

ANTAL ELECTRONIC

Feldbus- und Kommunikationstechnik

Handbuch

PDP2CAN

Version 3.10

© ANTAL ELECTRONIC
Höfles 4 • 91322 Gräfenberg
Telefon (+49) 09192/9256-0 • Fax (+49) 09192/9256-78

Wichtige Hinweise

Antal Electronic behält sich das Recht vor, Änderungen am vorliegenden Handbuch, an der im Handbuch beschriebenen Software und Produktänderungen, die dem technischen Fortschritt dienen, ohne Vorankündigung vorzunehmen. Kein Teil dieser Dokumentation darf ohne vorherige schriftliche Genehmigung von Antal Electronic in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) reproduziert oder unter Verwendung elektronischer System verarbeitet, vervielfältigt oder an Dritte weitergegeben werden.

Alle Informationen und technische Angaben in diesem Handbuch wurden mit größter Sorgfalt zusammengestellt. Antal Electronic kann jedoch weder Garantie noch juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen.

Wir weisen darauf hin, daß die in der Dokumentation verwendeten Soft- und Hardwarebezeichnungen und Markennamen der jeweiligen Firmen im allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Copyright © 1997, 1998, 1999, 2000 Antal Electronic. Alle Rechte vorbehalten.

Version 3.10
Juni 2000

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | SICHERHEITSTECHNISCHER HINWEIS | 4 |
| 2 | EINLEITUNG | 5 |
| 3 | HARDWAREAUFBAU | 6 |
| 3.1 | ANZEIGEELEMENTE UND ANSCHLUßMÖGLICHKEITEN | 6 |
| 3.2 | PROFIBUS-BUS-SCHNITTSTELLE | 8 |
| 3.3 | DIP-SCHALTER | 8 |
| 4 | KONFIGURATION DES MODULS | 9 |
| 4.1 | SCHALTER SW1 | 9 |
| 5 | FUNKTIONSWEISE | 10 |
| 5.1 | PARAMETRIERUNG | 10 |
| 5.1.1 | <i>Aufbau Parameterbyte</i> | 13 |
| 5.2 | KONFIGURIERUNG | 14 |
| 5.3 | DATENAUSTAUSCH | 14 |
| 5.4 | DIAGNOSEBEREICH | 14 |
| 5.5 | EMERGENCYBEREICH | 16 |
| 5.6 | CAN-MODULE PARAMETRIEREN (SDO-TRANSFER) | 17 |
| 5.6.1 | <i>Expedited-Transfers (bis zu 4 Bytes Dateninhalt)</i> | 17 |
| 5.6.2 | <i>Segmented-(„Normal“)-Transfers (> 4 Bytes Dateninhalt)</i> | 19 |
| 5.6.3 | <i>SDO-Timeout</i> | 20 |
| 5.6.4 | <i>Handshaking SDOtx (Sende-SDO)</i> | 20 |
| 5.6.5 | <i>Handshaking SDOrx (Empfangs-SDO)</i> | 20 |
| 5.7 | AUFSTARTVORGANG CAN-NETZWERK (CANOPEN-MASTER) | 21 |
| 5.8 | BEISPIEL | 21 |
| 6 | PROJEKTIERUNG BEI STEP 7 | 24 |
| 7 | CANOPEN | 27 |
| 7.1 | ALLGEMEIN | 27 |
| 7.1.1 | <i>CANopen</i> | 27 |
| 7.1.2 | <i>Übertragungsmedium</i> | 28 |
| 7.1.3 | <i>Buszugriffsverfahren</i> | 28 |
| 7.1.4 | <i>Can-Baudrate</i> | 29 |
| 7.1.5 | <i>Verkabelung unter CAN-Bus</i> | 29 |
| 7.1.6 | <i>Leistungsabschluß</i> | 30 |
| 7.1.7 | <i>Telegrammaufbau</i> | 30 |
| 8 | TECHNISCHE DATEN | 35 |
| | ALLGEMEINER HINWEIS! | 35 |

1 Sicherheitstechnischer Hinweis

- ! Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von qualifiziertem Personal vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.
- ! Die Montage, Aufstellung und Verdrahtung darf nur im spannungslosen Zustand der Baugruppe vorgenommen werden.
- ! Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.
- ! Spannungsversorgung nur über VDE-geprüfte und CE-gekennzeichnete Netzteile vornehmen.
- ! Es ist auf richtigen Anschluß der Spannungsversorgung und der Datenleitung zu achten.
- ! Wird das Produkt aus kalter Umgebung in den Arbeitsraum gebracht, kann Betauung auftreten. Vor Inbetriebnahme muß das Produkt absolut trocken sein. Das Produkt nicht in der Nähe von Wasser oder feuchter Umgebung montieren oder installieren.
- ! Das Produkt nicht auseinanderbauen oder das Gehäuse entfernen. Beim Öffnen des Gehäuses erlischt der Garantieanspruch.

2 Einleitung

Der PDP-2-CAN-Buskoppler ermöglicht Ihnen, Daten aus einem PROFIBUS-DP-Kreis in einen sekundären CAN-Kreis zu transformieren und umgekehrt.

Die PROFIBUS-Seite ist dabei als DP-Slave ausgelegt. Die Schnittstellen entsprechen der EN 50170 und sind über DC/DC-Wandler und Optokoppler galvanisch getrennt. Für die Abwicklung des Protokolls ist ein C515C Microcontroller, unterstützt vom SPC3 ASIC (Siemens), zuständig. Der DP-Slave unterstützt das komplette DP-Protokoll gemäß EN50170. Die Baudrate von 9.6kBaude bis 12Mbaude wird automatisch erkannt. **Die Menge der Ein- und Ausgangsinformationen beträgt maximal 320 Byte (das entspricht 10 CAN-Module mit je 2PDO's in Sende- und Empfangsrichtung zu je 8 Byte).**

Die CANopen-Seite ist als eigenständiger Master ausgelegt, der über den PROFIBUS gesteuert werden kann. Es lassen sich damit bis zu 15 CANopen-Slave Module, nach dem CiA-Standard DS-301 Version 3.0, an einem CANopen-Netzwerk betreiben. Alle parametrisierten Module werden vom CANopen-Master erkannt, aufgestartet, und deren Betriebszustand überwacht. Für den Datenaustausch werden bis zu 5 Sende-PDOs und 5 Empfangs-PDOs verwaltet, sowie einem Emergency-Telegramm zur Fehlernachricht an den Master. Zur Parametrierung der CANopen-Slave-Module können beliebige SDOs verschickt und empfangen werden.

Die CAN-Bus-Schnittstelle entspricht ISO/DIS 11898 und ist mit 1 kV DC galvanisch getrennt. Sie wurde mit dem CAN-Bus-Treiberbaustein 82C250 und dem integrierten Basic-CAN-Controller des C515C Microcontroller realisiert. Das serielle Interface (RS232, RS422, RS485 oder TTY) ist ebenfalls mit 1 kV DC galvanisch getrennt.

3 Hardwareaufbau

3.1 Anzeigeelemente und Anschlußmöglichkeiten

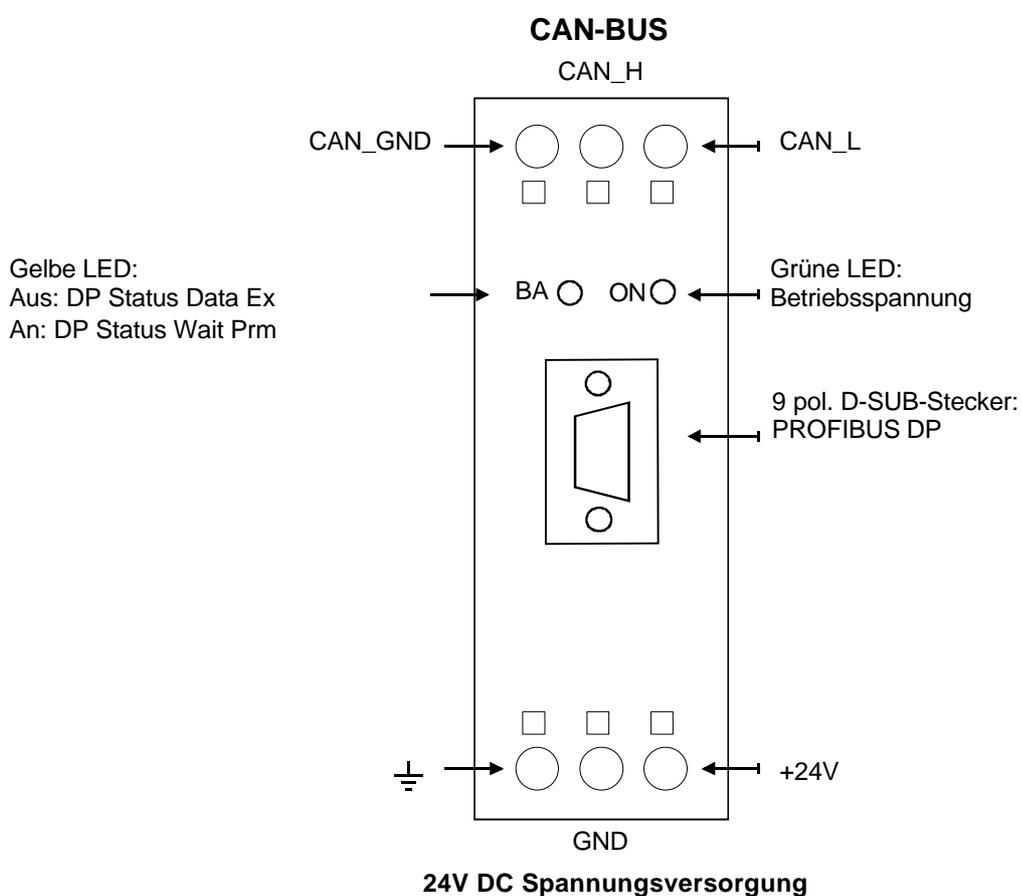


Abbildung 1 Anzeige- und Bedienelemente des PDP-2-CAN

Als Anzeigeelemente besitzt das PDP-2-CAN zwei LEDs. Mit der grünen LED wird eine korrekte Spannungsversorgung und mit der gelben LED der Status des PROFIBUS-Moduls angezeigt.

