



VIPA System 300V



IM | Handbuch

HB130D_IM | Rev. 09/46

November 2009

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

<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

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 1150/1180 (Hotline)

E-Mail: support@vipa.de

Inhaltsverzeichnis

| | |
|--|------------|
| Über dieses Handbuch | 1 |
| Sicherheitshinweise | 2 |
| Teil 1 Grundlagen | 1-1 |
| Sicherheitshinweis für den Benutzer | 1-2 |
| Allgemeine Beschreibung System 300V..... | 1-3 |
| Komponenten..... | 1-4 |
| Teil 2 Montage und Aufbaurichtlinien | 2-1 |
| Übersicht..... | 2-2 |
| Einbaumaße..... | 2-3 |
| Montage auf Profilschiene..... | 2-4 |
| Verdrahtung | 2-6 |
| Aufbaurichtlinien..... | 2-10 |
| Teil 3 Profibus-DP | 3-1 |
| Systemübersicht..... | 3-2 |
| Grundlagen | 3-3 |
| IM 353-1DP00 - DP-V0-Slave - Aufbau | 3-12 |
| IM 353-1DP00 - DP-V0-Slave - Projektierung | 3-15 |
| IM 353-1DP00 - DP-V0-Slave - Diagnosefunktionen | 3-16 |
| IM 353-1DP01 - DP-V1-Slave - Aufbau | 3-22 |
| IM 353-1DP01 - DP-V1-Slave - Projektierung | 3-25 |
| IM 353-1DP01 - DP-V1-Slave - DP-V1-Dienste | 3-28 |
| IM 353-1DP01 - DP-V1-Slave - Diagnosefunktionen | 3-30 |
| Aufbaurichtlinien..... | 3-38 |
| Inbetriebnahme | 3-43 |
| Einsatz der Diagnose-LEDs | 3-45 |
| Technische Daten | 3-46 |
| Teil 4 CANopen | 4-1 |
| Systemübersicht..... | 4-2 |
| Grundlagen | 4-3 |
| IM 353CAN - CANopen-Slave - Aufbau..... | 4-5 |
| IM 353CAN - CANopen-Slave - Schnelleinstieg | 4-9 |
| IM 353CAN - CANopen-Slave - Baudrate und Modul-ID | 4-13 |
| IM 353CAN - CANopen-Slave - Telegrammaufbau | 4-14 |
| IM 353CAN - CANopen-Slave - PDO | 4-16 |
| IM 353CAN - CANopen-Slave - SDO | 4-20 |
| IM 353CAN - CANopen-Slave - Objekt-Verzeichnis | 4-22 |
| IM 353CAN - CANopen-Slave - Emergency Object..... | 4-63 |
| IM 353CAN - CANopen-Slave - NMT - Netzwerk Management..... | 4-64 |
| Technische Daten | 4-66 |
| Anhang | A-1 |
| Index | A-1 |

Über dieses Handbuch

Das Handbuch beschreibt die bei VIPA erhältlichen System 300V Interface-Module (IM). Hier finden Sie neben einer Produktübersicht detaillierte Beschreibungen der einzelnen Module. Sie erhalten Informationen für den Anschluss und die Handhabung der System 300V IM-Module.

Überblick

Teil 1: Grundlagen

Im Rahmen dieser Einleitung erfolgt die Vorstellung des System 300V von VIPA als zentrales bzw. dezentrales Automatisierungssystem.

Teil 2: Montage und Aufbaurichtlinien

Alle Informationen, die für den Aufbau und die Verdrahtung einer Steuerung aus den Komponenten des System 300V erforderlich sind, finden Sie in diesem Kapitel.

Teil 3: Profibus-DP

Inhalt dieses Kapitels ist der Einsatz des System 300V unter Profibus. Hier wird die Projektierung und Parametrierung der Profibus-Slave-Module von VIPA beschrieben.

Teil 4: CANopen

Dieser Teil befasst sich mit dem CANopen-Slave IM 353CAN von VIPA und dessen Einsatz im CAN-Bus.

| | |
|---------------------------------|---|
| Zielsetzung und Inhalt | Dieses Handbuch beschreibt die Interface-Module (IM), die im System 300 eingesetzt werden können. Beschrieben wird 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: <ul style="list-style-type: none">• 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: <ul style="list-style-type: none">• 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

Die Interface-Module sind konstruiert und gefertigt für:

- System 300 Komponenten von VIPA und Siemens
- 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:

- Änderungen 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 300V von VIPA. In einer Übersicht werden die Möglichkeiten zum Aufbau von zentralen und dezentralen Systemen aufgezeigt.

Sie finden hier auch allgemeine Angaben zum System 300V wie Maße, Hinweise zur Montage und zu den Umgebungsbedingungen.

Inhalt

| Thema | Seite |
|---|------------|
| Teil 1 Grundlagen | 1-1 |
| Sicherheitshinweis für den Benutzer | 1-2 |
| Allgemeine Beschreibung System 300V | 1-3 |
| Komponenten | 1-4 |

Sicherheitshinweis für den Benutzer

Handhabung elektrostatisch gefährdeter Baugruppen

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

Zur Kennzeichnung dieser gefährdeten Baugruppen wird nachfolgendes Symbol verwendet:



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

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

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

Versenden von Baugruppen

Verwenden Sie für den Versand immer die Originalverpackung.

Messen und Ändern von elektrostatisch gefährdeten Bau- gruppen

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

- 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.

Allgemeine Beschreibung System 300V

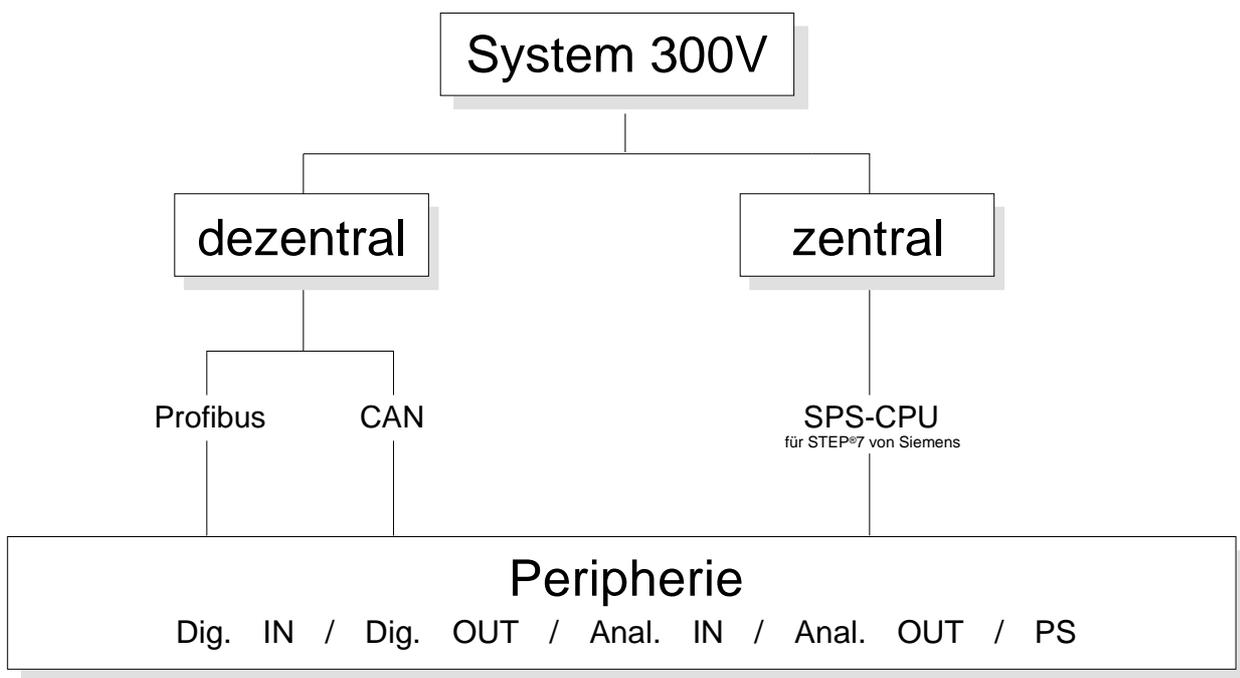
Das System 300V

Das System 300V ist ein modulares zentral wie dezentral einsetzbares Automatisierungssystem für Anwendungen im mittleren und oberen Leistungsbereich. Die einzelnen Module werden direkt auf der 530mm langen Profilschiene montiert und über Busverbinder, die von hinten an die Module gesteckt werden, gekoppelt.

Die einzelnen Module des VIPA System 300V sind baugleich zu Siemens. Durch den kompatiblen Rückwandbus sind somit Module von VIPA und Siemens mischbar.

Die CPUs des System 300V sind befehlskompatibel zur S7-300 von Siemens. Programmiert werden die CPUs mit der VIPA Programmiersoftware WinPLC7 oder dem SIMATIC Manager von Siemens sowie weiteren am Markt erhältlichen Programmierertools.

Die nachfolgende Abbildung soll Ihnen den Leistungsumfang des System 300V verdeutlichen:



Komponenten

Zentrales System Im System 300V stehen verschiedene SPS-CPU's zur Verfügung. Programmiert wird in STEP®7 von Siemens. Hierzu können Sie WinPLC7 von VIPA oder den SIMATIC Manager von Siemens verwenden.

CPU's mit integrierter Ethernet- bzw. Bus-Anschaltung 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.

Aufgrund der automatischen Adressierung können bei Einsatz der System 300V CPU's bis zu 32 Peripherie-Module angesprochen werden.

Dezentrales System Die SPS-CPU's bilden in Kombination mit einem Profibus DP-Master die Basis für ein Profibus-DP-Netzwerk nach DIN 19245-3.

Zur Projektierung des DP-Netzwerks können Sie den Hardware-Konfigurator von Siemens verwenden. Zusammen mit der Hardware-konfiguration übertragen Sie via MPI Ihr Projekt in Ihre CPU.

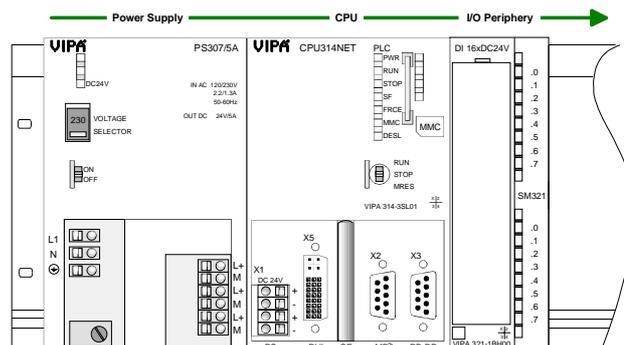
Eine weitere Komponente des dezentralen Systems ist der CAN-Slave. Er ermöglicht den Anschluss an das Feldbussystem CANopen.

Peripheriemodule Von VIPA erhalten Sie eine Vielzahl an Peripheriemodulen wie z.B. für digitale bzw. analoge Ein-/Ausgabe.

Die Peripheriemodule können zentral und dezentral betrieben werden.

- Aufbau/Maße**
- Profilschiene 530mm
 - Peripherie-Module mit Beschriftungsstreifen
 - Maße Grundgehäuse:
 - 1fach breit: (BxHxT) in mm: 40x125x120
 - 2fach breit: (BxHxT) in mm: 80x125x120
 - 3fach breit: (BxHxT) in mm: 120x125x120

Montage Bitte beachten Sie, dass Sie die Stromversorgung und Kopfmodule wie CPU's und Koppler nur links stecken dürfen.



- Betriebssicherheit**
- Anschluss über Federzugklemmen an Frontstecker
 - Aderquerschnitt 0,08...2,5mm² bzw. 1,5 mm²
 - 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
- Green Cable zur Projektierung**
- Zur Projektierung Ihres DP-Slaves können Sie Ihre Projekte von Ihrem PC über MPI seriell an die CPU übertragen, indem Sie das "Green Cable" verwenden. Bitte beachten Sie auch in diesem Zusammenhang die Hinweise zum Green Cable in diesem Kapitel!
- Integriertes Netzteil**
- Jede CPU bzw. jeder Buskoppler 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.
- Kompatibilität**
- Die Digitalen Ein-/Ausgabe-Module des System 300V von VIPA sind pin- und funktionskompatibel zu Siemens.
Die Projektierung erfolgt im SIMATIC Manager von Siemens.

**Hinweis!**

Bitte verwenden Sie zur Projektierung einer System 300V CPU von VIPA immer die **CPU 315-2DP (6ES7 315-2AF03 V1.2)** von Siemens aus dem Hardware-Katalog.

Bitte beachten Sie dass bei der CPU 31x die Profibus-Adresse 1 systembedingt reserviert ist.

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

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 System 300V erforderlich sind.

Inhalt

| Thema | Seite |
|---|------------|
| Teil 2 Montage und Aufbaurichtlinien | 2-1 |
| Übersicht | 2-2 |
| Einbaumaße | 2-3 |
| Montage auf Profilschiene | 2-4 |
| Verdrahtung | 2-6 |
| Aufbaurichtlinien | 2-10 |

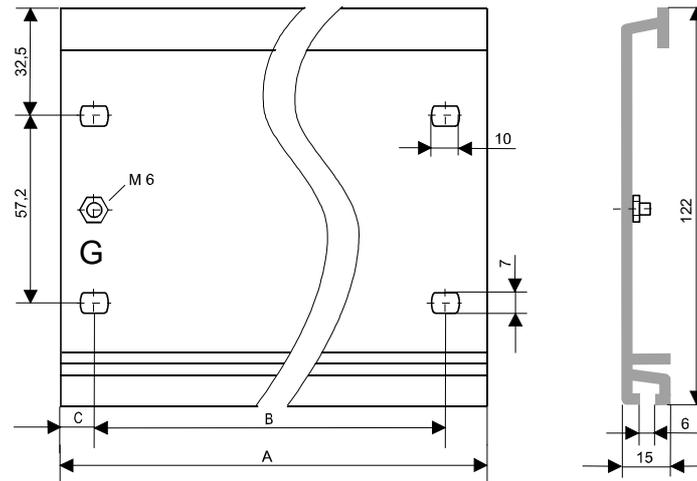
Übersicht

Allgemein

Die einzelnen Module werden direkt auf eine Profilschiene montiert und über den Rückwandbus-Verbinder verbunden. Vor der Montage ist der Rückwandbus-Verbinder von hinten an das Modul zu stecken.

Die Rückwandbusverbinder sind im Lieferumfang der Peripherie-Module enthalten.

Profilschiene

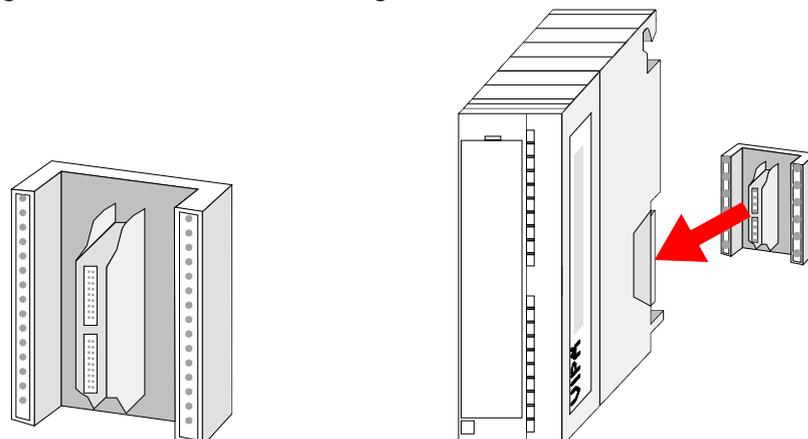


| Bestellnummer | A | B | C |
|-----------------|--------|-----------------|-------|
| VIPA 390-1AB60 | 160mm | 140mm | 10mm |
| VIPA 390-1AE80 | 482mm | 466mm | 8,3mm |
| VIPA 390-1AF30 | 530mm | 500mm | 15mm |
| VIPA 390-1AJ30 | 830mm | 800mm | 15mm |
| VIPA 390-9BC00* | 2000mm | keine Bohrungen | 15mm |

* Verpackungseinheit 10 Stück

Busverbinder

Für die Kommunikation der Module untereinander wird beim System 300V ein Rückwandbus-Verbinder eingesetzt. Die Rückwandbus-Verbinder sind im Lieferumfang der Peripherie-Module enthalten und werden vor der Montage von hinten an das Modul gesteckt.

