



DP/CAN-Koppler

PROFIBUS-DP zu CAN-Bus Koppler für CANopen® und CAN Layer 2

700-650-CAN01

Handbuch

Ausgabe 7 / 03.12.2012 ab HW1 & FW 1.29

Handbuch Bestellnummer: 900-650-CAN01/de

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung dieses Handbuches, oder Teilen daraus, vorbehalten. Kein Teil des Handbuches darf ohne schriftliche Genehmigung der Systeme Helmholtz GmbH in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, oder unter Verwendung elektronischer Systeme reproduziert, verarbeitet, vervielfältigt oder verbreitet werden. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Copyright © 2012 by

Systeme Helmholtz GmbH

Hannberger Weg 2, 91091 Großenseebach

Hinweis:

Der Inhalt dieses Handbuches ist von uns auf die Übereinstimmung mit der beschriebenen Hard- und Software überprüft worden. Da dennoch Abweichungen nicht ausgeschlossen sind, können wir für die vollständige Übereinstimmung keine Gewährleistung übernehmen. Die Angaben in diesem Handbuch werden jedoch regelmäßig aktualisiert. Bitte beachten sie beim Einsatz der erworbenen Produkte jeweils die aktuellste Version des Handbuchs, die im Internet unter www.helmholtz.de einsehbar ist und auch heruntergeladen werden kann.

Unsere Kunden sind uns wichtig. Wir freuen uns über Verbesserungsvorschläge und Anregungen.

Änderungen in diesem Dokument:

Stand	Datum	Änderung
4	18.06.2009	Korrektur der Eintragsnummern (Seite 20)
		Heartbeat bei CANopen® (FW. 1.29)
		Hinweis auf konsistente Daten erweitert.
		Hinweis zum Aufstartvorgang erweitert.
5	30.06.2010	Heartbeat Protokoll vom Slave wird unterstützt und kleine Korrekturen
6	22.12.2011	GSD Dateiname für CANopen korrigiert und kleinere Korrekturen
7	23.11.2012	DIP Schalter Tabelle korrigiert

Inhaltsverzeichnis

1	Sicherheitshinweise	7
1.1	Allgemein	7
1.2	Zugangsbeschränkung	8
1.3	Benutzerhinweise	8
1.4	Bestimmungsgemäßer Gebrauch	8
1.5	Bestimmungswidrigen Gebrauch vermeiden!	8
2	Installation und Montage	9
2.1	Vertikaler und horizontaler Aufbau	9
3	Systemübersicht	10
3.1	Anwendung und Funktionsbeschreibung	10
3.2	Anschlüsse	11
3.3	LED-Anzeigen	11
3.4	DIP-Schalter	11
3.5	Lieferumfang	12
3.6	Zubehör	12
4	Projektierung (CANopen® Master)	13
4.1	Gerät installieren und parametrieren	13
4.2	Festlegung des EA-Adressbereiches in der SPS	16
4.3	Konsistente Daten	16
4.4	Maximale Parametergrößen und Adressbereiche	17
4.5	Parametrieren der Module	17
5	Programmierung (CANopen® Master)	19
5.1	Datenaustausch	19
5.2	Aufstartvorgang CAN-Netzwerk	19
5.3	Diagnosebereich	20
5.4	Emergency Telegramme empfangen	21
5.4.1	Emergency Empfangsfach	21
5.4.2	Handshaking für Emergency-Telegramme	21
5.5	CAN-Module parametrieren (SDO-Transfer)	22
5.5.1	Expedited SDO-Transfers (bis zu 4 Bytes Daten)	22

5.5.2	SDO-Timeout	24
5.5.3	Handshaking SDOtx (Sende-SDO)	24
5.5.4	Handshaking SDOrx (Empfangs-SDO)	24
6	Projektierung (CAN Layer 2)	25
6.1	Gerät installieren und parametrieren	25
6.2	Festlegung der EA-Adressbereiches in der SPS	28
6.3	Konsistente Daten	29
6.4	Maximale Parametergrößen und Adressbereiche	29
6.5	Sende und Empfangsnachrichten parametrieren	30
6.6	Receive Objekt parametrieren	31
7	Programmierung (CAN Layer 2)	32
7.1	Datenaustausch	32
7.2	Handshake Bits	32
7.3	Empfangs- und Sende-Objekte	33
7.4	Receive Objekt	34
7.5	Transmit Objekt	35
7.6	Zyklisches Senden des Transmit Objektes	35
8	CANopen® Kommunikation	36
8.1	Allgemein	36
8.2	Objekte	36
8.3	Funktionen	37
8.4	Netzmanagement	38
9	Anhang	40
9.1	Technische Daten	40
9.2	Steckerbelegung	41
9.3	Weiterführende Dokumentation	41

1 Sicherheitshinweise

Beachten Sie die aufgeführten Sicherheitshinweise zu Ihrer eigenen Sicherheit und der Sicherheit Anderer. Die Sicherheitshinweise zeigen Ihnen mögliche Gefahren auf und geben Ihnen Hinweise, wie Sie Gefahrensituationen vermeiden können.