Der D-SUB-Anschluß stellt den PROFIBUS-DP Anschluß dar. Über die beiden dreipoligen Schraubklemmen wird zum einen die Spannung eingespeist und zum anderen der CAN-Bus angeschlossen.

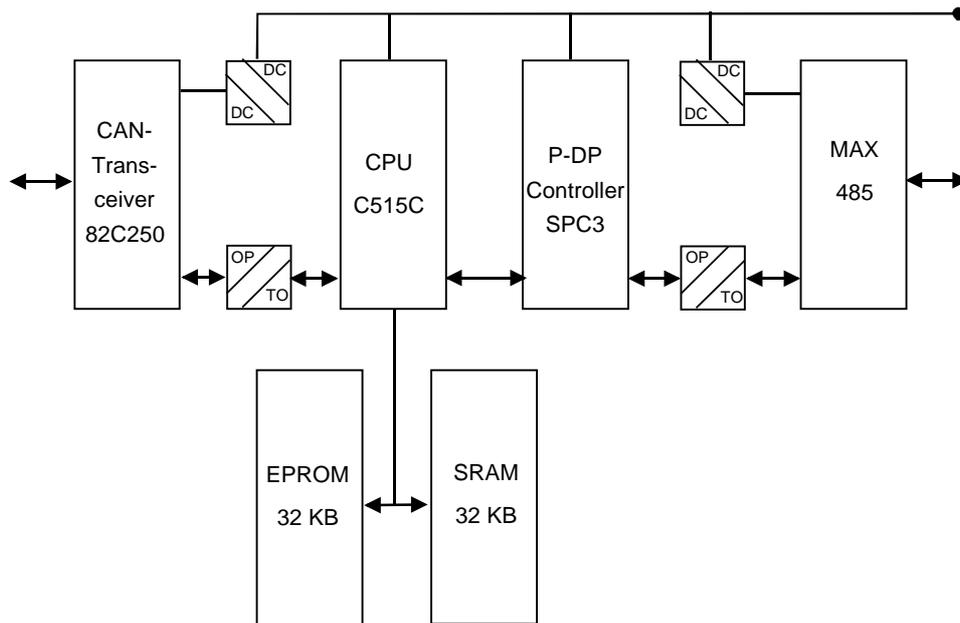


Abbildung 2: Blockschaltbild PDP2CAN

Die CAN-Bus-Schnittstelle entspricht ISO/DIS 11898 und ist mit 1 kV DC galvanisch getrennt. Sie wurde mit dem CAN-Bus-Treiberbaustein 82C250 und dem integrierten Basic-CAN-Controller des C515C Microcontroller realisiert. Für die Abwicklung des PROFIBUS-Protokolls ist SPC3 ASIC von Siemens zuständig. Er unterstützt das komplette PROFIBUS-DP Protokoll nach EN 50170. Die Schnittstelle ist über einen DC/DC-Wandler und Optokoppler galvanisch getrennt. Abbildung 2 zeigt das Blockschaltbild vom PDP2CAN.

3.2 PROFIBUS-Bus-Schnittstelle

Potentialtrennung: 1 kV DC über Optokoppler und DC/DC-Wandler

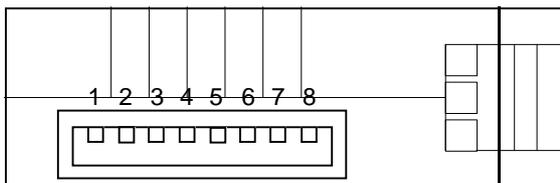
Übertragungsrate: 9,6kBit/s ... 12MBit/s

Belegung der 9 pol. D-SUB-Stecker

| Pin Nr. | Belegung |
|---------|--|
| 1 | n.c. |
| 2 | n.c. |
| 3 | Datenleitung B (RxD/TXD-P) |
| 4 | n.c. |
| 5 | DGND (Datenübertragungspotential, Masse zu 5V) |
| 6 | VP (Versorgungsspannung der Abschlußwiderstände) |
| 7 | n.c. |
| 8 | Datenleitung A (RxD/TxD-N) |
| 9 | n.c. |

3.3 DIP-Schalter

Der DIP-Schalter dient zur Einstellung der PROFIBUS-Moduladresse und der CAN-Baudrate. Er befindet sich auf der Unterseite des Moduls.



DIP-Schalter:

Abbildung 3 Lage der DIP-Schalter

4 Konfiguration des Moduls

4.1 Schalter SW1

Mit dem DIP-Schalter wird die PROFIBUS Moduladresse (Knotenadresse) und das CAN-Protokoll eingestellt. Über die Schalter 1 bis 7 wird die Knotenadresse im Bereich 1 bis 126 eingestellt. Die Schalter 8 konfiguriert das CAN-Protokoll (CANopen und CAN-Layer-2- Protokoll). Dabei gilt folgende Zuordnung:

Knotenadresse:

| | | | | | | | |
|---------------|---|---|---|---|----|----|----|
| Modul-ID: | 1 | 2 | 4 | 8 | 16 | 32 | 64 |
| Schalter Nr.: | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Die Einstellungen „0“ und „126“ sind nicht zulässig, es wird dann automatisch die Defaulteinstellung 1 verwendet.

Schalter ON bedeutet Bit = logisch 1
Schalter OFF bedeutet Bit = logisch 0

CAN-Protokoll:

Schalter S8 bestimmt das relevante CAN-Protokoll. Derzeit ist das CAN-Protokoll auf Layer 2 Basis und das weiter verbreitete Protokoll CANopen der CAN-Nutzerorganisation CiA implementiert. Weitere kundenspezifische Protokolle können erstellt werden.

| Protokoll | S8 |
|------------------|-----------|
| CAN-Layer-2 | off |
| CANopen | on |

Schalter ON bedeutet Bit = logisch 1
Schalter OFF bedeutet Bit = logisch 0

5 Funktionsweise

Der PDP-2-CAN-Buskoppler ermöglicht Ihnen, Daten aus einem PROFIBUS-DP-Kreis in einen sekundären CAN-Kreis zu transformieren und umgekehrt. Bevor nun Daten zwischen PROFIBUS und CAN ausgetauscht werden können, muß in der Parametrierungsphase und in der Konfigurierungsphase die Anzahl der CANopen-Module im CAN-Kreis, sowie deren Modul-ID und Länge der entsprechenden PDO's angegeben werden. Nach erfolgreicher Parametrierung und Konfigurierung werden die CAN-Module aufgestartet und überwacht. Über den Diagnosebereich im Prozeßabbild wird der Status der CAN-Module angezeigt, zusätzlich werden hier die vom CAN-Bus kommenden Emergency-Telegramme eingeblendet. Über ein SDO-Fenster im Prozeßabbild können CANopen-Module mittels SDO-Telegramme parametrisiert werden. Die Ein- und Ausgabedaten werden pro projektierten Guardingzyklus aktualisiert.

5.1 Parametrierung

Mit dem Parametriertelegramm identifiziert sich der Master mit dem Slave und legt fest, in welchem Modus der Slave arbeiten soll. Über anwenderspezifische Parameter werden folgende CAN-Parameter festgelegt (die Parametrierung wird an einem Beispiel verdeutlicht) :

- ◆ CAN-Baudrate
- ◆ Synchrontransfer
- ◆ Anzahl der CANopen-Knoten im CAN-Kreis, sowie deren Modul-ID und PDO-Längen in Sende und Empfangsrichtung

Die Parameterfolge im Einzelnen:

0x00: Das erste Parameterbyte muß immer 0x00 sein und wird für Profibusinterne Zwecke verwendet

CAN-Baudrate

Im ersten Byte im Parametertelegramm wird die CAN-Baudrate eingestellt. Es werden folgende CAN-Baudraten unterstützt:

| WERT(hex) | Baudrate |
|-----------|-------------|
| 00 | 1000 KBit/s |
| 01 | 500 KBit/s |
| 02 | 250 KBit/s |
| 03 | 125 KBit/s |
| 04 | 100 KBit/s |
| 05 | 50 KBit/s |
| 06 | 20 KBit/s |
| 07 | 10 KBit/s |

SYNC TIME

Hier wird die Sync_time in Einheiten von 1 ms (10 ms bis 64 ms möglich) eingestellt. Wertebereich (0A..40_(hex)). Sync-time = 0 bedeutet Synchron-Modus wird nicht unterstützt.