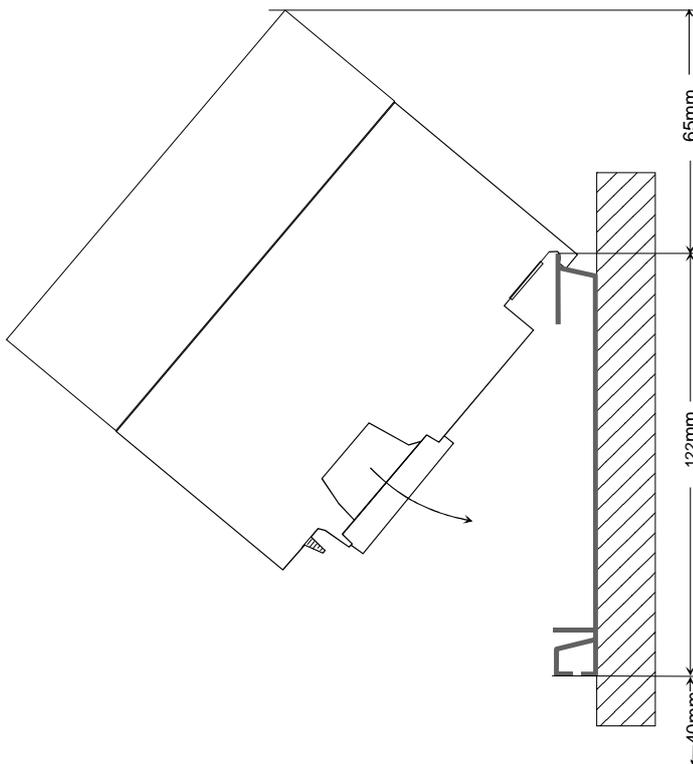


Einbaumaße

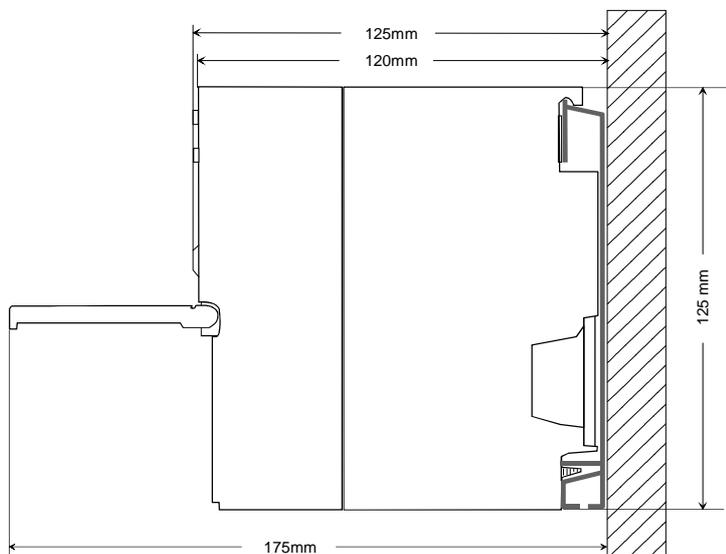
Übersicht Hier finden Sie alle wichtigen Maße des System 300V.

Maße Grundgehäuse
 1fach breit (BxHxT) in mm: 40 x 125 x 120
 2fach breit (BxHxT) in mm: 80 x 125 x 120
 3fach breit (BxHxT) in mm: 120 x 125 x 120

Montagemaße



Maße montiert



Montage auf Profilschiene

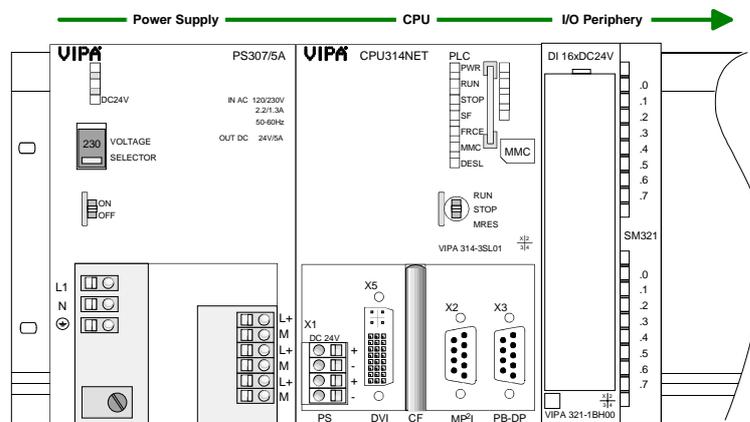
Aufbau:

Sie haben die Möglichkeit das System 300V waagrecht oder senkrecht aufzubauen. Beachten Sie bitte die hierbei zulässigen Umgebungstemperaturen:

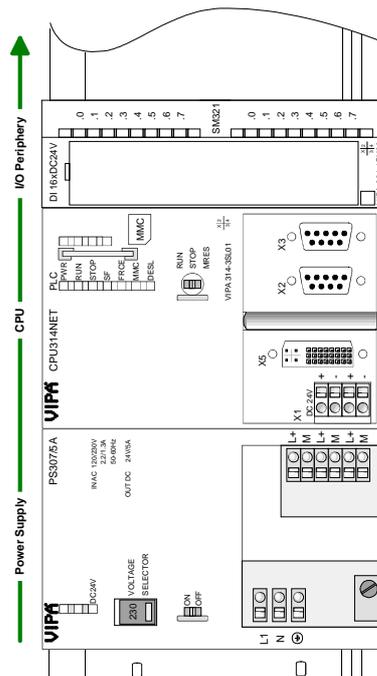
- waagrechter Aufbau: von 0 bis 60°
- senkrechter Aufbau: von 0 bis 40°

Der waagrechte Aufbau beginnt immer links mit der Stromversorgung und der CPU, rechts daneben werden die Peripherie-Module gesteckt.

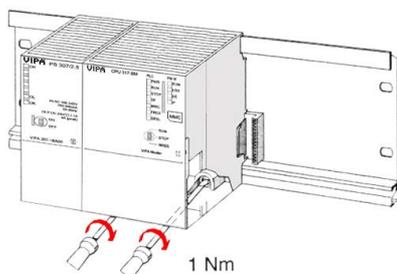
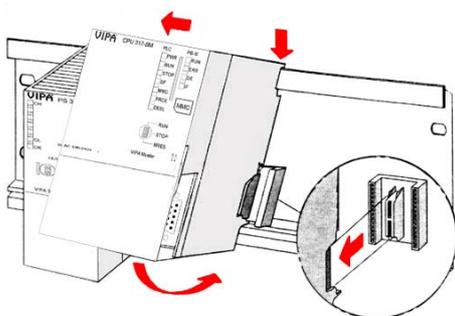
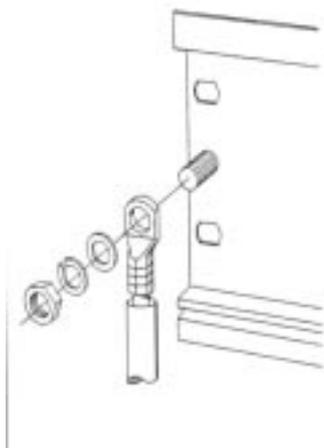
Es dürfen maximal 32 Peripherie-Module neben die CPU gesteckt werden.



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



Vorgehensweise



- Verschrauben Sie die Profilschiene mit dem Untergrund (Schraubengröße: M6) so, dass mindestens 65mm Raum oberhalb und 40mm unterhalb der Profilschiene bleibt.
- Wenn der Untergrund eine geerdete Metallplatte oder ein geerdetes Geräteblech ist, achten Sie auf eine niederohmige Verbindung zwischen Profilschiene und Untergrund.
- Verbinden Sie die Profilschiene mit dem Schutzleiter. Für diesen Zweck befindet sich auf der Profilschiene ein Stehbolzen mit M6-Gewinde.
- Der Mindestquerschnitt der Leitung zum Schutzleiter muss 10mm^2 betragen.
- Hängen Sie die Spannungsversorgung ein und schieben Sie diese nach links bis an den Erdungsbolzen der Profilschiene.
- Schrauben Sie die Spannungsversorgung fest.
- Nehmen Sie einen Busverbinder und stecken Sie ihn, wie gezeigt, von hinten an die CPU.
- Hängen Sie die CPU rechts von der Spannungsversorgung ein und schieben Sie diese bis an die Spannungsversorgung.
- Klappen Sie die CPU nach unten und schrauben Sie die CPU, wie gezeigt, fest.
- Verfahren Sie auf die gleiche Weise mit Ihren Peripherie-Modulen, indem Sie jeweils einen Rückwandbus-Verbinder stecken, Ihr Modul rechts des Vorgänger-Moduls einhängen, dieses nach unten klappen, in den Rückwandbus-Verbinder des Vorgängermoduls einrasten lassen und das Modul festschrauben.



Gefahr!

- Die Spannungsversorgungen sind vor dem Beginn von Installations- und Instandhaltungsarbeiten unbedingt freizuschalten, d.h. vor Arbeiten an einer Spannungsversorgung oder an der Zuleitung, ist die Spannungszuführung stromlos zu schalten (Stecker ziehen, bei Festanschluss ist die zugehörige Sicherung abzuschalten)!
- Anschluss und Änderungen dürfen nur durch ausgebildetes Elektro-Fachpersonal ausgeführt werden!

Verdrahtung

Übersicht

Die Spannungsversorgungen und CPUs werden ausschließlich mit Federklemm-Kontakten ausgeliefert. Für die Signalbaugruppen sind bei VIPA die Frontstecker mit Schraubkontakten erhältlich. Nachfolgend sind alle Anschlussarten der Spannungsversorgungen, CPUs und Ein-/Ausgabe-Module aufgeführt.



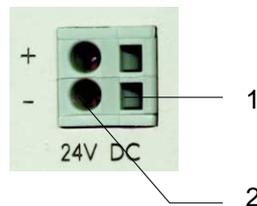
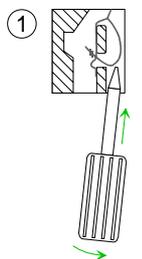
Gefahr!

- Die Spannungsversorgungen sind vor dem Beginn von Installations- und Instandhaltungsarbeiten unbedingt freizuschalten, d.h. vor Arbeiten an einer Spannungsversorgung oder an der Zuleitung, ist die Spannungszuführung stromlos zu schalten (Stecker ziehen, bei Festanschluss ist die zugehörige Sicherung abzuschalten)!
- Anschluss und Änderungen dürfen nur durch ausgebildetes Elektro-Fachpersonal ausgeführt werden!

Federklemmtechnik (grau)

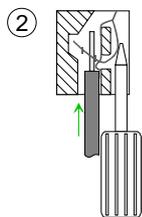
Für die Verdrahtung von Spannungsversorgungen, Buskopplern und Teilen der CPU werden graue Anschlussklemmen mit Federklemmtechnik eingesetzt. Sie können Drähte mit einem Querschnitt von $0,08\text{mm}^2$ bis $2,5\text{mm}^2$ anschließen. Es können sowohl flexible Litzen ohne Aderendhülse, als auch starre Leiter verwendet werden.

Die Leitungen befestigen Sie wie folgt an den Federklemmkontakten:



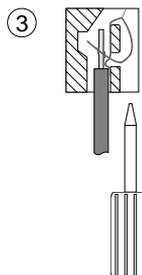
[1] Rechteckige Öffnung für Schraubendreher

[2] Runde Öffnung für Drähte



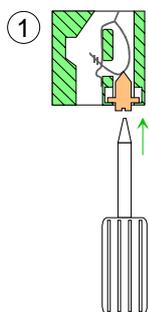
Die nebenstehende Abfolge stellt die Schritte der Verdrahtung in der Draufsicht dar.

- Zum Verdrahten stecken Sie wie in der Abbildung gezeigt einen passenden Schraubendreher leicht schräg in die rechteckige Öffnung.
- Zum Öffnen der Kontaktfeder müssen Sie den Schraubendreher in die entgegen gesetzte Richtung drücken und halten.
- Führen Sie durch die runde Öffnung Ihren abisolierten Draht ein. Sie können Drähte mit einem Querschnitt von $0,08\text{mm}^2$ bis $2,5\text{mm}^2$ anschließen.
- Durch Entfernen des Schraubendrehers wird der Draht über einen Federkontakt sicher mit dem Steckverbinder verbunden.

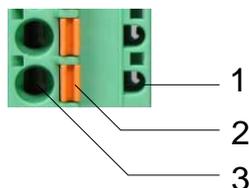


**Federklemmtechnik
(grün)**

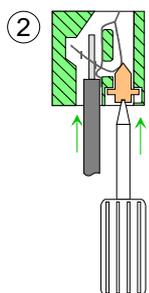
Zur Verdrahtung der Spannungsversorgung einer CPU beispielsweise kommen grüne Stecker mit Federzugklemmtechnik zum Einsatz.



Auch hier können Sie Drähte mit einem Querschnitt von $0,08\text{mm}^2$ bis $2,5\text{mm}^2$ anschließen. Hierbei dürfen sowohl flexible Litzen ohne Aderendhülle, als auch starre Leiter verwendet werden.

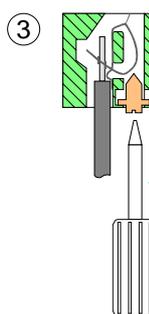


- [1] Prüfabgriff für 2mm Messspitze
- [2] Verriegelung (orange) für Schraubendreher
- [3] Runde Öffnung für Drähte



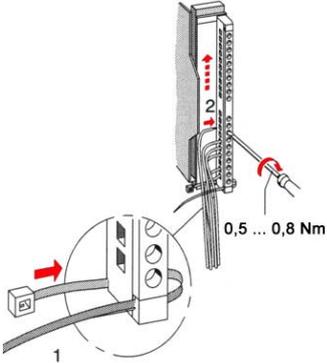
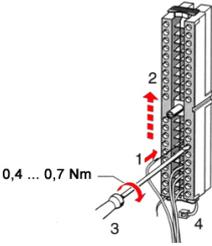
Die nebenstehende Abfolge stellt die Schritte der Verdrahtung in der Draufsicht dar.

- Zum Verdrahten drücken Sie mit einem geeigneten Schraubendreher, wie in der Abbildung gezeigt, die Verriegelung senkrecht nach innen und halten Sie den Schraubendreher in dieser Position.
- Führen Sie durch die runde Öffnung Ihren abisolierten Draht ein. Sie können Drähte mit einem Querschnitt von $0,08\text{mm}^2$ bis $2,5\text{mm}^2$ anschließen.
- Durch Entfernen des Schraubendrehers wird der Draht über einen Federkontakt sicher mit dem Steckverbinder verbunden.

**Hinweis!**

Im Gegensatz zur grauen Anschlussklemme, die weiter oben beschrieben ist, ist die grüne Anschlussklemme als Stecker ausgeführt, der im verdrahteten Zustand vorsichtig abgezogen werden kann.

Frontstecker der Ein-/Ausgabe-Module Nachfolgend ist die Verdrahtung der 2 Frontstecker-Varianten aufgezeigt:
 Für die Ein-/Ausgabe-Module sind bei VIPA folgende Stecker erhältlich:

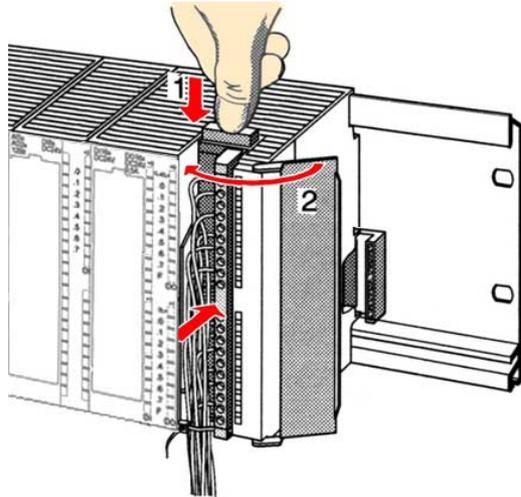
| <p align="center">20-fach Schraubtechnik VIPA 392-1AJ00</p> | <p align="center">40-fach Schraubtechnik VIPA 392-1AM00</p> |
|---|--|
|  |  |
| <p>Öffnen Sie die Frontklappe Ihres Ein-/Ausgabe-Moduls.</p> | |
| <p>Bringen Sie den Frontstecker in Verdrahtungsstellung. Hierzu stecken Sie den Frontstecker auf das Modul, bis er einrastet. In dieser Stellung ragt der Frontstecker aus dem Modul heraus und hat noch keinen Kontakt.</p> | |
| <p>Isolieren Sie Ihre Leitungen ab. Verwenden Sie ggf. Aderendhülsen.</p> | |
| <p>Fädeln Sie den beiliegenden Kabelbinder in den Frontstecker ein.</p> | |
| <p>Beginnen Sie mit der Verdrahtung von unten nach oben, wenn Sie die Leitungen nach unten aus dem Modul herausführen möchten, bzw. von oben nach unten, wenn die Leitungen nach oben herausgeführt werden sollen.</p> | |
| <p>Schrauben Sie die Anschlussschrauben der nicht verdrahteten Schraubklemmen ebenfalls fest.</p> | |
|  | <p>Legen Sie den beigefügten Kabelbinder um den Leitungsstrang und den Frontstecker herum.</p>  |
| <p>Ziehen Sie den Kabelbinder für den Leitungsstrang fest.</p> | |

Fortsetzung ...

