzahl der Ausgänge	0...8	4...8	8	4...8
Eingangsdaten	1Byte	2Byte	3Byte	2Byte
Ausgangsdaten	1Byte	1Byte	1Byte	1Byte
Eingangsspannung bei "1"	DC15...28,8V	DC15...28,8V	DC15...28,8V	DC15...28,8V
Eingangsspannung bei "0"	DC 0...5V	DC 0...5V	DC 0...5V	DC 0...5V
Verzögerungszeit	3ms	3ms	3ms	3ms
Ausgangsstrom pro Kanal	1A	1A	1A	1A
Modulbreite	4	4	6	6
Anzahl der Klemmen	2x11	-	-	4x11
Schnittstelle	RS 485; 9pol. SubD Stecker			
Übertragungsrate	10kBaud bis 1MBaud			



## Anhang

### A Index

<b>A</b>		Inhibit-Zeit.....	2-23
Aderquerschnitt.....	1-4	<b>K</b>	
Adresseinsteller .....	2-18	Komponenten .....	2-6
Anschlussbilder.....	2-13	<b>L</b>	
Aufbau .....	2-6	LEDs.....	2-6
Ausgabe-Teil.....	2-10	<b>M</b>	
<b>B</b>		Maße .....	1-5
Baudrate .....	2-18	Modul-ID .....	2-18
Buslänge.....	2-4	Montage.....	2-12
Buszugriff.....	2-3	Montagemaße .....	1-5
<b>C</b>		<b>N</b>	
CANopen .....	2-2	NMT.....	2-41
<b>D</b>		Node Guarding .....	2-42
Datenobjekte.....	2-14	<b>O</b>	
Digitale Ausgabe.....	2-10	Objektverzeichnis .....	2-26
Digitale Eingabe.....	2-9	<b>P</b>	
<b>E</b>		Parametrierung.....	2-3
EDS-Datei.....	2-4	PDO.....	2-2, 2-21
Ein-/Ausgabe-Teil .....	2-11	<b>S</b>	
Einbaumaße .....	1-5	Schnelleinstieg .....	2-15
Eingabe-Teil.....	2-9	SDO.....	2-2, 2-24
Einsatz .....	2-14	Sicherheitshinweise.....	1-2
Einschränkungen .....	2-4	Spannungsversorgung .....	2-8
Emergency Object .....	2-17, 2-40	Statusanzeige .....	2-7
<b>F</b>		SubD-Stecker .....	2-7
Fehlercodes .....	2-17	Systemübersicht.....	1-3, 2-5
Funktionscodes.....	2-20	Allgemeine Beschreibung.....	1-4
<b>G</b>		<b>T</b>	
Gerätemodell .....	2-14	Technische Daten .....	2-43
Grundlagen .....	1-1	Telegrammaufbau .....	2-19
CAN-Bus.....	2-2	<b>U</b>	
System 100V .....	1-3	Umgebungsbedingungen .....	1-4
<b>H</b>		<b>V</b>	
Hardwarebeschreibung.....	2-6	Verdrahtung.....	2-12
Heartbeat .....	2-16, 2-42	Verkabelung .....	2-8
<b>I</b>			
Identifizier .....	2-15		