-Anzahl CAN-Module

Hier wird die Anzahl, der im CAN-Kreis befindlichen CAN-Knoten angegeben (Wertebereich: 1..15)

Parameter CAN-Modul 1 (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

-Modul-ID (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Modul -ID des CAN-Knotens eingetragen (Wertebereich: 1..127; siehe Parameterbyte).

-Länge PDO 1 rx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Empfangspdo 1 festgelegt (Siehe Parameterbyte).

-Länge PDO 2 rx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Empfangspdo 2 festgelegt (Siehe Parameterbyte).

-Länge PDO 3 rx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Empfangspdo 3 festgelegt (Siehe Parameterbyte).

-Länge PDO 4 rx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Empfangspdo 4 festgelegt (Siehe Parameterbyte).

-Länge PDO 5 rx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Empfangspdo 5 festgelegt (Siehe Parameterbyte).

-Länge PDO 1 tx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Sendepdo 1 festgelegt (Siehe Parameterbyte).

-Länge PDO 2 tx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Sendepdo 2 festgelegt (Siehe Parameterbyte).

-Länge PDO 3 tx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Sendepdo 3 festgelegt (Siehe Parameterbyte).

-Länge PDO 4 tx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Sendepdo 4 festgelegt (Siehe Parameterbyte).

-Länge PDO 5 tx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Sendepdo 5 festgelegt (Siehe Parameterbyte).

Es sind nur die PDO-Längen im Parametertelegramm einzutragen, die auch vom CAN-Modul unterstützt werden.

.

Parameter CAN-Modul n (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

-Modul-ID (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Modul -ID des CAN-Knotens eingetragen (Wertebereich: 1..127; siehe Parameterbyte).

-Länge PDO 1 rx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Empfangspdo 1 festgelegt (Siehe Parameterbyte).

-Länge PDO 2 rx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Empfangspdo 2 festgelegt (Siehe Parameterbyte).

-Länge PDO 3 rx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Empfangspdo 3 festgelegt (Siehe Parameterbyte).

-Länge PDO 4 rx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Empfangspdo 4 festgelegt (Siehe Parameterbyte).

-Länge PDO 5 rx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Empfangspdo 5 festgelegt (Siehe Parameterbyte).

-Länge PDO 1 tx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Sendepdo 1 festgelegt (Siehe Parameterbyte).

-Länge PDO 2 tx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Sendepdo 2 festgelegt (Siehe Parameterbyte).

-Länge PDO 3 tx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Sendepdo 3 festgelegt (Siehe Parameterbyte).

-Länge PDO 4 tx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

Hier wird die Länge vom Sendepdo 4 festgelegt (Siehe Parameterbyte).

-Länge PDO 5 tx (siehe auch nächstes Kapitel 5.1.1 Aufbau Parameterbyte)

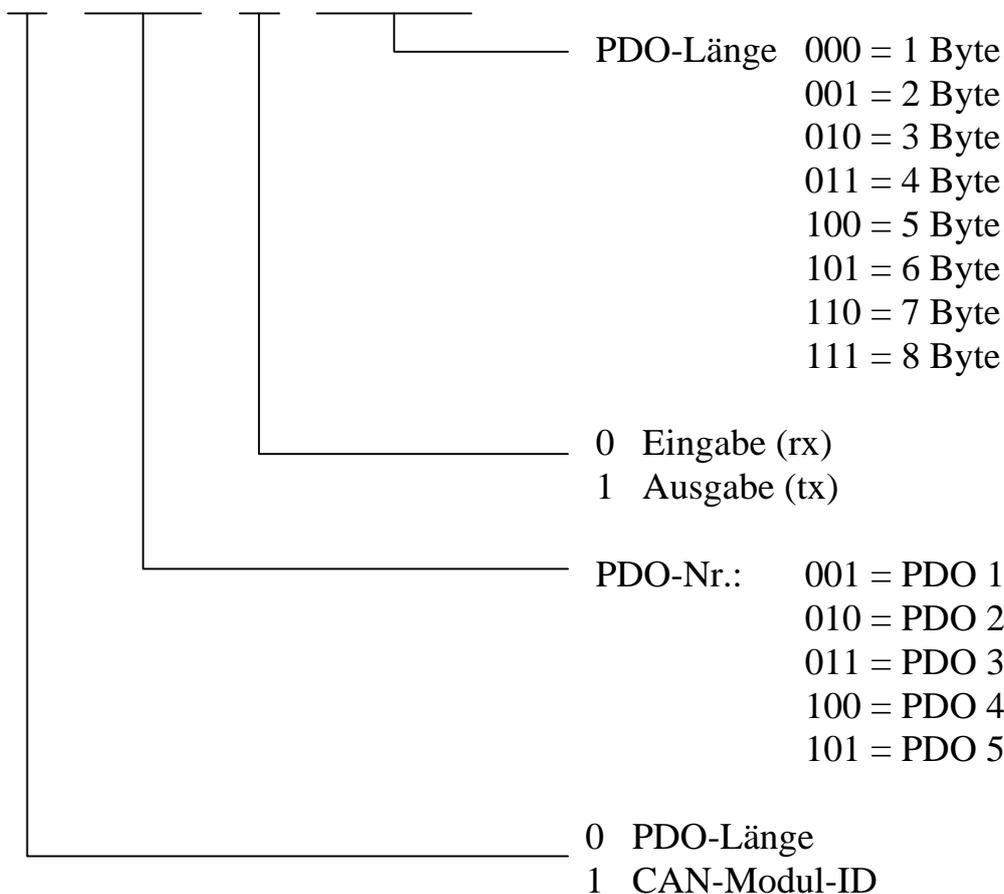
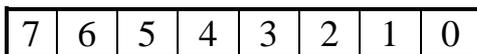
Hier wird die Länge vom Sendepdo 5 festgelegt (Siehe Parameterbyte).

Es sind nur die PDO-Längen im Parametertelegramm einzutragen, die auch vom CAN-Modul unterstützt werden.

5.1.1 Aufbau Parameterbyte

Um eine möglichst effiziente und speichersparende Parametrierung zu ermöglichen werden die Längen der unterstützten PDO's durch folgende Festlegung definiert. Bit 7 kennzeichnet CAN-Modul-ID oder PDO-Länge, die nächsten 3 Bit kennzeichnen die PDO-Nr, Bit 3 kennzeichnet jeweils Eingangs- oder Ausgangspdo und die niederwertigen 3 Bits die Länge des PDO's

Parameterbyte:



z.Bsp.: Modul-ID 1 --> 1 000 0 001 = 81 (hex)
 Modul-ID 32 --> 1 010 0 000 = A0 (hex)

PDO 1 rx, Länge 3Byte --> 0 001 0 010 = 12 (hex)
 PDO 4 tx, Länge 4Byte --> 0 100 1 011 = 4B(hex)

5.2 Konfigurierung

Nach dem Parametrieren hat der Master ein Konfiguriertelegramm an den entsprechenden Slave zu schicken. Über das Konfiguriertelegramm erhält der Slave die Informationen über die Länge der Ein- und Ausgabedaten.

Das Konfiguriertelegramm stellt der Anwender im Projektierungswerkzeug zusammen, wo er evtl. auch den Adressbereich angeben kann, in dem die Nutzdaten abgelegt sind (siehe Beispiel).

Entdeckt der Slave bei der Überprüfung, daß die Ein/Ausgabedatenlängen nicht mit der Parametrierung übereinstimmen, meldet er bei späterer Diagnoseabfrage falsche Konfigurierung an den Master. Er ist dann nicht für den Nutzdatenverkehr bereit.

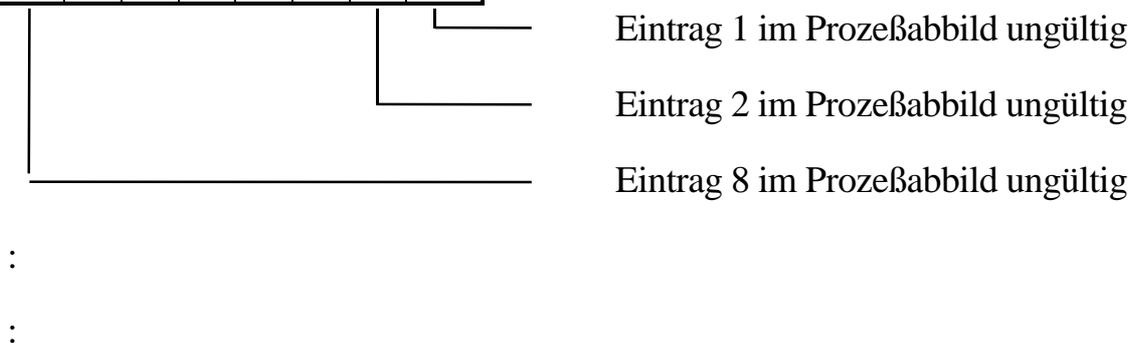
5.3 Datenaustausch

Nach dem der Master im Abschluß der Hochlaufphase die fehlerfreie Parametrierung und Konfigurierung erkannt hat, sendet er Datenaustauschtelegramme. Dazu sendet der PROFIBUS-Master zyklisch alle Daten der parametrierten Sendeidentifizier ans PDP-2-CAN. Jede Datenänderung (auf CANopen-Seite oder DP-Seite) wird sofort beim nächsten Zyklus übertragen. Zusätzlich werden die Eingabedaten der CANopen-Module pro Guardingzyklus (Default 1s) aktualisiert.