... Fortsetzung

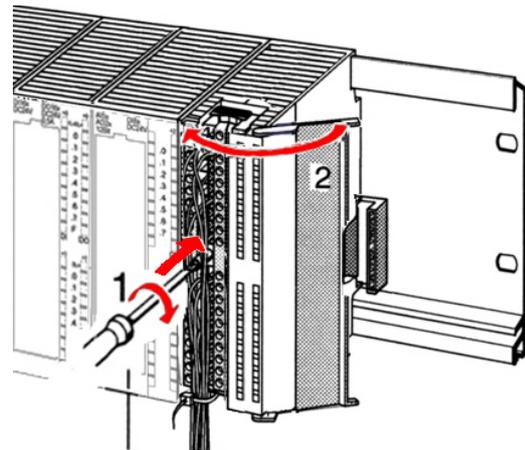
20-fach Schraubtechnik

Drücken Sie die Entriegelungstaste am Frontstecker an der Moduloberseite und drücken Sie gleichzeitig den Frontstecker in das Modul bis er einrastet.



40-fach Schraubtechnik

Schrauben Sie die Befestigungsschraube für den Frontstecker fest.



0.4 ... 0.7 Nm

Der Frontstecker ist nun elektrisch mit Ihrem Modul verbunden.

Schließen Sie die Frontklappe.

Füllen Sie den Beschriftungsstreifen zur Kennzeichnung der einzelnen Kanäle aus und schieben Sie den Streifen in die Frontklappe.

Aufbaurichtlinien

| | |
|------------------------------|---|
| Allgemeines | <p>Die Aufbaurichtlinien enthalten Informationen über den störsicheren Aufbau des System 300V. 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> |
| Was bedeutet EMV? | <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 300V 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> |
| Mögliche Störeinträge | <p>Elektromagnetische Störungen können sich auf unterschiedlichen Pfaden in Ihre Steuerung einkoppeln:</p> <ul style="list-style-type: none">• Felder• E/A-Signalleitungen• Bussystem• Stromversorgung• Schutzleitung <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">• galvanische Kopplung• kapazitive Kopplung• induktive Kopplung• Strahlungskopplung |

Grundregeln zur Sicherstellung der EMV

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

- Achten Sie bei der Montage Ihrer Komponenten auf eine gut ausgeführte flächenhafte Massung der inaktiven Metallteile.
 - Stellen Sie eine zentrale Verbindung zwischen der Masse und dem Erde/Schutzleitersystem her.
 - Verbinden Sie alle inaktiven Metallteile großflächig und impedanzarm.
 - Verwenden Sie nach Möglichkeit keine Aluminiumteile. Aluminium oxidiert leicht und ist für die Massung deshalb weniger gut geeignet.
- Achten Sie bei der Verdrahtung auf eine ordnungsgemäße Leitungsführung.
 - Teilen Sie die Verkabelung in Leitungsgruppen ein. (Starkstrom, Stromversorgungs-, Signal- und Datenleitungen).
 - Verlegen Sie Starkstromleitungen und Signal- bzw. Datenleitungen immer in getrennten Kanälen oder Bündeln.
 - Führen Sie Signal- und Datenleitungen möglichst eng an Masseflächen (z.B. Tragholme, Metallschienen, Schrankbleche).
- Achten Sie auf die einwandfreie Befestigung der Leitungsschirme.
 - Datenleitungen sind geschirmt zu verlegen.
 - Analogleitungen sind geschirmt zu verlegen. Bei der Übertragung von Signalen mit kleinen Amplituden kann das einseitige Auflegen des Schirms vorteilhaft sein.
 - Legen Sie die Leitungsschirme direkt nach dem Schrankeintritt großflächig auf eine Schirm-/Schutzleiterschiene auf und befestigen Sie die Schirme mit Kabelschellen.
 - Achten Sie darauf, dass die Schirm-/Schutzleiterschiene impedanzarm mit dem Schrank verbunden ist.
 - Verwenden Sie für geschirmte Datenleitungen metallische oder metallisierte Steckergehäuse.
- Setzen Sie in besonderen Anwendungsfällen spezielle EMV-Maßnahmen ein.
 - Beschalten Sie alle Induktivitäten mit Löschgliedern, die nicht von System 300V 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 300V 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. μA) 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 300V 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 der Profibus-DP-Slave im System 300. Nach einer kurzen Einführung und Systemvorstellung erhalten Sie hier alle Informationen zu Aufbau, Projektierung und Diagnose. Die technischen Daten runden das Kapitel ab.

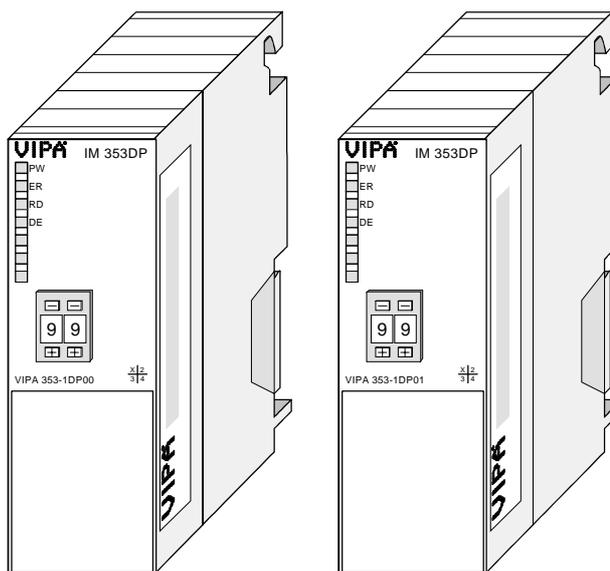
Inhalt

| Thema | Seite |
|---|------------|
| Teil 3 Profibus-DP | 3-1 |
| Systemübersicht..... | 3-2 |
| Grundlagen | 3-3 |
| IM 353-1DP00 - DP-V0-Slave - Aufbau | 3-12 |
| IM 353-1DP00 - DP-V0-Slave - Projektierung | 3-15 |
| IM 353-1DP00 - DP-V0-Slave - Diagnosefunktionen | 3-16 |
| IM 353-1DP01 - DP-V1-Slave - Aufbau | 3-22 |
| IM 353-1DP01 - DP-V1-Slave - Projektierung | 3-25 |
| IM 353-1DP01 - DP-V1-Slave - DP-V1-Dienste..... | 3-28 |
| IM 353-1DP01 - DP-V1-Slave - Diagnosefunktionen | 3-30 |
| Aufbaurichtlinien..... | 3-38 |
| Inbetriebnahme | 3-43 |
| Einsatz der Diagnose-LEDs | 3-45 |
| Technische Daten | 3-46 |

Systemübersicht

Folgende Profibus-DP-Slaves sind zur Zeit für das System 300 erhältlich:

- Profibus-DP-Slave IM 353DP mit DP-V0
- Profibus-DP-Slave IM 353DP mit DP-V0 / DP-V1



Bestelldaten

| Typ | Bestellnummer | Beschreibung | Seite |
|----------|----------------|-------------------------|-------|
| IM 353DP | VIPA 353-1DP00 | Profibus-DP-V0-Slave | 3-12 |
| IM 353DP | VIPA 353-1DP01 | Profibus-DP-V0/V1-Slave | 3-22 |

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 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 sind modulare Slave-Geräte, die Daten zwischen der System 300V 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 angebotenen 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.

| | |
|-------------------------------|---|
| Kommunikation | Das Busübertragungsprotokoll bietet zwei Verfahren für den Buszugriff: |
| Master mit Master | <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> |
| Master-Slave-Verfahren | <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> |
| Datenkonsistenz | <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. Profibus garantiert die Konsistenz mit der erforderlichen Länge.</p> |

Einschränkungen

- Peripherie-Module dürfen nur nach Power-Off gesteckt oder gezogen werden!
- max. Leitungslänge unter RS485 zwischen zwei Stationen 1200m (baudratenabhängig)
- Die maximale Baudrate liegt bei 12MBAud.

Diagnose

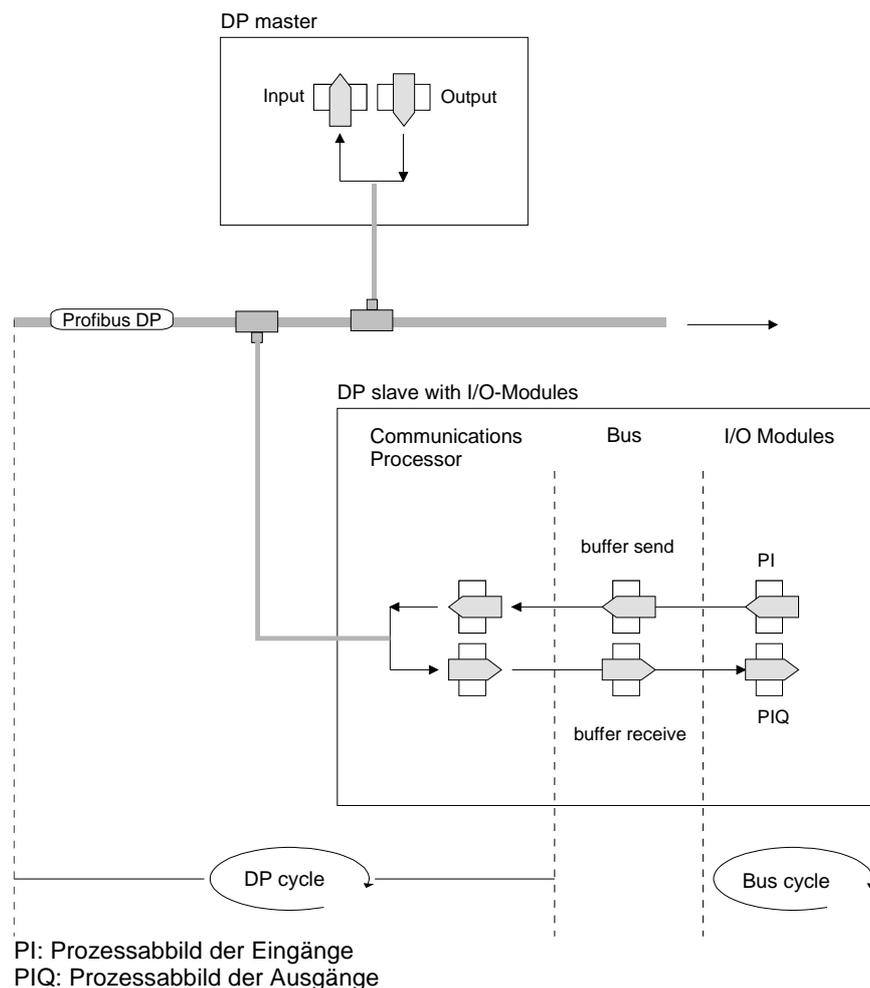
Die umfangreichen Diagnosefunktionen unter Profibus-DP ermöglichen eine schnelle Fehlerlokalisierung. Die Diagnosedaten werden über den Bus übertragen und beim Master zusammengefasst.

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.



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

DP-Zyklus In einem Profibus-Zyklus spricht der DP-Master alle seine DP-Slaves der Reihe nach mit einem Data Exchange an. Beim Data Exchange 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 Bus-Zyklus.

Bus-Zyklus \leq DP-Zyklus Zur Gewährleistung einer zeitgleichen Datenübertragung sollte die Bus-Zykluszeit immer kleiner oder gleich der DP-Zykluszeit sein. In der mitgelieferten GSD-Datei befindet sich der Parameter **min_slave_interval = 3ms**. Für einen durchschnittlichen Aufbau wird garantiert, dass spätestens nach 3ms die Profibus-Daten am Bus aktualisiert wurden. Sie dürfen also alle 3ms einen DataExchange mit dem Slave ausführen.



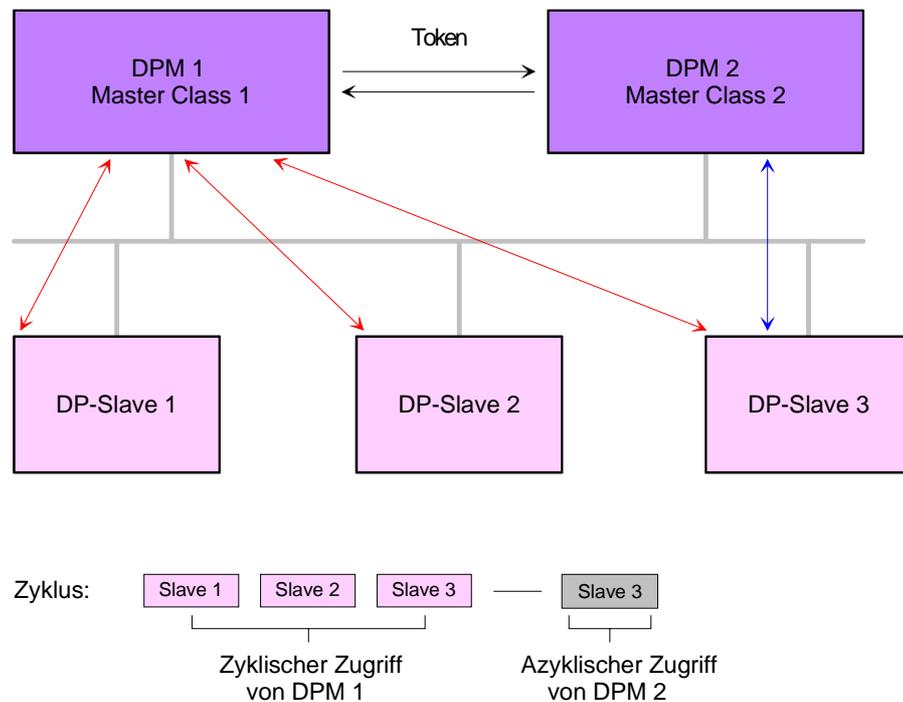
Hinweis!

Ab Ausgabestand 6 erlischt bei einem DP-V0-Slave die RUN-LED, sobald der 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).

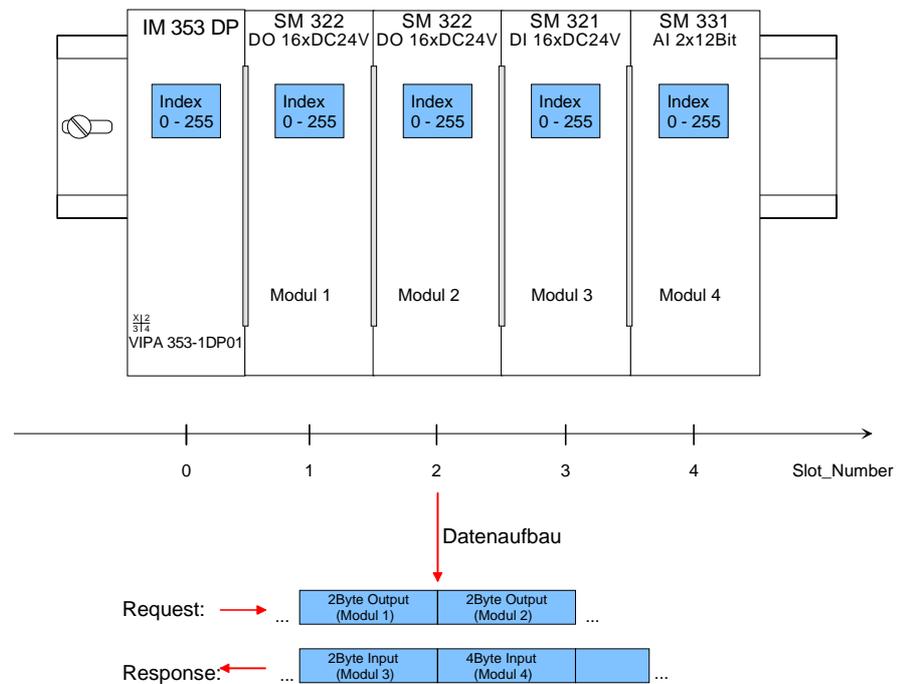
Adressierung mit Slot und Index

Bei der Adressierung von Daten geht Profibus davon aus, dass die Slaves 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 DP-Master 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* 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.



Hinweis!