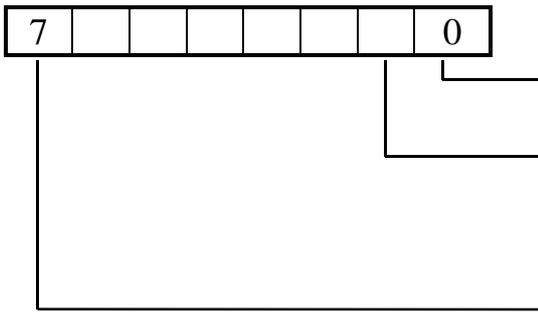
5.4 Diagnosebereich

Der Diagnosebereich besteht aus 8 Bytes. Er gibt darüber Auskunft welcher Bereich im Prozeßabbild ungültig und welches CAN-Modul nicht im Zustand operational ist.

Byte 0



Byte 5

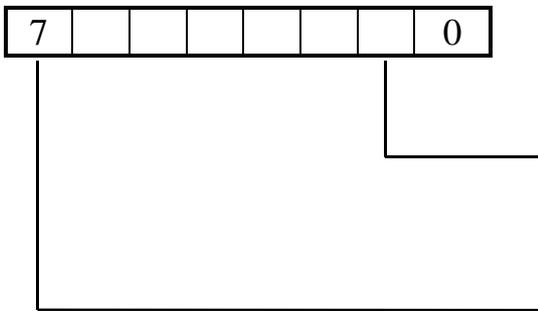


Eintrag 41 im Prozeßabbild ungültig

Eintrag 42 im Prozeßabbild ungültig

Eintrag 48 im Prozeßabbild ungültig

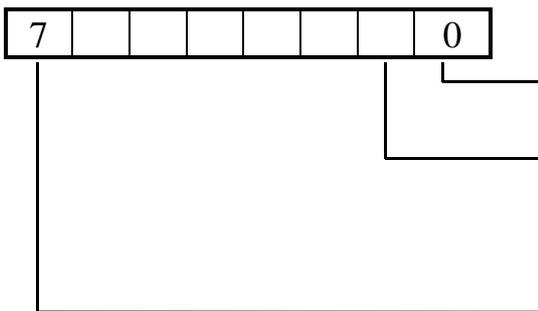
Byte 6



CAN-Modul 1 ist nicht operational

CAN-Modul 7 ist nicht operational

Byte 7



CAN-Modul 8 ist nicht operational

CAN-Modul 9 ist nicht operational

CAN-Modul 15 ist nicht operational

5.5 Emergencybereich

In dem Emergencybereich werden die EMERGENCY-Telegramme vom CAN-Bus kommend eingeblendet. Das erste vom CAN-Bus kommende EMERGENCY-Telegramm wird direkt in den Emergencybereich kopiert alle weiteren EMERGENCY-Telegramme werden in einem Ringpuffer im PDP2CAN zwischengespeichert und erst dann in den Emergencybereich geschrieben wenn das vorhergehende EMERGENCY-Telegramm vom Anwender gelesen wurde. Es ist aus diesem Grunde ein Handshaking programmiert worden. Das Emergency -Telegramm hat folgenden Aufbau:

EMERGENCY:

Telegrammlänge (gesamt): 11

Byte 0: CANopen Modul-ID (1 bis 15)

Byte 1: 0 bis 8 (Datenlänge)

Byte 2 bis 9: (Datenbytes)

Handshaking für EMERGENCY-Telegramme

Die Handshaking-Bits befinden sich im Highbyte vom Byte 0 im SDO-Telegramm.

Bit 4: (im Byte 0 Empfangs-SDO) gesetzt, zeigt an, neues EMERGENCY-Telegramm im Emergencybereich.

Bit 5: (im Byte 0 Empfangs-SDO) gesetzt, zeigt an, Daten im Emergencybereich sind gültig.

Bit 6: (im Byte 0 Sende-SDO) gesetzt, zeigt an, Daten sind vom Anwender gelesen worden.

Ablauf:

PDP2CAN: 1. wenn Bit 4 und Bit 6 = 0 und EMERGENCY-Telegramme im Ringbuffer, schreibe erstes EMERGENCY-Telegramm in Emergency-Bereich, setze Bit 4 und Bit 5

2. warte auf gesetztes Bit 6

3. wenn Bit 6 = 1, lösche Emergency-Bereich, lösche Bit 4 und Bit 5

Anwender: 1. warte auf gesetztes Bit 4 und Bit 5, lese EMERGENCY-Telegramm, setze Bit 6

2. warte auf gelöschte Bits 4 und Bit 5, lösche Bit 6

5.6 CAN-Module parametrieren (SDO-Transfer)

Die SDO-Kommunikation (SDO = **S**ervice **D**ata **O**bject) ist ein bestätigter bzw. quittierter Dienst des CANopen-Protokolls, d. h. daß man bei fehlerfreiem Betrieb auf eine SDO-Anfrage (Request) eine Antwort (Response) von dem angesprochenen Modul erhält. SDOs werden normalerweise für die Parametereinstellung (Write-Request-SDO) oder -abfrage (Read-Request-SDO) an einem Modul (CANopen-Slave) benötigt und sind für die schnelle Prozessdatenübertragung, die über PDOs (**P**rocess **D**ata **O**bjects) abgewickelt wird, nicht geeignet.

Es können sowohl „**Expedited**“-Transfers (bis zu 4 Byte Daten pro Transfer), als auch „**Segmented**“(Normal)-Transfers (beliebige Datenlänge pro Transfer) von dem PDP-2-CAN ausgeführt werden.

Die gesamte SDO-Kommunikation wird von dem PDP-2-CAN über ein 11 Byte großes Fenster im Prozessabbild abgewickelt.

ACHTUNG

Nach jeder Anforderung muß die Antwort des angesprochenen Moduls abgewartet werden, d. h. daß immer nur ein SDO-Transfer gleichzeitig abgewickelt werden kann!

Die SDO-Telegramme haben folgenden Aufbau:

5.6.1 Expedited-Transfers (bis zu 4 Bytes Dateninhalt)

CAL_WRITE_REQ (SDO-Variable Write senden):

Telegrammlänge (gesamt): 11

Byte 0: Handshake-Byte

Byte 1: 36 (Telegrammtyp)

Byte 2: CANopen Modul-ID (1 bis 15)

Byte 3: Index (High Byte)

Byte 4: Index (Low Byte)

Byte 5: Subindex

Byte 6: 1 bis 4 (Anzahl gültiger Datenbytes)

Byte 7 bis 10: (Datenbytes)

CAL_READ_REQ (SDO-Variable Read senden):

Telegrammlänge (gesamt): 6
Byte 0: Handshake-Byte
Byte 1: 39 (Telegrammtyp)
Byte 2: CANopen Modul-ID (1 bis 15)
Byte 3: Index (High Byte)
Byte 4: Index (Low Byte)
Byte 5: Subindex

CAL_READ_CNF_P (SDO-Var-Read-Resp pos. empf):

Telegrammlänge (gesamt): 11
Byte 0: Handshake-Byte
Byte 1: 40 (Telegrammtyp)
Byte 2: CANopen Modul-ID (1 bis 127)
Byte 3: 4 bis 7 (Datenlänge)
Byte 4: Index (High Byte)
Byte 5: Index (Low Byte)
Byte 6: Subindex
Byte 7 bis 10: (Datenbytes)

CAL_READ_CNF_N (SDO-Var-Read-Resp neg. empf):

Telegrammlänge (gesamt): 11
Byte 0: Handshake-Byte
Byte 1: 41 (Telegrammtyp)
Byte 2: CANopen Modul-ID (1 bis 127)
Byte 3: 7 (Datenlänge)
Byte 4: Index (High Byte)
Byte 5: Index (Low Byte)
Byte 6: Subindex
Byte 7: Error_Class
Byte 8: Error_Code
Byte 9: Additional_Code (High Byte)
Byte 10: Additional_Code (Low Byte)

CAL_WRITE_CNF_P (SDO-Var-Write-Resp pos.empf):

Telegrammlänge (gesamt): 7
Byte 0: Handshake-Byte
Byte 1: 37 (Telegrammtyp)
Byte 2: CANopen Modul-ID (1 bis 127)
Byte 3: 3 (Datenlänge)
Byte 4: Index (High Byte)
Byte 5: Index (Low Byte)
Byte 6: Subindex

CAL WRITE CNF N (SDO-Var-Write-Resp neg.empf):

Telegrammlänge (gesamt): 11
Byte 0: Handshake-Byte
Byte 1: 38 (Telegrammtyp)
Byte 2: CANopen Modul-ID (1 bis 15)
Byte 3: 7 (Datenlänge)
Byte 4: Index (High Byte)
Byte 5: Index (Low Byte)
Byte 6: Subindex
Byte 7: Error_Class
Byte 8: Error_Code
Byte 9: Additional_Code (High Byte)
Byte 10: Additional_Code (Low Byte)

5.6.2 Segmented-(„Normal“-)-Transfers (> 4 Bytes Dateninhalt)

CAL SEGMENTED REQ (Segmented SDO-Transfer senden):

Telegrammlänge (gesamt): 11
Byte 0: Handshake-Byte
Byte 1: 60 (Telegrammtyp)
Byte 2: CANopen Modul-ID (1 bis 15)
Byte 3: SDO-Telegramm-Specifier (CCS,t,X,n,e,s,c) (Byte 0 des CANopen SDO-Telegramms)
Byte 4: (Byte 1 des CANopen SDO-Telegramms) z. B. Index (Low Byte)
Byte 5: (Byte 2 des CANopen SDO-Telegramms) z. B. Index (High Byte)
Byte 6: (Byte 3 des CANopen SDO-Telegramms) z. B. Subindex
Byte 7 bis 10: (Bytes 4-7 des CANopen SDO-Telegramms) z. B. Datenbytes

CAL SEGMENTED RSP (Segmented SDO-Transfer empfangen):

Telegrammlänge (gesamt): 11
Byte 0: Handshake-Byte
Byte 1: 61 (Telegrammtyp)
Byte 2: CANopen Modul-ID (1 bis 15)
Byte 3: SDO-Telegramm-Specifier (SCS,t,X,n,e,s,c) (Byte 0 des CANopen SDO-Telegramms)
Byte 4: (Byte 1 des CANopen SDO-Telegramms) z. B. Index (Low Byte)
Byte 5: (Byte 2 des CANopen SDO-Telegramms) z. B. Index (High Byte)
Byte 6: (Byte 3 des CANopen SDO-Telegramms) z. B. Subindex
Byte 7 bis 10: (Bytes 4-7 des CANopen SDO-Telegramms) z. B. Datenbytes

5.6.3 SDO-Timeout

SDO-Timeout:

Telegrammlänge (gesamt): 11

Byte 0: Handshake-Byte

Byte 1: 240 (Telegrammtyp)

Byte 2: CANopen Modul-ID (1 bis 15)

Byte 3 bis 10: 0

Dieses Telegramm erscheint im Empfangsbereich (SDOrx) wenn auf eine SDO-Anfrage nach **200 ms** keine Antwort vom Slave gekommen ist.

5.6.4 Handshaking SDOtx (Sende-SDO)

Bit 0: gesetzt, zeigt an, Daten sind gültig, SDO-Telegramm absenden

Bit 0: gelöscht, zeigt an, Daten sind ungültig, warte auf nächstes SDO-Telegramm, zwischen zwei SDO-Telegrammen muß dieses Bit auf 0 gesetzt werden.

5.6.5 Handshaking SDOrx (Empfangs-SDO)

Bit 0: gesetzt, SDO-Transfer ist aktiv

Bit 1: gesetzt, zeigt gültige Daten im Empfangspuffer an (Data Valid)

Bit 2: gesetzt, zeigt an, SDO-Telegramm wurde gesendet (Data Send)

5.7 Aufstartvorgang CAN-Netzwerk (CANopen-Master)

Das Erkennen und Aufstarten der am CAN-Netzwerk angeschlossenen Slave-Module läuft folgendermaßen ab:

1. Einlesen der Guarding COB-ID (Objekt 0x100E, 0x00) aller Module (1 bis 127)
2. Einlesen der Emergency COB-ID (Objekt 0x1014, 0x00), nur der aktiven Module
3. Einlesen der PDO1M2S COB-ID (Objekt 0x1400, 0x01), nur der aktiven Module
4. Einlesen der PDO2M2S COB-ID (Objekt 0x1401, 0x01), nur der aktiven Module
5. Einlesen der PDO1S2M COB-ID (Objekt 0x1800, 0x01), nur der aktiven Module
6. Einlesen der PDO2S2M COB-ID (Objekt 0x1801, 0x01), nur der aktiven Module
7. Einlesen des PDO1M2S Transmisstiontype (Objekt 0x1400, 0x02), nur wenn PDO gültig
8. Einlesen des PDO2M2S Transmisstiontype (Objekt 0x1401, 0x02), nur wenn PDO gültig
9. Schreiben der Guardtime (Objekt 0x100C, 0x00) an die aktiven Module (Default: 1 s)
10. Schreiben des Lifetimefactors (Objekt 0x100D, 0x00) an die aktiven Module (Default: 2)
11. 100 ms warten
12. NMT-Broadcast-Start-Node
13. Guarding der Module startet
14. In jedem Guardingzyklus wird zusätzlich zur Guardinganforderung der aktuelle Zustand der Eingänge (falls vorhanden) abgefragt, sowie die Ausgänge (falls vorhanden) aktualisiert.
15. Antwortet ein aktives Modul auf eine Guardinganforderung nicht mehr, so wird automatisch versucht es wieder aufzustarten und es laufen die Vorgänge 1. bis 12. für dieses Modul nochmals ab, sobald es wieder als Busteilnehmer erkannt wurde.

5.8 Beispiel

Um Fehlersuch- und Inbetriebnahmezeiten zu sparen, empfiehlt sich als erstes, eine Querverweisliste über die in dem Can-Kreis befindlichen I/O-Module, deren CAN-Identifizier und Bytelängen zu erstellen. Die CAN-Identifizier werden als Node Address normalerweise mit DIP-Schaltern auf den Knotenmodulen eingestellt.

Beispiel A

Drei CAN-Busknoten sind folgendermaßen konfiguriert:

1. Knoten:
 - digitales Eingangsmodul mit 1 mal 8Bit Eingängen.
 - digitales Ausgangsmodul mit 1 mal 8Bit Ausgängen.
 - Die Knotenadresse ist auf 11 eingestellt.
2. Knoten:
 - digitales Eingangsmodul mit 8 mal 8Bit Eingängen.
 - digitales Ausgangsmodul mit 8 mal 8Bit Ausgängen.
 - Die Knotenadresse ist auf 32 eingestellt.

3.Knoten:

- digitales Eingangsmodul mit 8 mal 8Bit Eingängen.
- digitales Ausgangsmodul mit 4 mal 64 Bit Ausgängen
- Die Knotenadresse ist auf 2 eingestellt.

Daraus ergibt sich folgende Querverweistabelle:

| | Modul ID | PDO-Länge in Byte | Eintrag anw. Parameter (hex) | Konfigurationsdaten GSD-Datei |
|-------------------------|----------|-------------------|------------------------------|-------------------------------|
| Immer 00 | | | 00 | |
| CAN-Baudrate (500kBaud) | | | 01 | |
| Synchronbetrieb (aus) | | | 00 | |
| Anzahl CAN-Module | | | 03 | |

| | | | | |
|-----------------------|--|--|--|-------|
| EA-Bereich für SDO | | | | SDOrx |
| EA-Bereich für SDO | | | | SDOtx |
| EA-Bereich für Diagn. | | | | DIAG |
| EA-Bereich für Teleg. | | | | EMGY |

| | | | | |
|------------------------|----|---------|----|--------------|
| Modul-ID (1 xxx x xxx) | 11 | | 8B | |
| PDO 1 rx (0 001 0 xxx) | | 1 (000) | 10 | PDOrx 1 Byte |
| PDO 2 rx (0 010 0 xxx) | | | | |
| PDO 3 rx (0 011 0 xxx) | | | | |
| PDO 4 rx (0 100 0 xxx) | | | | |
| PDO 5 rx (0 101 0 xxx) | | | | |
| PDO 1 tx (0 001 1 xxx) | | 1 (000) | 18 | PDOtx 1 Byte |
| PDO 2 tx (0 010 1 xxx) | | | | |
| PDO 3 tx (0 011 1 xxx) | | | | |
| PDO 4 tx (0 100 1 xxx) | | | | |
| PDO 5 tx (0 101 1 xxx) | | | | |

| | | | | |
|------------------------|----|---------|----|--------------|
| Modul-ID (1 xxx x xxx) | 32 | | A0 | |
| PDO 1 rx (0 001 0 xxx) | | 1 (111) | 17 | PDOrx 8 Byte |
| PDO 2 rx (0 010 0 xxx) | | | | |
| PDO 3 rx (0 011 0 xxx) | | | | |
| PDO 4 rx (0 100 0 xxx) | | | | |
| PDO 5 rx (0 101 0 xxx) | | | | |
| PDO 1 tx (0 001 1 xxx) | | 1 (111) | 1F | PDOtx 8 Byte |
| PDO 2 tx (0 010 1 xxx) | | | | |
| PDO 3 tx (0 011 1 xxx) | | | | |
| PDO 4 tx (0 100 1 xxx) | | | | |
| PDO 5 tx (0 101 1 xxx) | | | | |

| | | | | |
|------------------------|---|--------|----|--------------|
| Modul-ID (1 xxx x xxx) | 2 | | 82 | |
| PDO 1 rx (0 001 0 xxx) | | 1(111) | 17 | PDOrx 8 Byte |

| | | | | |
|------------------------|--|---------|----|--------------|
| PDO 2 rx (0 010 0 xxx) | | | | |
| PDO 3 rx (0 011 0 xxx) | | | | |
| PDO 4 rx (0 100 0 xxx) | | | | |
| PDO 5 rx (0 101 0 xxx) | | | | |
| PDO 1 tx (0 001 1 xxx) | | 1 (111) | 1F | PDOtx 8 Byte |
| PDO 2 tx (0 010 1 xxx) | | 1 (111) | 2F | PDOtx 8 Byte |
| PDO 3 tx (0 011 1 xxx) | | 1 (111) | 3F | PDOtx 8 Byte |
| PDO 4 tx (0 100 1 xxx) | | 1 (111) | 4F | PDOtx 8 Byte |
| PDO 5 tx (0 101 1 xxx) | | | | |

Daraus ergeben sich folgende Parametrierdaten und Konfigurierdaten (in hex, durch Komma getrennt):

Hex-Parameter (PDP2CAN-spezifische):

00,01,00,03,8B,10,18,A0,17,1F,82,17,1F,2F,3F,4F,00,00,00,00,00....