Folgende Konventionen gelten für die Adressierung bei Einsatz des Siemens SIMATIC Manager:

DP-Slave-Koppler: Angabe der *Diagnoseadresse* als ID

Module des DP-Slave-Kopplers: Angabe der *Moduladresse* als ID. Hierbei muss bei einer Ausgabebaugruppe zusätzlich Bit 15 der Moduladresse gesetzt sein (z.B. aus Adresse 0004h wird 8004h). Bei einer Mischbaugruppe ist die kleinere der beiden Adressen anzugeben.

**Dienste
azyklischer
Datenverkehr**

Für den Einsatz der DP-V1 Dienste ist darauf zu achten, dass Ihre CPU DP-V1-Kommunikation unterstützt. Näheres hierzu finden Sie in der Beschreibung Ihrer CPU. Hierbei stehen Ihnen folgende System-Funktions-Bausteine zur Verfügung:

- SFB 52 Datensatz aus einem DP-Slave lesen
- SFB 53 Datensatz in einen DP-Slave schreiben
- SFB 54 Alarm von einem DP-Slave empfangen

Nachfolgend sind die Dienste für den azyklischen Datenverkehr aufgeführt, die diese Funktionsbausteine verwenden.

Nähere Informationen zu den Diensten und zu den DP-V0/V1-Kommunikationsprinzipien finden Sie in der Profibus-Norm IEC 61158.

**DPM 1
(Master Klasse 1)**

| 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 wird 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
(Master Klasse 2)**

| 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. | |

RS485 Schnittstelle als Übertragungsmedium

Profibus verwendet als Übertragungsmedium eine geschirmte, verdrehte Zweidrahtleitung auf Basis der RS485-Schnittstelle. Die Übertragungsrate liegt bei maximal 12MBAud.

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 DP-Slave befindet sich eine 9polige Buchse. Über diese Buchse koppeln Sie den Profibus-DP-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.

Adressierung

Jeder Teilnehmer am Profibus identifiziert sich mit einer Adresse. Diese Adresse darf nur einmal in diesem Bussystem vergeben sein und kann bei den VIPA System 300V Slaves zwischen 1 und 99 liegen.

GSD-Datei

Von VIPA erhalten Sie jeden Profibus Slave zusammen mit einem Datenträger. Auf dem Datenträger befinden sich unter anderem alle GSD-Dateien der Profibus-Module von VIPA.

Die Zuordnung der GSD-Datei zu Ihrem DP-Slave entnehmen Sie bitte der folgenden Tabelle:

| Bestellnummer | GSD-Datei |
|-----------------------|---------------------------|
| VIPA 353-1DP00 | VIPA056B.gsd |
| VIPA 353-1DP01(DP-V0) | VI0009AF.gsd [*] |
| VIPA 353-1DP01(DP-V1) | VI0109AF.gsd |

^{*} Diese GSD-Datei ist für Profibus-Master zu verwenden, die kein DP-V1 unterstützen.

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

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-V0-Slave im Hardware-Katalog von Siemens unter:

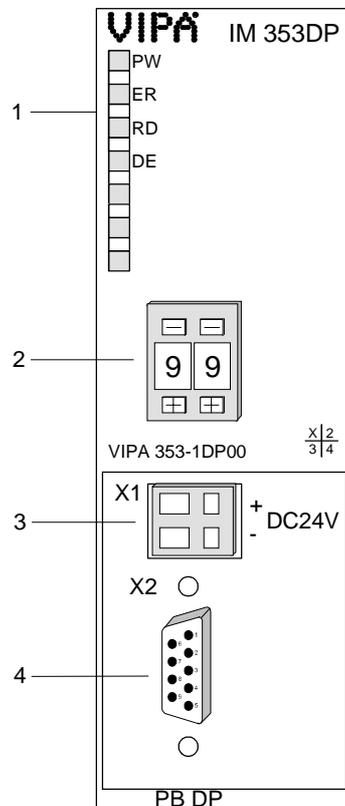
*Profibus-DP>Weitere Feldgeräte>I/O>VIPA_System_300V>
VIPA 353-1DP00*

IM 353-1DP00 - DP-V0-Slave - Aufbau

Eigenschaften

- Profibus DP-Slave 9,6kBaude bis 12Mbaude
- Profibus-DP Slave für max. 32 Peripherie-Module (max. 16 Analog-Module)
- Max. 152Byte Eingabe- und 152Byte Ausgabe-Daten
- LEDs für Busdiagnose
- Internes Diagnoseprotokoll mit Zeitstempel
- Integriertes DC 24V-Netzteil zur Versorgung der Peripherie-Module (max. 3,5A)

Frontansicht 353-1DP00



- [1] LED Statusanzeigen
- [2] Adresseinsteller

Folgende Komponenten befinden sich unter der Frontklappe

- [3] Anschluss für DC24V-Spannungsversorgung
- [4] RS485-Schnittstelle

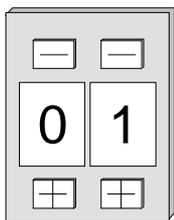
Komponenten

LEDs

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

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

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.



Hinweis!

Bitte beachten Sie, dass zur eindeutigen Identifikation des Profibus-Slaves, die bei der Projektierung vorgegebene Profibusadresse und die am Adress-Schalter eingestellte Adresse identisch ist.

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.



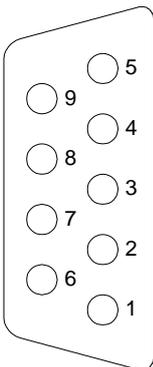
Achtung!

Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung!

RS485-Schnittstelle

Über eine 9polige 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. |

IM 353-1DP00 - DP-V0-Slave - Projektierung

Allgemeines

Die Projektierung erfolgt als Hardware-Konfiguration in Ihrem Profibus-DP-Master Projektierool wie beispielsweise dem Siemens SIMATIC Manager. Hierbei ordnen Sie Ihrem DP-Master das entsprechenden Profibus-DP-Slave-Modul zu. Eine direkte Zuordnung erfolgt über die Profibus-Adresse, die Sie am DP-Slave über den Adress-Schalter und in den DP-Slave-Eigenschaften einzustellen haben.

Durch Einbindung der GSD-Datei VIPA056B.gsd wird der IM 353-1DP00 DP-Slave als "VIPA_DP300V" im Hardware-Katalog aufgeführt unter:

Profibus-DP>Weitere Feldgeräte>I/O>VIPA_System_300V

GSD-Datei VIPA056B.gsd

Von VIPA erhalten Sie den IM 353-1DP00 DP-Slave zusammen mit einem Datenträger. Auf dem Datenträger befinden sich unter anderem alle GSD-Dateien der Profibus-Module von VIPA.

Installieren Sie die Datei VIPA056B.gsd in Ihrem Projektierool. Nähere Hinweise zur Installation einer GSD-Datei finden Sie im Handbuch zu Ihrem Projektierool.

Projektierung

- Bauen Sie Ihr Profibus-System auf.
- Starten Sie Ihr Projektierool mit einem neuen Projekt.
- Projektieren Sie ein Master-System und legen Sie ein neues Profibus-Subnetz an.
- Zur Projektierung des IM 353-1DP00 entnehmen Sie den "VIPA_DP300V" aus dem Hardware-Katalog und ziehen Sie diesen auf das DP-Master Subnetz.
- Geben Sie in den Eigenschaften des DP-Slave eine Profibus-Adresse zwischen 1 und 99 an und stellen Sie diese Adresse am Adress-Schalter ein.
- Parametrieren Sie den DP-Slave (siehe Parameter).
- Übertragen Sie Ihr Projekt in die SPS.

Parameter

Diagnosealarm

Sie haben die Möglichkeit einen Diagnosealarm zu aktivieren, den Sie durch Anwahl von "JA" aktivieren bzw. "NEIN" deaktivieren können.

Im Auslieferungszustand ist der Diagnosealarm aktiviert. Hierbei verzweigt die CPU im Fehlerfall in den OB 82. Dort können Sie auf den Fehler reagieren und unter Einsatz des SFC 13 die Diagnose auswerten. Aufbau und die Auswertung der Diagnose des IM 353-1DP00 ist auf den nachfolgenden Seiten beschrieben.



Hinweis

Bei jeder Änderung der Modulzusammenstellung sind die Busparameter neu zu berechnen!

IM 353-1DP00 - 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.

Sofern Sie den Diagnosealarm über die Parametrierung aktiviert haben, sendet der DP-Slave auf Anforderung vom Master oder im Fehlerfall Diagnosedaten. Hierbei verzweigt die CPU in den OB 82. Dort können Sie auf den Fehler reagieren und unter Einsatz des SFC 13 die Diagnose auswerten.

Zusätzlich werden bei dem DP-Slave 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!

Diagnosedaten manuell sichern

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

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.

Diagnosemeldung bei Spannungs- ausfall

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.

Diagnosedaten IM 353-1DP00

Die Diagnosedaten bestehen aus:

- Norm-Diagnosedaten (Byte 0 ... 5)
- Kennungsbezogene Diagnosedaten (Byte 6 ... 10)
- Modulbezogene Diagnosedaten (Byte 11 ... 22)
- Kanalbezogene Diagnosedaten (Byte 23 ... 25)

Norm-Diagnose

| | |
|--------|---------------------|
| Byte 0 | Stationsstatus 1 |
| Byte 1 | Stationsstatus 2 |
| Byte 2 | Stationsstatus 3 |
| Byte 3 | Master-Adresse |
| Byte 4 | Ident-Nummer (low) |
| Byte 5 | Ident-Nummer (high) |

Kennungsbezogene Diagnose

| | |
|--------------------|--------------------------------------|
| Byte 6 | Länge und Code Kennungsbez. Diagnose |
| Byte 7 ... Byte 10 | Kennungsbezogene Diagnosemeldungen |

Modulbezogene Diagnose

| | |
|---------------------|---------------------------------------|
| Byte 11 | Länge und Code Modulbezogene Diagnose |
| Byte 12 | Modulstatus |
| Byte 13 ... Byte 14 | 0 (fix) |
| Byte 15 ... Byte 22 | Modulbezogene Diagnosemeldungen |

Kanalbezogene Diagnose

| | |
|---------|---------------------------|
| Byte 23 | Kennungsnummer des Moduls |
| Byte 24 | Nummer des Kanals |
| Byte 25 | Fehler-Code |

**Norm-
Diagnosedaten
IM 353-1DP00**

Die Norm-Diagnosedaten haben folgenden Aufbau:

Norm-Diagnose

| Byte | Bit 7 ... Bit 0 |
|------|---|
| 0 | Bit 0: 0 (fix) 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: 0 (fix) Bit 6: Falsche Parametrierung Bit 7: 0 (fix) |
| 1 | Bit 0: Slave muss neu parametrierung werden Bit 1: Statistische Diagnose Bit 2: 1 (fix) Bit 3: Ansprechüberwachung aktiv Bit 4: Freeze-Kommando erhalten Bit 5: Sync-Kommando erhalten Bit 6: reserviert Bit 7: 0 (fix) |
| 2 | Bit 6 ... 0: reserviert Bit 7: Diagnosedaten Überlauf |
| 3 | Masteradresse nach Parametrierung FFh: Slave ist ohne Parametrierung |
| 4 | Identnummer High Byte |
| 5 | Identnummer Low Byte |

**Kennungsbezogene
Diagnosedaten
IM 353-1DP00**

Die Kennungsbezogene Diagnose sagt aus, ob Module des System 300V DP-Slave fehlerhaft sind oder nicht.

Die Länge der Kennungsbezogenen Diagnosedaten ist fest auf 5Byte eingestellt.

Kennungsbezogene Diagnose

| Byte | Bit 7 ... Bit 0 |
|------|--|
| 6 | Bit 5 ... 0: Länge kennungsbezogene Diagnosedaten 000101: Länge 5Byte (fix) Bit 7 ... 6: Code für kennungsbezogene Diagnose 01: Code 01 (fix) |
| 7 | Bit 0: Modul auf Steckplatz 1 Bit 1: Modul auf Steckplatz 2 Bit 2: Modul auf Steckplatz 3 Bit 3: Modul auf Steckplatz 4 Bit 4: Modul auf Steckplatz 5 Bit 5: Modul auf Steckplatz 6 Bit 6: Modul auf Steckplatz 7 Bit 7: Modul auf Steckplatz 8 |
| 8 | Bit 0: Modul auf Steckplatz 9 Bit 1: Modul auf Steckplatz 10 . . . Bit 7: Modul auf Steckplatz 16 |
| 9 | Bit 0: Modul auf Steckplatz 17 Bit 1: Modul auf Steckplatz 18 . . . Bit 7: Modul auf Steckplatz 24 |
| 10 | Bit 0: Modul auf Steckplatz 25 Bit 1: Modul auf Steckplatz 26 . . . Bit 7: Modul auf Steckplatz 32 |

Die Bits in Byte 7 und 10 werden gesetzt, wenn:

- das entsprechende Modul gezogen wird.
- ein nicht projektiertes Modul gesteckt wird.
- auf ein gestecktes Modul nicht zugegriffen werden kann.
- das entsprechende Modul einen Diagnosealarm meldet.

**Modulbezogene
Diagnosedaten
IM 353-1DP00**

Die Modulbezogene Diagnose gibt den Status der projektierten Module wieder und stellt eine Detaillierung der Kennungsbezogenen Diagnose bezüglich der Konfiguration dar.

Die Länge der Modulbezogenen Diagnosedaten ist fest auf 12Byte eingestellt.

Modulstatus

| Byte | Bit 7 ... Bit 0 |
|------|--|
| 11 | Bit 5 ... 0: Länge modulbezogene Diagnosedaten 001100: Länge 12Byte (fix) Bit 7 ... 6: Code für modulbezogene Diagnose 00: Code 00 (fix) |
| 12 | Bit 6 ... 0: Statustyp 0000010: Modulstatus Bit 7: 1 (fix) |
| 13 | immer 00 |
| 14 | immer 00 |
| 15 | Bit 1 ... 0: Modul auf Steckplatz 1 Bit 3 ... 2: Modul auf Steckplatz 2 Bit 5 ... 4: Modul auf Steckplatz 3 Bit 7 ... 6: Modul auf Steckplatz 4 |
| 16 | Bit 1 ... 0: Modul auf Steckplatz 5 : Bit 7 ... 6: Modul auf Steckplatz 8 |
| 17 | Bit 1 ... 0: Modul auf Steckplatz 9 : Bit 7 ... 6: Modul auf Steckplatz 12 |
| 18 | Bit 1 ... 0: Modul auf Steckplatz 13 : Bit 7 ... 6: Modul auf Steckplatz 16 |
| 19 | Bit 1 ... 0: Modul auf Steckplatz 17 : Bit 7 ... 6: Modul auf Steckplatz 20 |
| 20 | Bit 1 ... 0: Modul auf Steckplatz 21 : Bit 7 ... 6: Modul auf Steckplatz 24 |
| 21 | Bit 1 ... 0: Modul auf Steckplatz 25 : Bit 7 ... 6: Modul auf Steckplatz 28 |
| 22 | Bit 1 ... 0: Modul auf Steckplatz 29 : Bit 7 ... 6: Modul auf Steckplatz 32 |

Die Bits in Byte 15 ... 22 können folgende Werte annehmen:

- 00: Modul OK; gültige Daten*
- 01: Modulfehler; ungültige Daten (Modul defekt)*
- 10: falsches Modul; ungültige Daten*
- 11: kein Modul; ungültige Daten*

Kanalbezogene Diagnosedaten IM 353-1DP00

Die kanalbezogene Diagnose gibt Auskunft über Kanalfehler von Modulen und stellt eine Detaillierung der kennungsbezogenen Diagnose dar.

Die Länge der Kanalbezogenen Diagnosedaten ist fest auf 3Byte eingestellt.

Die Kanalbezogene Diagnose folgt nur dann nach der Modulbezogenen Diagnose, wenn einer der unter Byte 25 aufgeführten Fehler auftritt wie Bus-Init-Error, Bus-QVZ und Bus-Error.

Kanalbezogene Diagnose

| Byte | Bit 7 ... Bit 0 |
|------|---|
| 23 | Bit 5 ... 0: Kennungsnummer des Moduls, das die Diagnose liefert 000000 ... 011111: Kennungsnummer (Beispiel: Steckplatz 1 hat die Kennungsnummer 0 usw.) Bit 7 ... 6: Code für kanalbezogene Diagnose 10: Code 10 (fix) |
| 24 | Bit 5 ... 0: Nummer des Kanals, der die Diagnose liefert 000000 ... 111111: Kanalnummer Bit 7 ... 6: Ein-/Ausgabe 01: Eingabe 10: Ausgabe 11: Ein-/Ausgabe |
| 25 | Bit 4 ... 0: Fehler-Code 10101: <i>Bus-Init-Error</i> : Das Schreiben von Parametern an das entsprechende Modul ist fehlgeschlagen. 10110: <i>Bus-QVZ</i> : Es gab einen Lese-/Schreibfehler auf dem Rückwandbus 10111: <i>Bus-Error</i> : Die Anzahl der am Rückwandbus befindlichen Module stimmt nicht überein mit der Anzahl der projektierten Module. Bit 7 ... 5: Kanaltyp 101: Wort (fix) |

IM 353-1DP01 - DP-V1-Slave - Aufbau

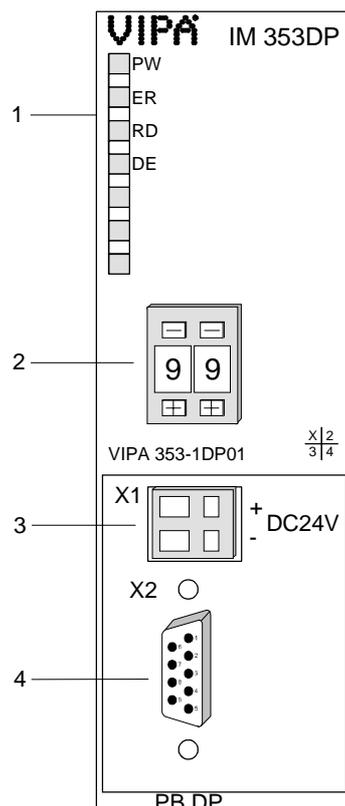
Eigenschaften

- 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 (max. 3,5A)
- Unterstützung aller Profibus-Datenraten

Einsatz als DP-V1-Slave

- 1 MSAC_C1-Verbindung (Read, Write) mit 244Byte Daten (4Byte DP-V1-Header + 240Byte Nutzdaten)
- 3 MSAC_C2-Verbindungen (Initiale, Read, Write, DataTransport, Abort) mit jeweils 244Byte Daten (4Byte DP-V1-Header + 240Byte Nutzdaten)

Frontansicht 353-1DP01



- [1] LED Statusanzeigen
- [2] Adresseinsteller

Folgende Komponenten befinden sich unter der Frontklappe

- [3] Anschluss für DC24V-Spannungsversorgung
- [4] RS485-Schnittstelle

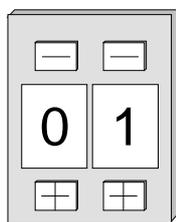
Komponenten

LEDs

Der Profibus-Slave besitzt 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 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 "Data exchange" wenn der Bus-Zyklus schneller als der Profibus-Zyklus läuft. Ist ausgeschaltet im Zustand "Data exchange" wenn der 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 | Grün | DE (Data exchange) zeigt an, dass eine Kommunikation mit dem Profibus stattfindet. |

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.



Hinweis

Bitte beachten Sie, dass zur eindeutigen Identifikation des Profibus-Slaves, die bei der Projektierung vorgegebene Profibusadresse und die am Adress-Schalter eingestellte Adresse identisch ist.

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.



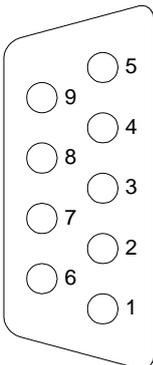
Achtung!

Bitte achten Sie auf richtige Polarität bei der Spannungsversorgung!

RS485-Schnittstelle

Über eine 9polige 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. |

IM 353-1DP01 - DP-V1-Slave - Projektierung

Allgemeines

Die Projektierung erfolgt als Hardware-Konfiguration in Ihrem Profibus-DP-Master Projektierool wie beispielsweise dem Siemens SIMATIC Manager. Hierbei ordnen Sie Ihrem DP-Master das entsprechende Profibus-DP-Slave-Modul zu. Eine direkte Zuordnung erfolgt über die Profibus-Adresse, die Sie am DP-Slave über den Adress-Schalter und in den DP-Slave-Eigenschaften einzustellen haben.

Durch Einbindung der entsprechenden GSD-Datei wird der IM 353-1DP01 DP-Slave als "VIPA 353-1DP01 (DP-V0 oder DP-V1)" aufgeführt unter:

Profibus-DP>Weitere Feldgeräte>I/O>VIPA_System_300V

DP-V0/DP-V1-Funktionalität über GSD-Datei

Von VIPA erhalten Sie den IM 353-1DP01 DP-Slave zusammen mit einem Datenträger. Auf dem Datenträger befinden sich unter anderem alle GSD-Dateien der Profibus-Module von VIPA.

Je nach installierter GSD-Datei werden folgende Module im Hardware-Katalog abgelegt:

| Modul | GSD-Datei |
|------------------------|--------------|
| VIPA 353-1DP01 (DP-V0) | VI0009AF.gsd |
| VIPA 353-1DP01 (DP-V1) | VI0109AF.gsd |

Installieren Sie die entsprechende GSD-Datei in Ihrem Projektierool. Nähere Hinweise zur Installation einer GSD-Datei finden Sie im Handbuch zu Ihrem Projektierool.

Projektierung

- Bauen Sie Ihr Profibus-System auf.
- Starten Sie Ihr Projektierool mit einem neuen Projekt.
- Projektieren Sie ein Master-System und legen Sie ein neues Profibus-Subnetz an.
- Zur Projektierung des IM 353-1DP01 entnehmen Sie je nach gewünschter Funktionalität den "VIPA 353-1DP01 (DPV0)" oder "VIPA 353-1DP01 (DPV1)" aus dem Hardware-Katalog und ziehen Sie diesen auf das DP-Master Subnetz.
- Geben Sie in den Eigenschaften des DP-Slave eine Profibus-Adresse zwischen 1 und 99 an und stellen Sie diese Adresse am Adress-Schalter ein.
- Parametrieren Sie den DP-Slave (siehe Parameter).
- Übertragen Sie Ihr Projekt in die SPS.



Hinweis

Bitte beachten Sie, dass Sie bei der Hardware-Konfiguration folgende Module auf den ersten drei Steckplätzen einbinden:

Config for Slot1
Config for Slot2
Config for Slot3

Bei Einsatz des Siemens SIMATIC Manager werden diese Module automatisch eingebunden.

**Parameterdaten
 IM 353-1DP01
 DP-V0**

Bei Verwendung des IM 353-1DP01 (DP-V0) haben Sie folgende Parameterdaten:

| Byte | Bit 7 ... Bit 0 | Default |
|----------|---|-------------------|
| 0 | Bit 2 ... 0: 0 (fix) Bit 3: 0 = WD-Timebase 10ms 1 = WD-Timebase 1ms Bit 4: 0 (fix) Bit 5: 0 = Publisher-Mode deaktiviert 1 = Publisher-Mode aktiviert | 00h ¹⁾ |
| 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 (fix) Bit 5: 0 = V0: Diagnosealarm deaktiviert 1 = V0: Diagnosealarm aktiviert Bit 6: 0 = V0: Prozessalarm deaktiviert 1 = V0: Prozessalarm aktiviert Bit 7: 0 (fix) | 70h |
| 8 | Bit 7 ... 0: 0 (fix) | 00h |
| 9 ... 12 | 00h (fix) | 00h |

¹⁾ Bei Einsatz des Siemens SIMATIC Manager wird dieser Wert automatisch eingestellt und kann nicht geändert werden.

**Parameterdaten
IM 353-1DP01
DP-V1**

Bei Verwendung des IM 353-1DP01 (DP-V1) haben Sie folgende Parameterdaten:

| Byte | Bit 7 ... Bit 0 | Default |
|----------|--|-------------------|
| 0 | Bit 2 ... 0: 0 (fix) Bit 3: 0 = WD-Timebase 10ms 1 = WD-Timebase 1ms Bit 4: 0 (fix) Bit 5: 0 = Publisher-Mode deaktiviert 1 = Publisher-Mode aktiviert Bit 6: 0 = Fail-Safe-Mode deaktiviert 1 = Fail-Safe-Mode aktiviert Bit 7: 0 = DP-V1-Betrieb sperren 1 = DP-V1-Betrieb freigeben | C0h ¹⁾ |
| 1 | Bit 0: Anlauf bei Sollausbau ungleich Istausbau (muss immer 0 sein, ansonsten erhalten Sie einen Parametrierfehler) Bit 3 ... 1: 0 (fix) Bit 4: 0 = V1: Herstellerspezifischer Alarm deaktiviert 1 = V1: Herstellerspezifischer Alarm aktiviert Bit 5: 0 = V1: Diagnosealarm deaktiviert 1 = V1: Diagnosealarm aktiviert Bit 6: 0 = V1: Prozessalarm deaktiviert 1 = V1: Prozessalarm aktiviert Bit 7: 0 (fix) | 70h |
| 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 7 ... 0: 0 (fix) | 00h |
| 9 ... 12 | 00h (fix) | 00h |

¹⁾ Bei Einsatz des Siemens SIMATIC Manager wird dieser Wert automatisch eingestellt und kann nicht geändert werden.

IM 353-1DP01 - DP-V1-Slave - DP-V1-Dienste

Übersicht

Für den Einsatz der DP-V1 Dienste ist darauf zu achten, dass Ihre CPU DP-V1-Kommunikation unterstützt. Näheres hierzu finden Sie in der Beschreibung Ihrer CPU. Hierbei stehen Ihnen folgende System-Funktions-Bausteine zur Verfügung:

- SFB 52 Datensatz aus einem DP-Slave lesen
- SFB 53 Datensatz in einen DP-Slave schreiben
- SFB 54 Alarm von einem DP-Slave empfangen

Es werden defaultmäßig 1 Klasse-1-Master- und bis zu maximal 3 Klasse-2-Master-Verbindung mit 244Byte Daten (4Byte DP-V1-Header plus 240Byte 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 Klasse-2-Master, der dann nur azyklisch mit dem Slave kommuniziert, benutzt werden und verfügt über einen eigenen Verbindungsaufbau.

Daten des DP-V1-Slave

Für den Zugriff aus dem Siemens SIMATIC Manager auf den DP-V1-Koppler ist als *ID* die *Diagnoseadresse* zu verwenden, die Sie in den Eigenschaften vorgeben können.

Durch Angabe folgender Datensatz-Nr. als *Index* haben Sie lesenden (R) bzw. schreibenden (W) Zugriff auf folgende Elemente des DP-Slave:

| Index | Zugriff | Beschreibung |
|-------|---------|--|
| A0h | R | Gerätename im ASCII-Code (VIPA 353-1DP01) |
| A1h | R | Hardware-Ausgabestand im ASCII-Code (V1.00) |
| A2h | R | Software-Ausgabestand im ASCII-Code (V1.00) |
| A3h | R | Serien-Nummer des Gerätes im ASCII-Code (z.B. 000347 = 30h, 30h, 30h, 33h, 34h, 37h) |
| A4h | R | Gerätekonfiguration (siehe Tabelle auf Folgeseite mit Modulkennung, die einem Modultyp zugeordnet ist) |
| D0h | R | Anzahl der gespeicherten Diagnosen |
| | W | Beliebiger Schreibbefehl löscht alle Diagnoseeinträge |
| D1h | R | Diagnoseeintrag der Reihe nach lesen |
| | W | Beliebiger Schreibbefehl speichert Diagnoseeinträge dauerhaft im FLASH-ROM |

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 Diagnoseeintrag |
| Zeitstempel | Doppelwort | interner Zeitstempel |
| Diagnose (max. 20Byte) | Byte | Diagnoseeintrag (Alarm), der intern abgelegt wird |

Gerätekonfiguration Mit dem Index A4h können Sie die Gerätekonfiguration des DP-Slaves ausgeben. Die Zuordnung der Typkennung zu einem Modultyp 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 14 / 2C | 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 |

Daten der Funktionsmodule

Für den Zugriff auf Funktionsmodule über den Siemens SIMATIC Manager ist als *ID* die *Moduladresse* zu verwenden, die Sie unter Eigenschaften vorgeben können.

Durch Angabe folgender Datensatz-Nr. als *Index* haben Sie lesenden (R) bzw. schreibenden (W) Zugriff auf folgende Elemente eines Funktionsmoduls:

| Index | Zugriff | Beschreibung |
|-------|---------|--------------------------|
| 00h | R | Diagnose - Datensatz 0 |
| | W | Modulparameter übergeben |
| 01h | R | Diagnose - Datensatz 1 |

IM 353-1DP01 - DP-V1-Slave - Diagnosefunktionen

| | |
|---|---|
| Übersicht | <p>Die umfangreichen Diagnosefunktionen unter Profibus-DP ermöglichen eine schnelle Fehlerlokalisierung. Die Diagnosedaten werden über den Bus übertragen und beim Master zusammengefasst.</p> <p>Als weitere Funktion wurde bei DP-V1 die gerätebezogene Diagnose verfeinert und in die Kategorien Alarme und Statusmeldungen aufgliedert.</p> <p>Zusätzlich werden bei dem DP-Slave die letzten 100 Alarm-Meldungen mit einem Zeitstempel in einem RAM gespeichert bzw. im Flash gesichert und können mit einer Software ausgewertet werden.</p> <p>Setzen Sie sich hierzu bitte mit der VIPA-Hotline in Verbindung!</p> <p>Sie können aber auch über DP-V1-Dienste auf die Diagnose zugreifen.</p> |
| Unterschied Diagnose DP-V0 zu DP-V1 | <p>Diagnose-Aufbau und -Verhalten sind unter DP-V0 und DP-V1 identisch. Der einzige Unterschied besteht darin, dass bei Einsatz in einem System 300 bei einem Prozessalarm unter DP-V0 der OB 82 und unter DP-V1 der OB 40 aufgerufen wird.</p> |
| Interne Diagnose Systemmeldungen | <p>Das System legt auch Diagnosemeldungen ab wie die Zustände "Ready" bzw. "DataExchange", die nicht an den Master weitergeleitet werden.</p> <p>Mit jedem Zustandswechsel zwischen "Ready" und "DataExchange" sichert der Profibus-Slave den Diagnose-RAM-Inhalt in einem Flash-ROM und schreibt diesen mit jedem Neustart in das RAM zurück.</p> |
| Diagnosedaten manuell sichern | <p>Über die kurzzeitige Einstellung von 00 am Adress-Schalter bzw. mit einem DP-V1-Schreibbefehl auf den Index D1h des DP-Slave können Sie die Diagnose-Daten während des "DataExchange" im Flash-ROM sichern.</p> |
| Diagnosemeldung bei Spannungsausfall | <p>Bei Spannungsausfall bzw. sinkender Spannung wird sofort ein Zeitstempel im EEPROM gespeichert. Sollte noch genügend Spannung vorhanden sein, erfolgt eine Diagnoseausgabe an den Master.</p> <p>Beim nächsten Neustart wird eine Unterspannung/Abschaltung-Diagnosemeldung aus dem Zeitstempel des EEPROMs generiert und im Diagnose-RAM abgelegt.</p> |

**Aufbau der
353-1DP01
Diagnosedaten**

Die Diagnose-Meldungen, die vom Profibus-Slave erzeugt werden, haben je nach Parametrierung eine maximale 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 | wird nur bei Transfer über Profibus an den Master vorangestellt | über Parametrierung sperr- oder freischaltbar |
| x ... x+4 | Kennungsbezogene Diagnose | | |
| x ... x+11 | Modulstatus | | |
| max. 9·(x ... x+2) | Kanalbezogene Diagnose | | |
| x ... x+19 | Alarm | Diagnose, die intern abgelegt wird | |

**Norm-Diagnose-
daten**

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

Die Slave-Normdiagnosedaten haben folgenden Aufbau:

Norm-Diagnose

| Byte | Bit 7 ... Bit 0 |
|------|---|
| 0 | Bit 0: 0 (fix) 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: 0 (fix) Bit 6: Falsche Parametrierung Bit 7: 0 (fix) |
| 1 | Bit 0: Slave muss neu parametriert werden Bit 1: Statistische Diagnose Bit 2: 1 (fix) Bit 3: Ansprechüberwachung aktiv Bit 4: "FREEZE"-Kommando erhalten Bit 5: "SYNC"-Kommando erhalten Bit 6: reserviert Bit 7: 0 (fix) |
| 2 | Bit 6 ... 0: 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 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*.

Die kennungsbezogene Diagnose kann über die Parametrierung aktiviert werden und hat folgenden Aufbau:

Kennungsbezogene Diagnose

| Byte | Bit 7 ... Bit 0 |
|------|---|
| X | Bit 5 ... 0: 000101 (fix) Länge kennungsbezogene Diagnose Bit 7 ... 6: 01 (fix) Code für kennungsbezogenen Diagnose |
| X+1 | Die Bits der Module je Steckplatz werden gesetzt, wenn: <ul style="list-style-type: none"> - 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 |

Modulstatus

Mit dem Modulstatus erhalten Sie nähere Informationen zum Fehler, der in einem Modul aufgetreten ist.