Konfigurierungsdaten:

SDOrx
SDOtx
DIAG
EMGY
PDOrx 1 Byte
PDOtx 1 Byte
PDOrx 8 Byte
PDOtx 8 Byte
PDOrx 8 Byte
PDOtx 8 Byte
PDOtx 8 Byte
PDOtx 8 Byte
PDOtx 8 Byte

Bei den Konfigurierungsdaten müssen die Objekte SDOrx, SDOtx, DIAG, EMGY auf jedenfall in obiger Reihenfolge konfiguriert werden; ansonsten meldet das PDP2CAN einen Parametrierfehler.

6 Projektierung bei STEP 7

Gehen Sie wie folgt vor:

GSDdatei PDP2CAN.GSD in den Unterpfad ..\S7DATA\GSD kopieren.

Befehl GSD-Dateien aktualisieren, damit der Hardwarekatalog aktualisiert wird.

Slave PDP2CAN aus dem Pfad 'Weitere FELDGERÄTE' und dort unter 'Sonstige' aktivieren.

Per Drag & Drop ist nun der PDP2CAN-Slave auf dem Profibus(1)-Netz zu platzieren.

Weisen Sie dem Slave die gewünschte Stationsadresse zu:

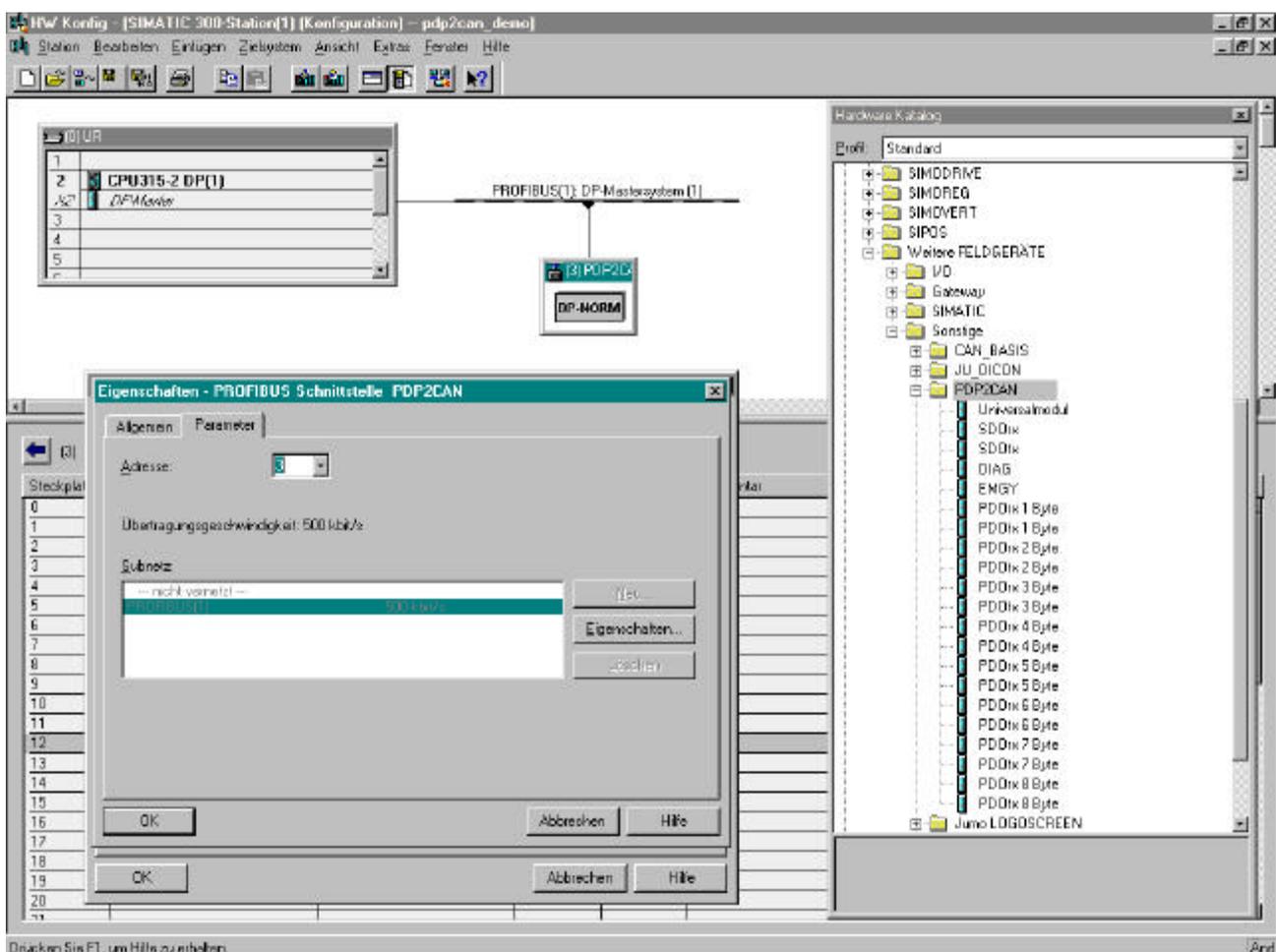


Abbildung 4: Stationsadresse PDP2CAN

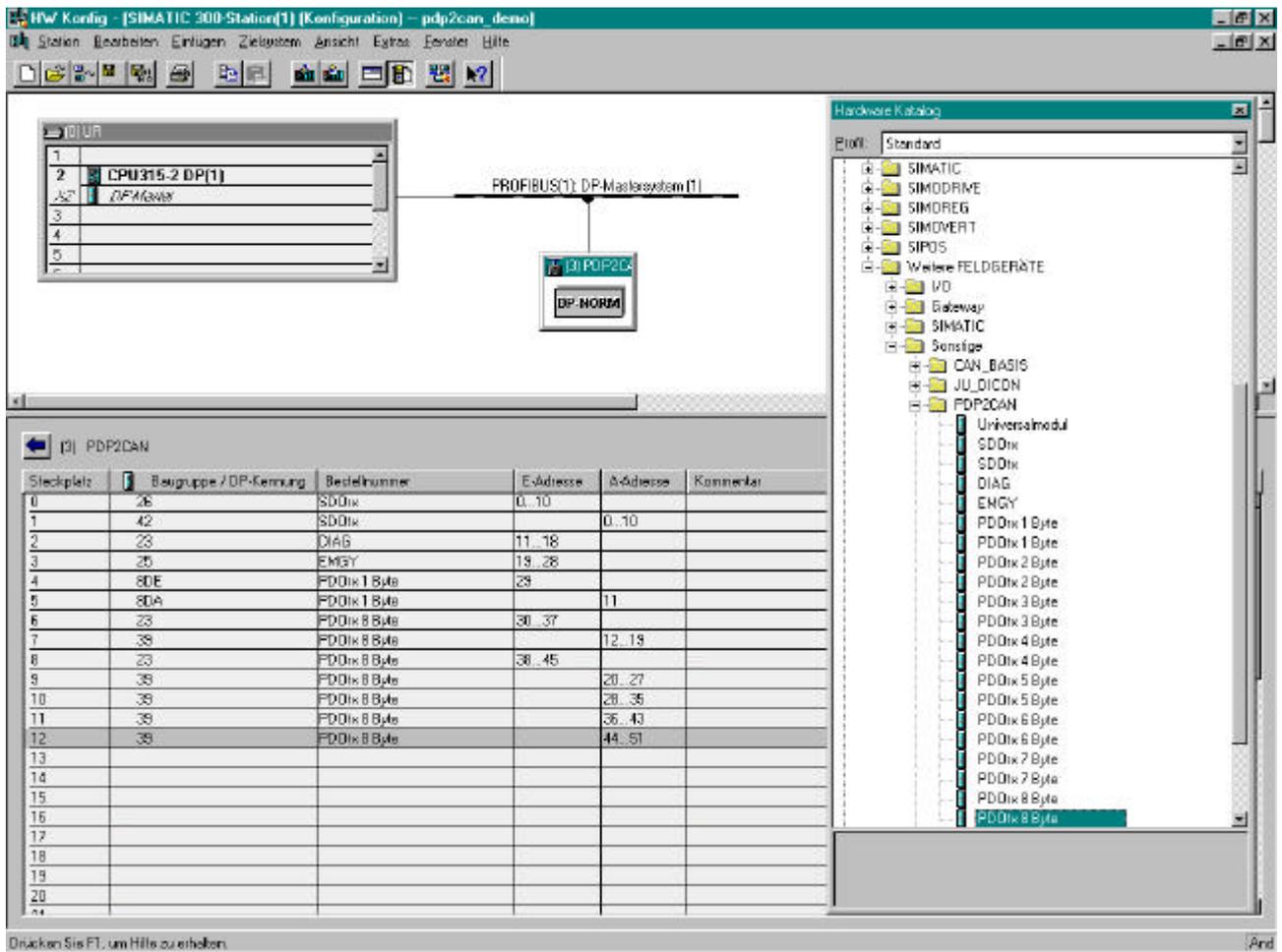


Abbildung 7: Konfiguration DP-Slave

7 CANopen

7.1 Allgemein

Der CAN-Bus (Control Area Network) 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 nochmals 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=CAN application layer). Ein solches Anwenderprofil ist CANopen, dessen Standardisierung der CiA CAN in Automation e.V. übernimmt.