Der Modulstatus kann über die Parametrierung aktiviert werden und hat folgenden Aufbau:

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 |

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 |

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 wird, sofern in der Parametrierung aktiviert, immer als letzter Teil an das Diagnosetelegramm angehängt.

Aufbau

Je nach Alarmtyp hat der Alarmteil folgenden Aufbau:

| Byte | Element | Beschreibung |
|------------|---------------|--|
| x...x+3 | Alarmstatus | Beinhaltet Informationen über den Alarmtyp |
| x+4...x+19 | Diagnosealarm | Die 16Byte entsprechen dem Datensatz 1 der CPU-Diagnose |
| x+4...x+7 | Prozessalarm | Die 4Byte 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: 00 (fix) 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 _{kommend} 10: Diagnosealarm _{gehend} 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 parametriert Bit 7: Parametrierfehler |
| x+5 | Bit 3 ... 0: Modulklasse 1111: Digitalmodul 0101: Analogmodul 1000: FM 1100: CP Bit 4: Kanalinformation vorhanden Bit 5: Anwenderinformation vorhanden Bit 6: 0 (fix) Bit 7: 0 (fix) |
| 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 (fix) |
| 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.

Aufbaurichtlinien

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

| | | |
|---------------------|---|-------|
| 9,6 ... 187,5kBaude | → | 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.
 - Der Bus ist an beiden Enden abzuschließen.
 - Master und Slaves sind beliebig mischbar.
- Aufbau und Einbindung in Profibus**
- Bauen Sie Ihr Profibus-System mit den entsprechenden Modulen auf.
 - Übertragen Sie die mitgelieferte GSD-Datei in Ihr Projektiersystem und projektieren Sie Ihr System, geben Sie hierbei eine gültige Profibus-adresse an.
 - 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.

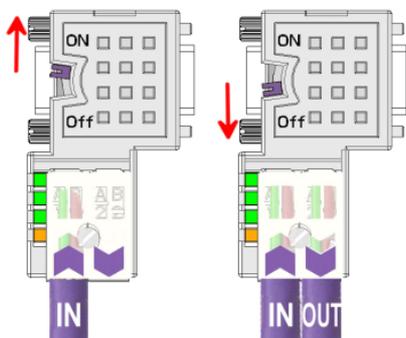


Hinweis!

Die Profibus-Leitung muss mit ihrem Wellenwiderstand abgeschlossen werden. Bitte beachten Sie, dass Sie bei dem jeweilig letzten Teilnehmer den Bus durch Zuschalten eines Abschlusswiderstands abschließen.

Leitungsabschluss mit "EasyConn"

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



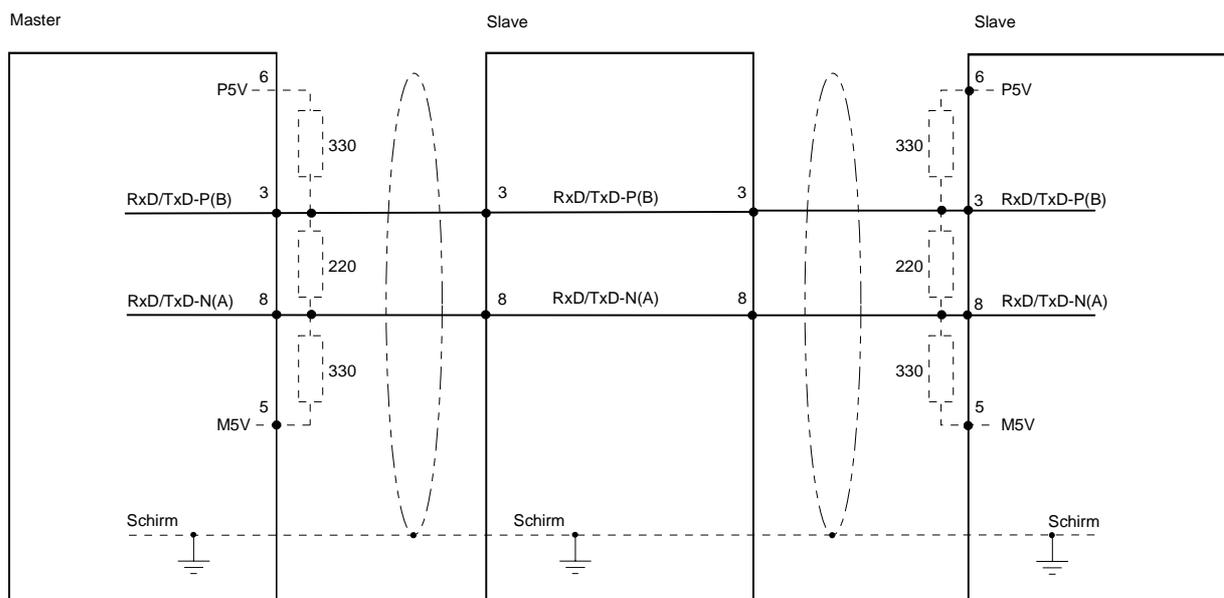
Achtung!

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.

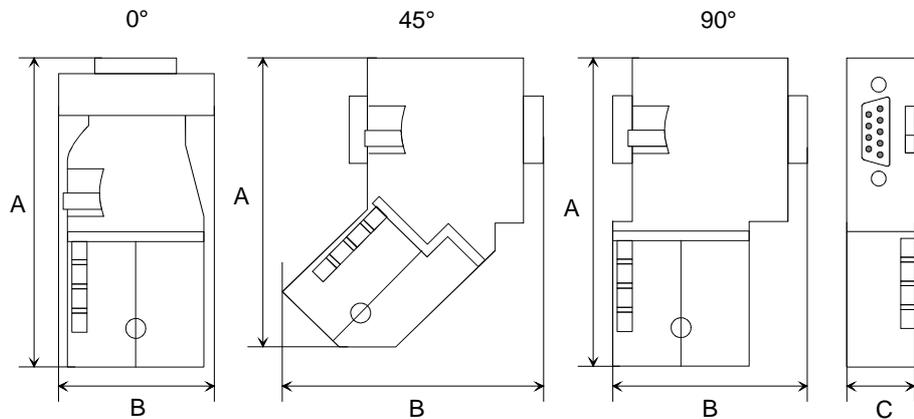
In der nachfolgenden Abbildung sind die Abschlusswiderstände der jeweiligen Anfangs- und Endstation stilisiert dargestellt.



"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 Busanschlussstecker 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:



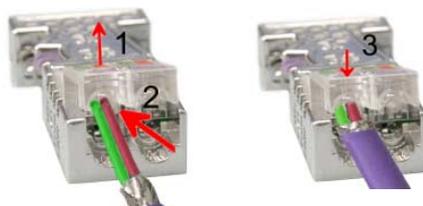
Hinweis!

Zum Anschluss des EasyConn-Steckers verwenden Sie bitte die Standard Profibus-Leitung Typ A (EN50170). Ab Ausgabestand 5 können auch hochflexible Bus-Kabel verwendet werden: Lapp Kabel Best.-Nr.: 2170222, 2170822, 2170322. 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

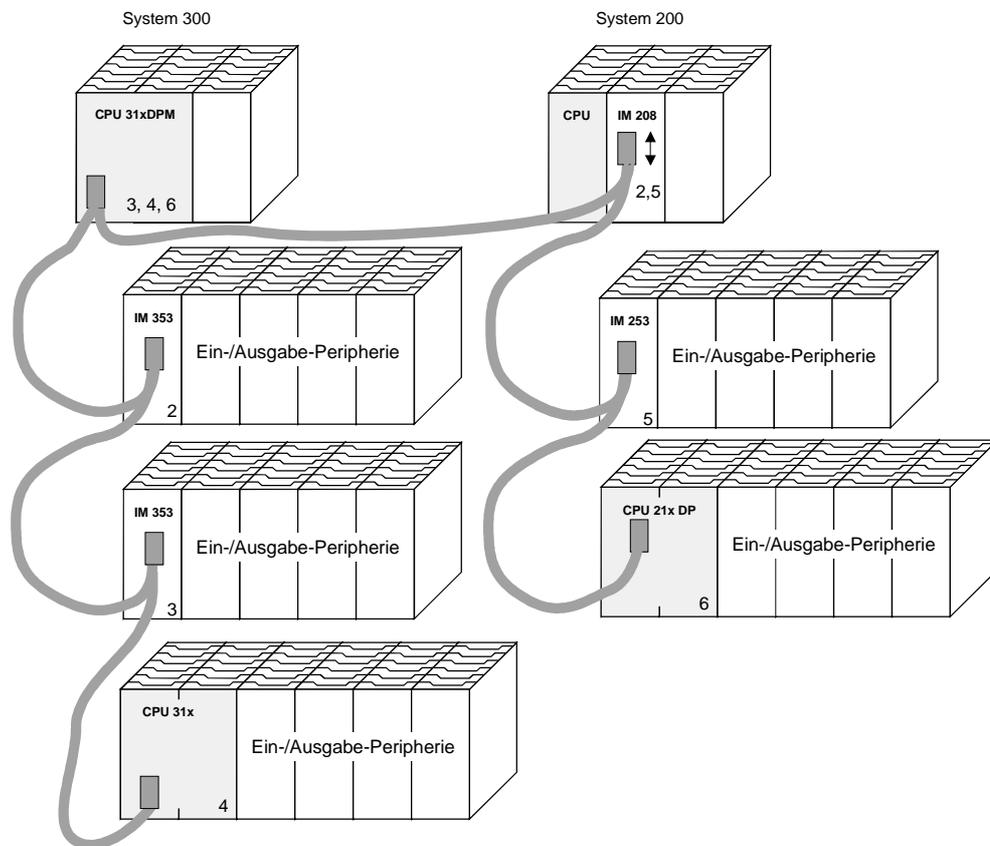


- Lösen Sie die Schraube
- Klappen Sie den Kontaktdeckel auf
- Stecken Sie beide Adern in die dafür vorgesehenen Öffnungen (Farbzuordnung wie unten beachten!)
- Bitte beachten Sie, dass zwischen Schirm und Datenleitungen kein Kurzschluss entsteht!
- Schließen Sie den Kontaktdeckel
- Ziehen Sie die Schraube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

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

Multi Master System

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



Inbetriebnahme

Übersicht

- Bauen Sie Ihr Profibus-System auf.
- Projektieren Sie Ihr Mastersystem.
- Stellen Sie eine gültige Profibusadresse ein.
- Transferieren Sie Ihr Projekt in Ihren Master.
- Verbinden Sie Ihre Master- und Slave-Module mit Ihrem Profibus.
- Schalten Sie die Spannungsversorgung ein.

Aufbau

Bauen Sie Ihr Profibus-System mit den gewünschten Peripherie-Modulen auf. Jeder Profibus-Slave-Koppler besitzt ein eingebautes Netzteil. Es 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 3,5A versorgen kann.

Profibus und Rückwandbus sind galvanisch voneinander getrennt.

Projektierung im Mastersystem

Projektieren Sie Ihre Profibus-Master in Ihrem Master-System. Sie können WinNCS von VIPA einsetzen oder den Hardware-Konfigurator von Siemens.

Adressierung

Stellen Sie an den Profibus-Slave-Modulen die entsprechende Profibus-Adresse ein, die Sie bei der Projektierung vergeben haben.

Projekt transferieren

Abhängig vom eingesetzten Master gibt es verschiedene Möglichkeiten Ihr Projekt in Ihren DP-Master zu übertragen.

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!



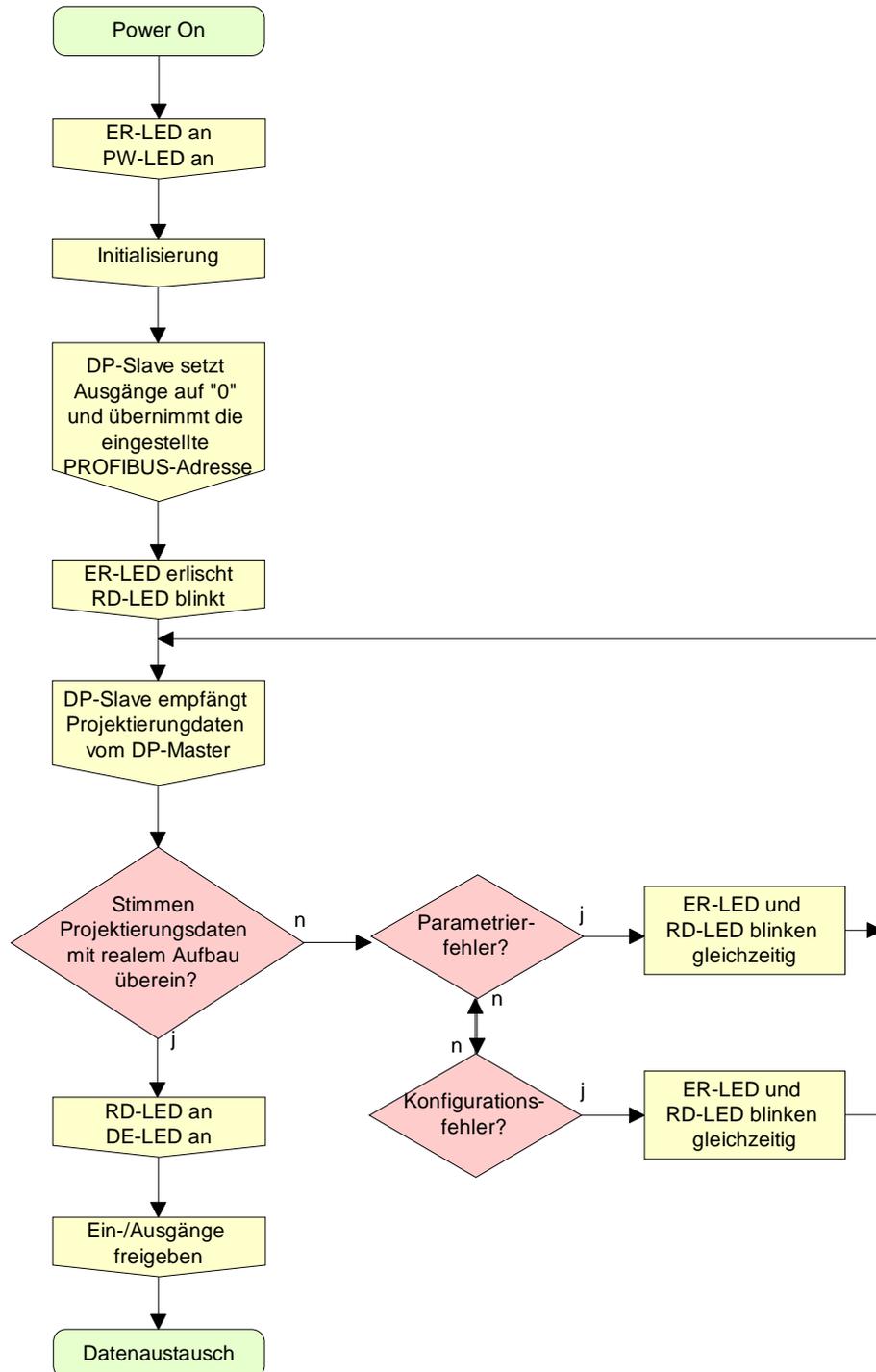
Hinweis!