7.1.1 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. 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

Es 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.

7.1.2 Übertragungsmedium

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 Bustreiberbaustein begrenzt. Die maximale Netzausdehnung ist durch Signallaufzeiten begrenzt. Bei 1 MBaud ist z.B. eine Netzausdehnung von 40m und bei 80 kBaud 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 120 Ω Abschlußwiderstand am Ende.

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.

7.1.3 Buszugriffsverfahren

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 priorisierendem 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.

7.1.4 Can-Baudrate

| <u>CAN-Baudrate</u> | <u>Gar. max. Buslänge</u> |
|---------------------|---------------------------|
| 1 Mbaud | 25 m |
| 500 kBaud | 100 m |
| 250 kBaud | 250 m |
| 125 kBaud | 500 m |
| 100 kBaud | 600 m |
| 50 kBaud | 1000 m |
| 20 kBaud | 2500 m |
| 10 kBaud | 5000 m |

7.1.5 Verkabelung unter CAN-Bus

CAN-Bus verwendet als Übertragungsmedium eine abgeschirmte Dreidrahtleitung.

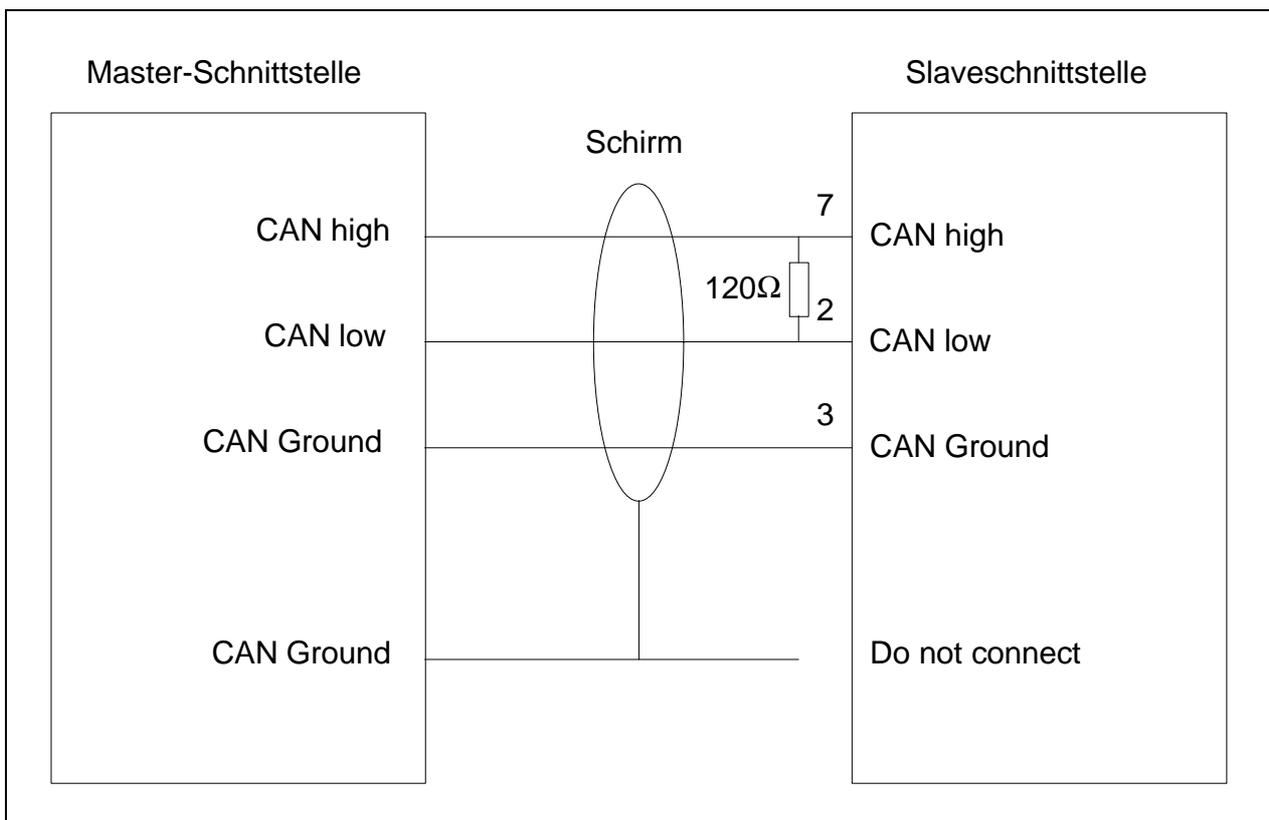


Abb. 7-1: CAN-Bus: Verkabelung

7.1.6 Leitungsabschluß

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

An den Leitungsenden muß das Buskabel immer mit jeweils einem Abschlußwiderstand von 120 W abgeschlossen werden, um Reflexionen und damit Übertragungsprobleme zu vermeiden!

7.1.7 Telegrammaufbau

Alle CANopen Telegramme besitzen folgenden Aufbau:

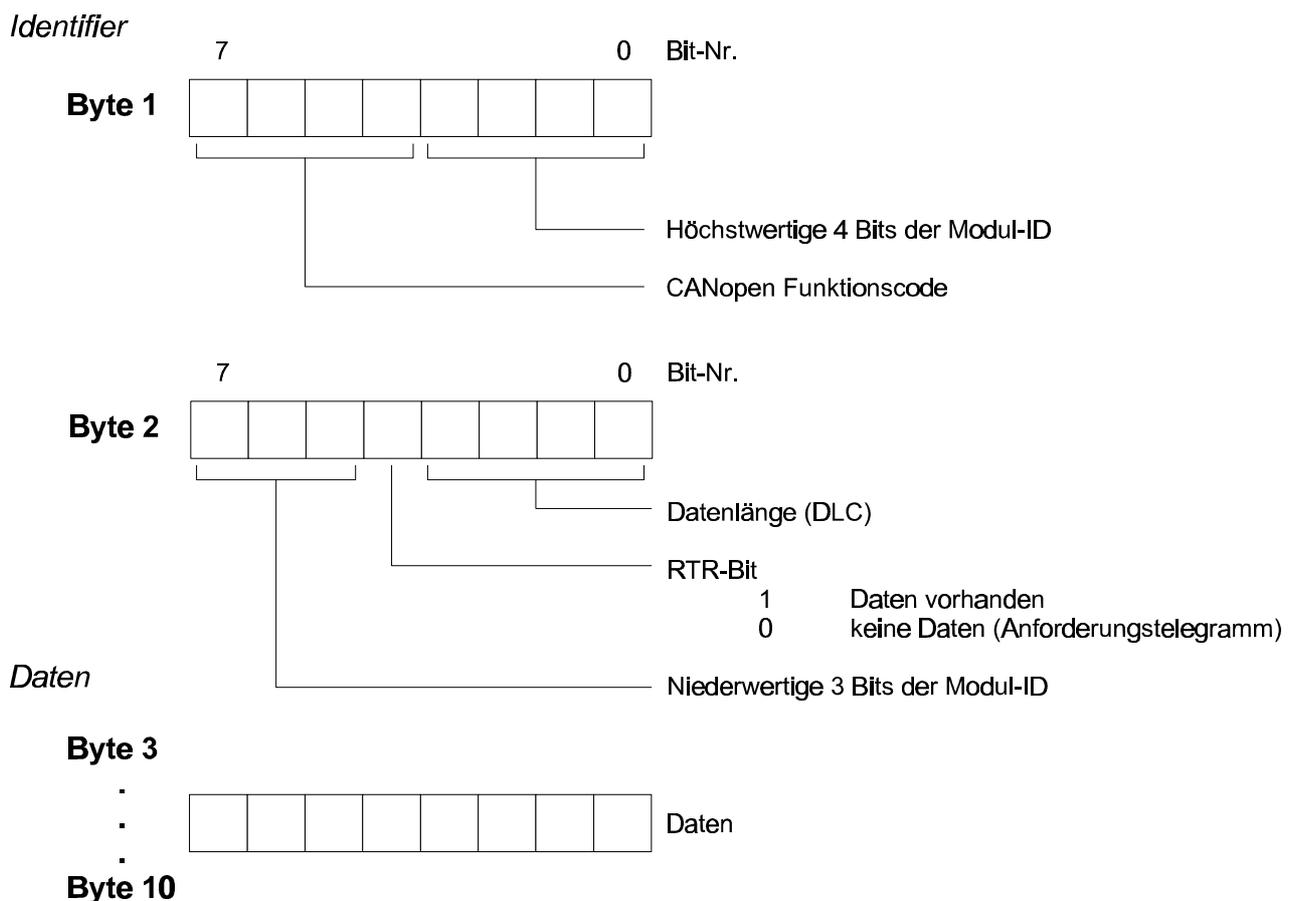


Abb. 7-2: CANopen: Telegrammaufbau

Der Unterschied zu einem Schicht-2-Telegramm besteht in einer zusätzlichen Unterteilung des 2 Byte Identifiers in einen Funktionsteil und einer 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. Jedem Objekt ist ein Funktionscode zugeordnet. Bitte entnehmen Sie den entsprechenden Funktionscode der nachfolgenden Tabelle.