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

**Anlaufverhalten
DP-Slave**

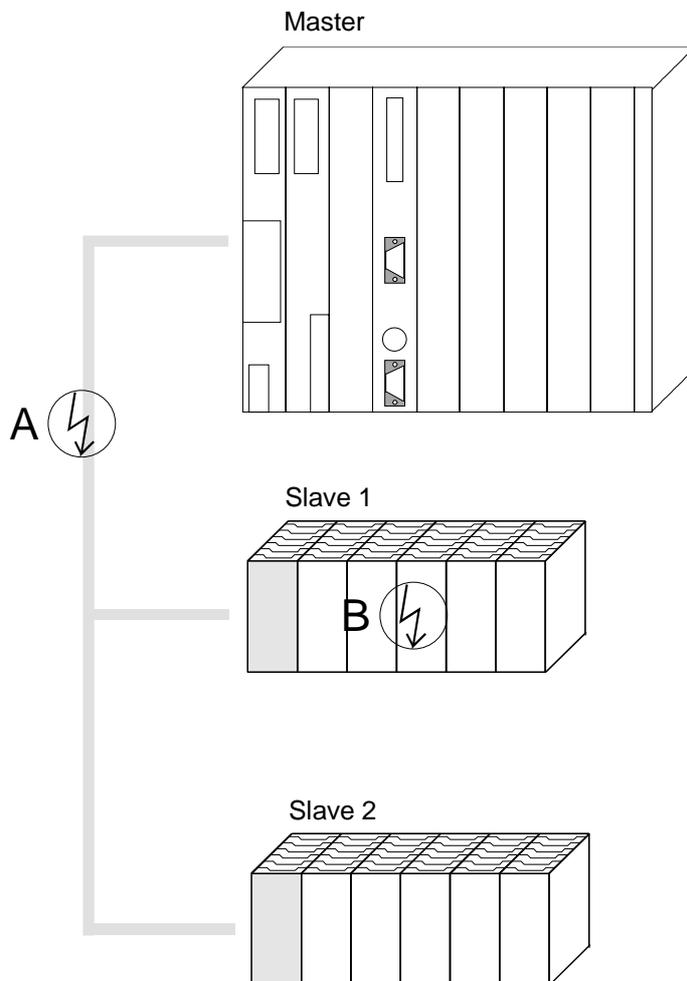
Nach dem Einschalten durchläuft der DP-Slave einen Selbsttest. Hierbei überprüft er seine internen Funktionen und die Kommunikation über den Rückwandbus. Nach fehlerfreiem Hochlauf geht der Buskoppler in den Zustand "READY" über. Im Zustand READY erhält der DP-Slave vom DP-Master seine Parameter und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



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 |

Technische Daten

IM 353DP

| Elektrische Daten | VIPA 353-1DP00 (DP-V0) | VIPA 353-1DP01 (DP-V0/V1) |
|-----------------------------------|---|------------------------------|
| Spannungsversorgung | DC 24V, über Front von ext. Netzteil | |
| Stromaufnahme | max. 1A | |
| Stromversorgung int. Bus | 3,5A | |
| Potenzialtrennung | ≥ AC 500V | |
| Statusanzeige | über LEDs auf der Frontseite | |
| Anschlüsse/Schnittstellen | RS485: 9polige SubD-Buchse | Profibus-Ankopplung |
| Profibus Schnittstelle | | |
| Ankopplung | RS485: 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 | 9,6kBaud bis 12MBaud (automatische Einstellung) | |
| Gesamtlänge | ohne Repeater 100m bei 12MBaud; mit Repeater bis 1000m | |
| max. Teilnehmeranzahl | 32 Stationen in jedem Segment ohne Repeater. Mit Repeater erweiterbar auf 126. | |
| Diagnosefunktionen | | |
| Standard-Diagnose | Speicherung der letzten 100 Diagnosen im Flash-ROM. | |
| Erweiterte Diagnosemöglichkeit | nein | möglich |
| Kombination mit Peripheriemodulen | | |
| max. Modulanzahl | 32 einzeilig | |
| max. digital | 32 | |
| max. analog | 16 | |
| max. Eingänge | 152Byte | 244Byte |
| max. Ausgänge | 152Byte | 244Byte |
| Maße und Gewicht | | |
| Abmessungen (BxHxT) in mm | 40x125x120 | |
| Gewicht | 170g | |

Teil 4 CANopen

Überblick

Inhalt dieses Kapitels ist die Beschreibung des IM 353CAN CANopen-Slave Moduls von VIPA. Nach einer Systemvorstellung folgt die Beschreibung des Moduls.

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, NMT und den Technischen Daten endet das Kapitel.

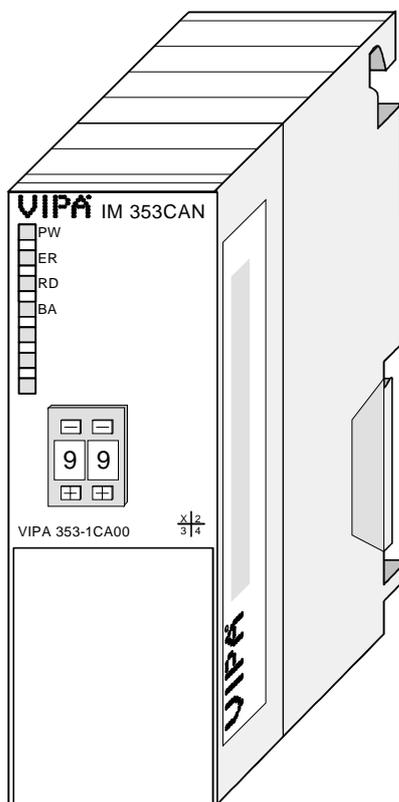
Inhalt

| Thema | Seite |
|--|------------|
| Teil 4 CANopen | 4-1 |
| Systemübersicht..... | 4-2 |
| Grundlagen | 4-3 |
| IM 353CAN - CANopen-Slave - Aufbau..... | 4-5 |
| IM 353CAN - CANopen-Slave - Schnelleinstieg | 4-9 |
| IM 353CAN - CANopen-Slave - Baudrate und Modul-ID | 4-13 |
| IM 353CAN - CANopen-Slave - Telegrammaufbau | 4-14 |
| IM 353CAN - CANopen-Slave - PDO | 4-16 |
| IM 353CAN - CANopen-Slave - SDO | 4-20 |
| IM 353CAN - CANopen-Slave - Objekt-Verzeichnis | 4-22 |
| IM 353CAN - CANopen-Slave - Emergency Object..... | 4-63 |
| IM 353CAN - CANopen-Slave - NMT - Netzwerk Management..... | 4-64 |
| Technische Daten | 4-66 |

Systemübersicht

CANopen-Slave IM 353CAN

Folgende CAN-Bus-Koppler sind von VIPA verfügbar:



Bestelldaten

| Typ | Bestellnummer | Beschreibung | Seite |
|-----------|----------------|-----------------------|-------|
| IM 353CAN | VIPA 353-1CA00 | CAN-Bus CANopen-Slave | 4-5 |

Grundlagen

Allgemeines

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

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

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

CANopen

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

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

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

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

**Übertragungs-
medium**

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

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

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

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

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

Alle Teilnehmer im Netz kommunizieren mit der gleichen Baudrate.

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

**Buszugriffs-
verfahren**

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

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

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

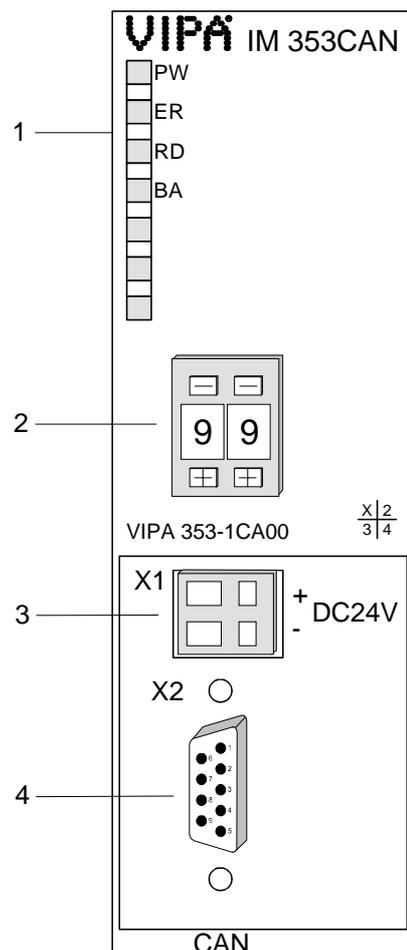
Die Buszugriffssteuerung bei CAN geschieht mit Hilfe der zerstörungsfreien, bitweisen Arbitrierung. Hierbei bedeutet zerstörungsfrei, dass der Gewinner der Arbitrierung sein Telegramm nicht erneut senden muss. Beim gleichzeitigen Mehrfachzugriff von Teilnehmern auf den Bus wird automatisch der wichtigste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

IM 353CAN - CANopen-Slave - Aufbau

Eigenschaften

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

Frontansicht 353-1CA00



- [1] LED Statusanzeigen
- [2] Adress- bzw. Baudraten-einsteller (Codiertaster)
- [3] Anschluss für externe 24V Spannungsversorgung
- [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 | Gelb | Signalisiert eine anliegende Betriebsspannung. |
| ER | Rot | Leuchtet bei Fehler in der Rückwandbusübertragung. |
| RD | Grün | Blinkt mit 1Hz bei positivem Selbsttest und erfolgreicher Initialisierung Leuchtet bei Datenübertragung über den V-Bus. |
| BA | Gelb | Aus bei positivem Selbsttest und erfolgreicher Initialisierung. 1Hz Blinken im Zustand "Pre-Operational". Leuchtet im Zustand "Operational". 10Hz Blinken im Zustand "Prepared". |

Statusanzeige durch LED-Kombination

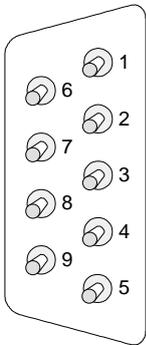
Durch Kombination der LEDs werden verschiedene Zustände angezeigt:

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

9poliger SubD-Stecker

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

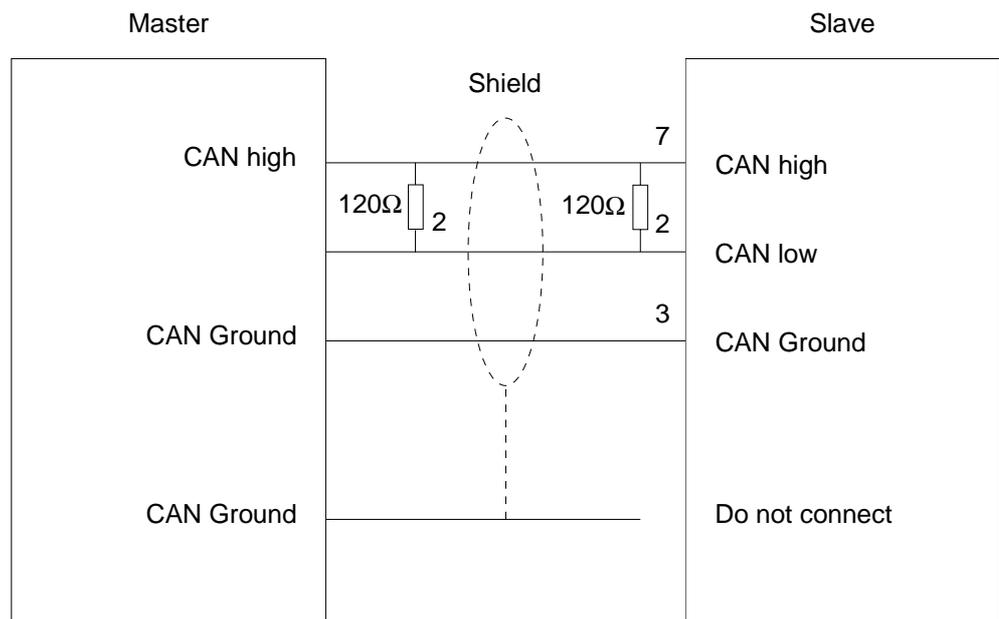
Die Anschlussbelegung dieser Schnittstelle zeigt folgende Abbildung:



| Pin | Belegung |
|-----|----------------------|
| 1 | n.c. |
| 2 | CAN low |
| 3 | CAN Ground |
| 4 | n.c. |
| 5 | n.c. |
| 6 | optional Ground |
| 7 | CAN high |
| 8 | n.c. |
| 9 | optional pos. Supply |

Verkabelung unter CAN-Bus

CAN-Bus verwendet als Übertragungsmedium eine abgeschirmte Dreidrahtleitung.



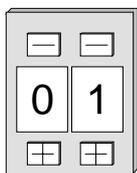
Leitungsabschluss

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!

**Adress-Schalter
für Baudrate und
Modul-ID**

Über diesen Adress-Schalter 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.

**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. 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!

IM 353CAN - 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 300V 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 300V 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

| | | | | | |
|---------------------------|------------------|------------------|------------------|-----|------------------|
| Identifizier 11Bit | DI 0 8Bit | DI 1 8Bit | DI 2 8Bit | ... | DI 7 8Bit |
|---------------------------|------------------|------------------|------------------|-----|------------------|

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

| | | | | | |
|---------------------------|------------------|------------------|------------------|-----|-----------------|
| Identifizier 11Bit | DO 0 8Bit | DO 1 8Bit | DO 2 8Bit | ... | DO 7 Bit |
|---------------------------|------------------|------------------|------------------|-----|-----------------|

Analoge Ein-/Ausgänge

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

| | | | | |
|---------------------------|-------------------|-------------------|-------------------|-------------------|
| Identifizier 11Bit | AI 0 1Wort | AI 1 1Wort | AI 2 1Wort | AI 3 1Wort |
|---------------------------|-------------------|-------------------|-------------------|-------------------|

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

| | | | | |
|---------------------------|-------------------|-------------------|-------------------|-------------------|
| Identifizier 11Bit | AI 0 1Wort | AI 1 1Wort | AI 2 1Wort | AI 3 1Wort |
|---------------------------|-------------------|-------------------|-------------------|-------------------|

Node Guarding

Da das System 300V 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

| |
|---------------------------|
| Identifizier 11Bit |
|---------------------------|

Der System 300V-Knoten antwortet mit einem Telegramm, das ein Statusbyte enthält:

Identifizier 0x700+Knotenadresse + Statusbyte

| | |
|---------------------------|--------------------|
| Identifizier 11Bit | Status 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 300V 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

| | |
|---------------------------|--------------------|
| Identifizier 11Bit | Status 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

| | | | | | | | | |
|---------------------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Identifizier 11Bit | EC0 | EC1 | Ereg | Inf0 | Inf1 | Inf2 | Inf3 | Inf4 |
|---------------------------|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|

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

Fortsetzung ...

... Fortsetzung Emergency Objekt

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



Hinweis!

Mit den beschriebenen Telegrammen sind Sie nun in der Lage, das System 300V zu starten und zu stoppen, Eingänge zu lesen, Ausgänge zu schreiben und die Module zu überwachen.

Nachfolgend sind alle Funktionen nochmals detailliert beschrieben.

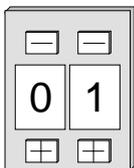
IM 353CAN - CANopen-Slave - Baudrate und Modul-ID

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 Adress-Schalter



- Stellen Sie am Adress-Schalter 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 Adress-Schalter 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 353CAN - 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 353 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

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 Eingabe 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 |

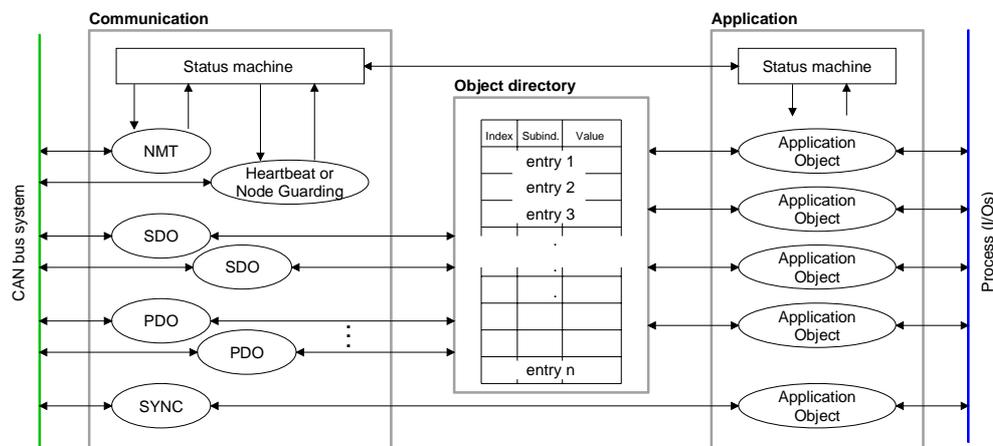


Hinweis!

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 353CAN - 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 353CAN 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.

**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 300V 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

| | |
|--------------------------------|---|
| PDO-Linking | <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> |
| PDO Kommunikationsarten | <p>CANopen bietet folgende Möglichkeiten der Prozessdatenübertragung:</p> <ul style="list-style-type: none">• Ereignisgesteuert• Gepollt• Synchronisiert |
| Ereignisgesteuert | <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> |
| Gepollt | <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> |
| Synchronisiert | <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 353CAN - 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-Identifier 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 353CAN - 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 – 0x1029 | allgemeine kommunikationsspezifische Parameter (z.B. der Gerätename) |
| 0x1400 – 0x1409 | Kommunikationsparameter (z.B. Identifier) der Receive-PDOs |
| 0x1600 – 0x1609 | 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 – 0x1809 | Kommunikations- und Mappingparameter der |
| 0x1A00 – 0x1A09 | 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 |
| 0x1007 | | Synchronous windows length |
| 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 | Consumer heartbeat time |
| 0x1017 | X | Producer heartbeat time |
| 0x1018 | | Identity object |
| 0x1027 | | Module list |
| 0x1029 | | Error behavior |
| 0x1400 - 0x1409 | X | Communication parameter RxPDOs (master to slave) |
| 0x1600 - 0x1609 | X | Mapping parameter RxPDOs |
| 0x1800 - 0x1809 | X | Communication parameter TxPDOs (slave to master) |
| 0x1A00 - 0x1A09 | X | Mapping parameter TxPDOs |
| 0x2001 | | CAN baudrate |
| 0x2100 | | Kill EEPROM |
| 0x2101 | | SJA1000 message filter |
| 0x2400 | X | PDO control |
| 0x3001 - 0x3010 | X | Module parameterization |
| 0x3401 | X | Module parameterization |
| 0x6000 | | Digital input 8 bit (see DS 401) |
| 0x6002 | X | Polarity digital input 8 bit (see DS 401) |
| 0x6100 | | Digital input 16 bit (see DS 401) |
| 0x6102 | | Polarity digital input 16 bit (v DS 401) |
| 0x6120 | | Digital input 32 bit (see DS 401) |
| 0x6122 | | Polarity digital input 32 bit (see DS 401) |
| 0x6200 | | Digital output 8 bit (see DS 401) |
| 0x6202 | X | Polarity digital output 8 bit (see DS 401) |
| 0x6206 | X | Fault mode digital output 8 bit (see DS 401) |
| 0x6207 | X | Fault state digital output 8 bit (see DS 401) |
| 0x6300 | | Digital output 16 bit (see DS 401) |

Fortsetzung ...

... Fortsetzung
Objektverzeichnis
Übersicht

| Index | | Content of Object |
|--------|---|---|
| 0x6302 | | Polarity digital output 16 bit (see DS 401) |
| 0x6306 | | Fault mode digital output 16 bit (see DS 401) |
| 0x6307 | | Fault state digital output 16 bit (see DS 401) |
| 0x6320 | | Digital output 32 bit (see DS 401) |
| 0x6322 | | Polarity digital output 32 bit (see DS 401) |
| 0x6326 | | Fault mode digital output 32 bit (see DS 401) |
| 0x6327 | | Fault state digital output 32 bit (see DS 401) |
| 0x6401 | | Analog input (see DS 401) |
| 0x6411 | | Analog output (see DS 401) |
| 0x6421 | X | Analog input interrupt trigger (see DS 401) |
| 0x6422 | | Analog input interrupt source (see DS 401) |
| 0x6423 | X | Analog input interrupt enable (see DS 401) |
| 0x6424 | X | Analog input interrupt upper limit (see DS 401) |
| 0x6425 | X | Analog input interrupt lower limit (see DS 401) |
| 0x6426 | X | Analog input interrupt delta limit (see DS 401) |
| 0x6443 | X | Fault mode analog output (see DS 401) |
| 0x6444 | X | Fault state analog output (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 |
|--------------------------------------|-----------------------|
| Additional information Device | profile number |
| 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 ... A maximum of 254 error states |
| | 1 | Actual error | Unsigned32 | ro | N | | |
| | ... 254 | ... | ... Unsigned32 | ... ro | ... N | ... | |

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

| | |
|------------------------|------------|
| MSB | LSB |
| Additional information | Error code |

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

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

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

Durch einen Reset oder Power Cycle wird der Fehlerspeicher gelöscht.

Number of PDOs

| Index | Sub index | Name | 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 μ 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 μ 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 353 1CA00 = VIPA CANopen-Slave IM 353-1CA00

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 | | Hardware version number of bus coupler |

VIPA IM 353 1CA00 = 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 353 1CA00 = 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.

Node 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 parameter

| 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 parameter

| 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 353-1CA00: 0x3531CA00

Module list

| 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 |

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 (1) oder nicht (0), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1).

Der Subindex 2 enthält die Übertragungsart.

**Communication
parameter RxPDO2**

| Index | Sub-index | Name | 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 parameter 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 parameter 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 parameter 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 (1) oder nicht (0), Bit 30 teilt mit, ob ein RTR-Zugriff auf dieses PDO zulässig ist (0) oder nicht (1). Der Subindex 2 enthält die Übertragungsart, Subindex 3 die Wiederholungsverzögerung zwischen zwei gleichen PDOs. Wenn ein "Event Timer" mit einem Wert ungleich 0 existiert, wird nach Ablauf dieses Timers das PDO übertragen.

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

Communication parameter TxPDO2

| Index | Sub-index | Name | 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. COB-ID TxPDO3 |
| | 1 | COB-ID | Unsigned32 | rw | N | 0x80000380 + 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 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. COB-ID TxPDO4 |
| | 1 | COB-ID | Unsigned32 | rw | N | 0x80000480 + 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 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. COB-ID TxPDO5 |
| | 1 | COB-ID | Unsigned32 | rw | N | 0x80000680 + 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 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. |
| | 1 | COB-ID | Unsigned32 | rw | N | 0x800004C0 + NODE_ID | COB-ID TxPDO9 |
| | 2 | Transmission type | Unsigned8 | rw | N | 0xFF | Transmission type of the PDO |
| | 3 | Inhibit time | Unsigned16 | rw | N | 0x0000 | Repetition delay [value x 100 µs] |
| | 5 | Event time | Unsigned16 | rw | N | 0x0000 | Event timer [value x 1 ms] |

Communication parameter TxPDO10

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

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

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:

$$0(ID.10 \text{ bis } ID.3) \equiv (AC.7 \text{ bis } AC.0) \vee (AM.7 \text{ bis } AM.0) \equiv 11111111$$

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.

Diese Funktion eignet sich ausschließlich für Master, die kein Heartbeat bzw. Node Guarding unterstützen.

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 parametrisiert werden.

Beispiel 2**Set FM250 to Counter Mode 0x08 and 0x0B****Read default configuration**

Read SubIndex 0 M2S: 0x40 0x02 0x30 0x00 0x00 0x00 0x00 0x00
S2M: 0x4F 0x02 0x30 0x00 0x04 0x00 0x00 0x00
Read SubIndex 1 M2S: 0x40 0x02 0x30 0x01 0x00 0x00 0x00 0x00
S2M: 0x43 0x02 0x30 0x01 0x00 0x00 0x00 0x00

Write new configuration

Write SubIndex 1 M2S: 0x23 0x02 0x30 0x01 0x08 0x0B 0x00 0x00
S2M: 0x60 0x02 0x30 0x01 0x00 0x00 0x00 0x00

Read new configuration

Read SubIndex 0 M2S: 0x40 0x02 0x30 0x00 0x00 0x00 0x00 0x00
S2M: 0x4F 0x02 0x30 0x00 0x04 0x00 0x00 0x00
Read SubIndex 1 M2S: 0x40 0x02 0x30 0x01 0x00 0x00 0x00 0x00
S2M: 0x43 0x02 0x30 0x01 0x08 0x0B 0x00 0x00

Module parameterization

| Index | Sub-index | Name | Typ | Attr. | Map. | Default value | Meaning |
|--------|-----------|--------------------|------------|-------|------|------------------------------------|-------------------|
| 0x3401 | 0x00 | Number of Elements | Unsigned8 | ro | N | depending on the components fitted | Number of Entries |
| | 0x01 | 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.

Digital input 8 bit

| 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 | 72nd input block | Unsigned8 | ro | Y | | |

**Polarity digital
input 8 bit**

| 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

Digital input 16 bit

| 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 |

Polarity digital input 16 bit

| 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

Digital input 32 bit

| 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 |

**Polarity digital
input 32 bit**

| 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

Digital output 8 bit

| 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 |

**Polarity digital
output 8 bit**

| 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

**Fault mode digital
output 8 bit**

| 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 Ausgabekanal einen bestimmten Wert annimmt, den Sie im Objekt 0x6207 vorgeben.

1 = den Wert in Objekt 0x6207 übernehmen

0 = Ausgabewert im Fehlerfall fixieren

Fault state digital output 8 bit

| 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.

Digital output 16 bit

| 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 |

Polarity digital output 16 bit

| 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

Fault mode digital output 16 bit

| 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

Fault state digital output 16 bit

| 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.

Digital output 32 bit

| 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 |

Polarity digital output 32 bit

| 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

Fault mode digital output 32 bit

| 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

Fault state digital output 32 bit

| 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 input

| 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 output

| 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

| 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

Analog input interrupt enable

| 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)

Analog input interrupt upper limit

| 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.

Analog input interrupt lower limit

| 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.

Analog input interrupt delta limit

| 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).

Fault mode analog output

| 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

Fault state analog output

| 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 353CAN - 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 353CAN - CANopen-Slave - NMT - Netzwerk Management

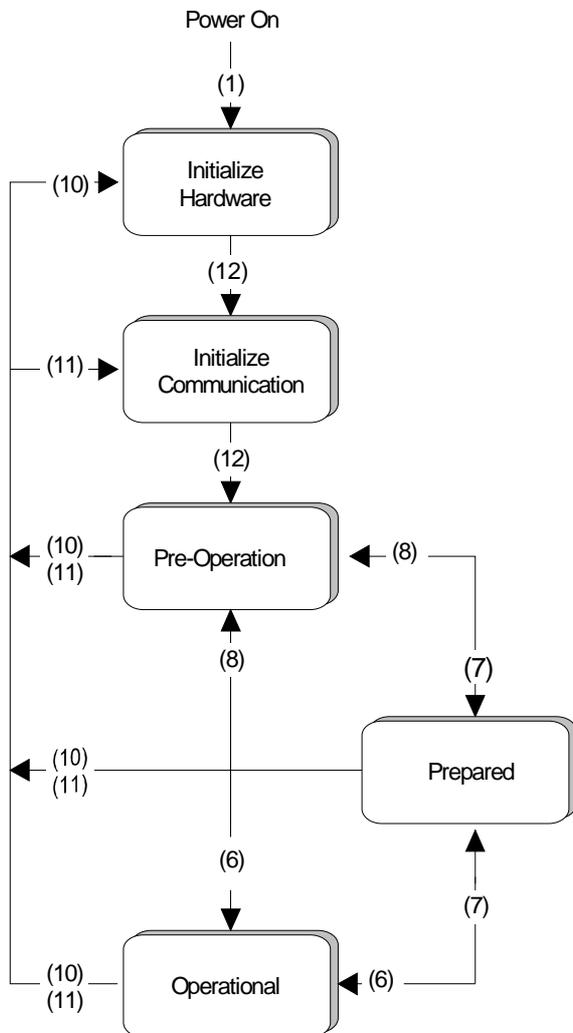
Das Netzwerkmanagement (NMT) spezifiziert globale Dienste für Netzwerküberwachung und -Management. Dazu gehört neben dem An- und Abmelden einzelner Teilnehmer auch die Überwachung der Teilnehmer während des Betriebs- und die Behandlung von Ausnahmezuständen.

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-Koppler IM 353CAN

| | |
|-----------------------------------|--|
| Elektrische Daten | VIPA 353-1CA00 |
| Spannungsversorgung | DC 24V (20,4 ... 28,8V) über Front von ext. Netzteil |
| Stromaufnahme | max. 0,7A |
| Ausgangsstrom Rückwandbus | max. 3,5A |
| Potenzialtrennung | ≥ AC 500V |
| Statusanzeige | über LEDs auf der Frontseite |
| Anschlüsse/Schnittstellen | 9poliger SubD (Stecker) CAN-Bus-Ankopplung |
| CAN-Bus Schnittstelle | |
| Ankopplung | 9poliger SubD-Stecker |
| Netzwerk Topologie | Linearer Bus, aktiver Busabschluss an einem Ende, Stichleitungen sind möglich. |
| Medium | Abgeschirmtes dreiadriges Kabel, Schirmung darf, abhängig von Umgebungsbedingungen, entfallen. |
| Übertragungsrate | 10kBaud bis 1MBaud |
| max. Gesamtlänge | ohne Repeater 1000m bei 50kBaud |
| digitale Ein-/Ausgänge | Je Koppler maximal 32 E/A-Module frei kombinierbar. |
| Kombination mit Peripheriemodulen | |
| max. Modulanzahl | 32 (abhängig von der Stromaufnahme) |
| max. Eingänge/Ausgänge | je 80Byte (80Byte = 10 PDOs à 8Byte) |
| Maße und Gewicht | |
| Abmessungen (BxHxT) in mm | 25,4x76x78 |
| Gewicht | 80g |

Anhang

A Index

- 3
- 353-1CA00 4-5
- Adress-Schalter 4-8, 4-13
 - Aufbau 4-5
 - Baudrate 4-13
 - CAN-Identifizier 4-9
 - Diagnosefunktionen 4-63
 - Fehlermeldungen 4-11, 4-21
 - Funktionscodes 4-15
 - Heartbeat 4-65
 - Identifizier 4-17
 - Kommunikationsarten 4-18
 - Komponenten 4-6
 - LEDs 4-6
 - Modul-ID 4-13
 - Modultypen 4-32
 - NMT 4-64
 - Node Guarding 4-65
 - Objektverzeichnis 4-22
 - PDO 4-16
 - Linking 4-18
 - RS485-Schnittstelle 4-7
 - Schnelleinstieg 4-9
 - SDO 4-20
 - Spannungsversorgung 4-8
 - Technische Daten 4-66
 - Telegrammaufbau 4-14
 - Übertragungsart 4-19
- 353-1DP00 3-12
- Adress-Schalter 3-13
 - Aufbau 3-12
 - Diagnose 3-16
 - Komponenten 3-13
 - LEDs 3-13
 - Projektierung 3-15
 - RS485-Schnittstelle 3-14
 - Spannungsversorgung 3-14
 - Technische Daten 3-46
- 353-1DP01 3-22
- Adress-Schalter 3-23
 - Alarmer 3-35
 - Aufbau 3-22
 - Diagnose 3-30
 - DP-V1-Dienste 3-28
 - Komponenten 3-23
 - LEDs 3-23
 - Projektierung 3-25
 - RS485-Schnittstelle 3-24
 - Spannungsversorgung 3-24
 - Technische Daten 3-46
- B**
- Bus-Zyklus 3-7
- C**
- CANopen 4-1
- 353-1CA00 4-5
 - Buszugriff 4-4
 - Emergency Objekt 4-11
 - Fehlermeldungen 4-11
 - Grundlagen 4-3
 - Heartbeat 4-11
 - Node Guarding 4-10
 - Systemübersicht 4-2
 - Übertragungsmedium 4-4
- COB-ID 4-17
- D**
- DP-Zyklus 3-7
- E**
- EasyConn 3-40
- Emergency Objekt 4-11
- F**
- Fehlermeldungen
- 353-1CA00 4-11, 4-21
 - 353-1DP00 3-16
 - 353-1DP01 3-30
- G**
- Grundlagen
- CANopen 4-3
 - Profibus-DP 3-3
- H**
- Heartbeat 4-11, 4-65
- M**
- min_slave_interval 3-7
- N**
- NMT 4-64
- Node Guarding 4-10, 4-65
- P**
- PDO 4-16
- Profibus-DP 3-1
- 353-1DP00 3-12
 - 353-1DP01 3-22

| | | | |
|-------------------------------|----------|-------------------------------|-----------|
| Adressierung | 3-11 | Aufbaurichtlinien | 2-1, 2-10 |
| Aufbaurichtlinien | 3-38 | Busverbinder | 2-2 |
| Datenkonsistenz | 3-5 | Dezentrales System | 1-4 |
| DP-V0 | 3-3, 3-6 | Einbaumaße | 2-3 |
| DP-V1 | 3-3, 3-8 | Einleitung | 1-1 |
| Adressierung | 3-9 | EMV | 2-10 |
| Datenübertragung | 3-3 | Grundregeln | 2-11 |
| Dienste | 3-10 | Komponenten | 1-4 |
| EasyConn | 3-39 | Montage | 2-1, 2-5 |
| Grundlagen | 3-3 | Peripheriemodule | 1-4 |
| GSD-Datei | 3-11 | Schirmung von Leitungen | 2-12 |
| Inbetriebnahme | 3-43 | Sicherheitshinweise | 1-2 |
| Leitungsabschluss | 3-39 | Störeinwirkungen | 2-10 |
| Master | 3-4 | Übersicht | 1-3 |
| Netze | 3-41 | Umgebungsbedingungen | 1-5 |
| Segmentlänge | 3-38 | Verdrahtung | 2-6 |
| Slave | 3-4 | Frontstecker | 2-8 |
| Datenverkehr | 3-5 | Zentrales System | 1-4 |
| Stecker | 3-40 | Systemübersicht | |
| Systemübersicht | 3-2 | CANopen | 4-2 |
| Token-Passing-Verfahren | 3-5 | Profibus-DP | 3-2 |
| Übertragungsprotokoll | 3-5 | <i>T</i> | |
| S | | Technische Daten | |
| SDO | 4-20 | 353-1CA00 | 4-66 |
| System 300V | | 353-1DP00 | 3-46 |
| Aderquerschnitt | 1-5 | 353-1DP01 | 3-46 |
| Aufbau | 2-4 | | |