7.1.7.1 CANOPEN FUNKTIONSCODES

Nachfolgend sind die unter CANopen definierten Objekte mit Funktionscode aufgelistet.

| <u>Objekt</u> | <u>Function Code (4 Bits)</u> | <u>Empfänger</u> | <u>Definition</u> | <u>Funktion</u> |
|---------------|-------------------------------|---------------------|-------------------|-----------------------|
| NMT | 0000 | Broadcast | CiA DS-301 | Netzwerkmanagem. |
| EMERGENCY | 0001 | Master | CiA DS-301 | Fehlertelegramm |
| PDO1S2M | 0011 | Master, Slave (RTR) | CiA DS-301 | Digital Eing. Daten 1 |
| PDO1M2S | 0100 | Slave | CiA DS-301 | Digital Ausg. Daten 1 |
| PDO2S2M | 0101 | Master, Slave (RTR) | CiA DS-301 | Analog Eing. Daten 1 |
| PDO2M2S | 0110 | Slave | CiA DS-301 | Analog Ausg. Daten 1 |
| PDO3S2M | 0111 | Master, Slave (RTR) | Applikationspez. | D o. A Eing. Daten 2 |
| PDO3M2S | 1000 | Slave | Applikationspez. | D o. A Ausg. Daten 2 |
| PDO4S2M | 1001 | Master, Slave (RTR) | Applikationspez. | D o. A Eing. Daten 3 |
| PDO4M2S | 1010 | Slave | Applikationspez. | D o. A Ausg. Daten 3 |
| PDO5S2M | 1101 | Master, Slave (RTR) | Applikationspez. | D o. A Eing. Daten 4 |
| PDO5M2S | 1111 | Slave | Applikationspez. | D o. A Ausg. Daten 4 |
| SDO1S2M | 1011 | Master | CiA DS-301 | Konfigurationsdaten |
| SDO1M2S | 1100 | Slave | CiA DS-301 | Konfigurationsdaten |
| Node Guarding | 1110 | Master, Slave (RTR) | CiA DS-301 | Modulüberwachung |

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 DPS 401 Version 1.4" detailliert beschrieben.

7.1.7.2 CANOPEN OBJEKTE

PDO:

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

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. 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. Für die SDO-Transfers sind alle nötigen Fehlermeldungen (nach DS 301) mit "Error-Class", "Error-Code" und "Additional-Code" implementiert.

Emergency Objekt:

Um anderen Teilnehmern am CANopen-Bus interne Gerätefehler mit hoher Priorität mitteilen zu können, werden Emergency Objekt ausgesandt.

Für das Emergency-Telegramm ist der nach dem Boot-Up im Objektverzeichnis in der Variablen 1014h fest eingestellte **COB-Identifier** in Hexadezimaldarstellung: **080h + Modul-ID**.

CANopen EMERGENCY-Telegramm-Inhalt:

| Byte-Nr. | Inhalt |
|----------|---|
| 0 | Emergency Error Code (DS-301) low Byte |
| 1 | Emergency Error Code (DS-301) high Byte |

| | |
|---|-------------------------------------|
| 2 | Emergency Error Register (DS-301) |
| 3 | Applikationsspezifischer Error Code |
| 4 | Zusätzliche Error Information 1 |
| 5 | Zusätzliche Error Information 2 |
| 6 | Zusätzliche Error Information 3 |
| 7 | Zusätzliche Error Information 4 |

Node Guarding:

CANopen definiert 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 100Eh fest auf 700h + Modul-ID eingestellt. Wird während des Guardingbetriebs innerhalb der "Guard-Time" (Objekt 100Ch) 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" (100Ch) und "Life-Time-Factor" (100Dh) eingestellt ist, versetzt sich das Modul automatisch in den Zustand "Pre-Operational".

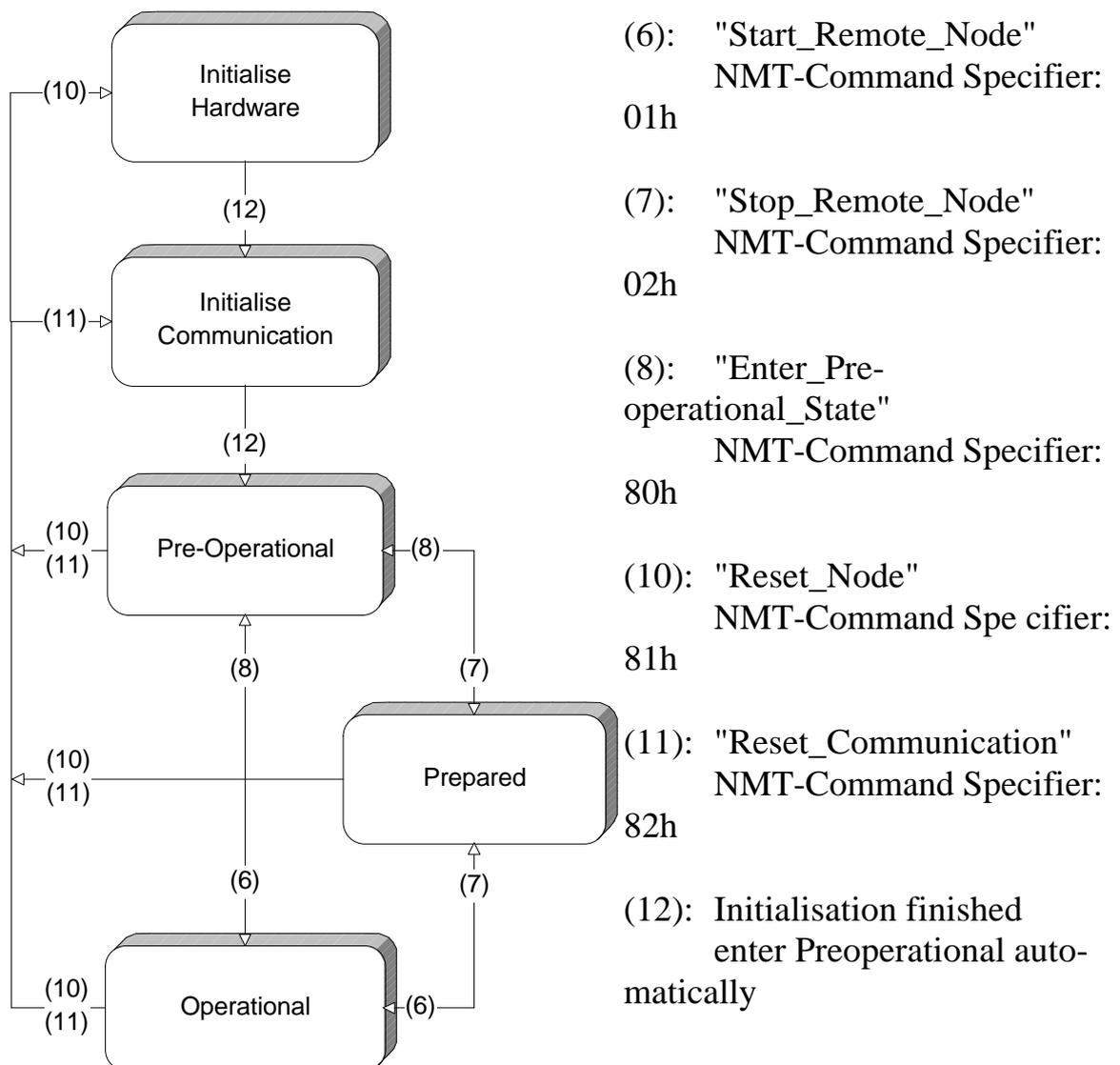
Wird entweder die "Guard-Time" (Objekt 100Ch) oder der "Life-Time-Factor" (100Dh) mittels SDO-Download vom Master auf Null eingestellt, so findet keine Überprüfung auf Ablauf der Guardingzeit statt, und das Modul bleibt im aktuellen Zustand.

NMT

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 0000h. Eine additive Modul-ID ist nicht erforderlich. Die Länge beträgt immer 2 Datenbytes. Das erste Datenbyte enthält den NMT-Command Specifier:

NMT-Services (DS 301):



Das zweite Datenbyte enthält die Modul-ID (00h für ein Broadcast Command).

8 Technische Daten

| | |
|------------------------------|--|
| Spannungsversorgung: | 24V DC |
| Stromaufnahme: | ca. 150mA |
| Galvanische Trennung: | 1 kV DC |
| Busgeschwindigkeit CAN: | max. 1 MBit |
| Busgeschwindigkeit Profibus: | ma. 12 MBit |
| Schutzart: | IP 20 |
| Abmessungen: | Höhe: 80mm Breite: 23mm Tiefe: 90mm |
| Befestigung: | Hutschiennenmontage |
| Umgebungstemperatur | |
| Betrieb: | 0° - 55° C max. 95% Luftfeuchtigkeit ohne Kondensation |
| Lagerung: | 0° - 70° C |

Allgemeiner Hinweis!

Um die EMV Bestimmungen einzuhalten, sind alle Datenleitungen mit einem Schirm zu versehen. Dieser Schirm muß auf das Erdpotential aufgelegt werden. Alle Erdklemmen an unserer Baugruppe sind auch auf das Erdpotential zu legen. Werden diese Maßnahmen nicht getroffen, so übernimmt die Fa. Antal Electronic keine Haftung für Einhaltung der EMV Schutzmaßnahmen.