Im vorliegenden Handbuch werden folgende Piktogramme verwendet:



Achtung, macht auf Gefahren und Fehlerquellen aufmerksam



gibt einen Hinweis



Gefahr allgemein oder spezifisch



*Gefahr eines **Stromschlages***

1.1 Allgemein

Der DP/CAN-Koppler wird nur als Bestandteil eines Gesamtsystems eingesetzt.



Der Betreiber einer Maschinenanlage ist für die Einhaltung der für den speziellen Einsatzfall geltenden Sicherheits- und Unfallverhütungsvorschriften verantwortlich.



Bei der Projektierung sind die einsatzspezifischen Sicherheits- und Unfallverhütungsvorschriften zu beachten.



Not-Aus-Einrichtungen gemäß EN 60204 / IEC 204 müssen in allen Betriebsarten der Maschinenanlage wirksam bleiben. Es darf zu keinem undefinierten Wiederanlauf der Anlage kommen.



In der Maschinenanlage auftretende Fehler, die Material- oder Personenschäden verursachen können, müssen durch zusätzliche externe Einrichtungen abgefangen werden. Diese Einrichtungen müssen auch im Fehlerfall einen sicheren Betriebszustand gewährleisten. Solche Einrichtungen sind z.B. elektromechanische Sicherheitsschalter, mechanische Verriegelungen usw. (siehe EN 954-1, Risikoabschätzung).



Sicherheitsrelevante Funktionen niemals über das Bedienterminal ausführen oder einleiten.



*Zutritt zu den
Baugruppen nur für
berechtigte Personen!*

1.2 Zugangsbeschränkung

Die Baugruppen sind offene Betriebsmittel und dürfen nur in elektrischen Betriebsräumen, Schränken oder Gehäusen installiert werden. Der Zugang zu den elektrischen Betriebsräumen, Schränken oder Gehäusen darf nur über Werkzeug oder Schlüssel möglich sein und nur unterwiesenem oder zugelassenem Personal gestattet werden. Siehe auch Kapitel 2.

1.3 Benutzerhinweise

Dieses Handbuch richtet sich an Projektue und Monteure des DP/CAN-Kopplers.

Es soll dem Projektue als Programmierhandbuch und Nachschlagewerk dienen. Dem Monteur sollen alle zur Montage notwendigen Daten bereitgestellt werden.

Der DP/CAN-Koppler ist ausschließlich zum Gebrauch in einem PROFIBUS-DP Netzwerk zu verwenden. Aus diesem Grund sind von Projektue, Anwender und Monteur die für den jeweiligen Einsatzfall geltenden Normen, Sicherheits- und Unfallverhütungsvorschriften unbedingt zu beachten. Der Betreiber des Automatisierungssystems ist für die Einhaltung dieser Vorschriften verantwortlich.

1.4 Bestimmungsgemäßer Gebrauch

Die DP/CAN-Koppler darf nur, wie im Handbuch beschrieben, als Kommunikationssystem verwendet werden.

1.5 Bestimmungswidrigen Gebrauch vermeiden!

Sicherheitsrelevante Funktionen dürfen nicht über den DP/CAN-Koppler allein gesteuert werden.

2 Installation und Montage

Die Installation des DP/CAN-Kopplers muss nach VDE 0100 IEC 364 erfolgen. Da es sich um „OPEN Type“ Baugruppen handelt, müssen sie in einen (Schalt-) Schrank eingebaut werden. Umgebungstemperatur: 0 °C – 60 °C.



Bevor Installationsarbeiten durchgeführt werden, alle Systemkomponenten spannungsfrei schalten.



Gefahr eines Stromschlages!



Bei der Montage sind die einsatzspezifischen Sicherheits- und Unfallverhütungsvorschriften zu beachten.

2.1 Vertikaler und horizontaler Aufbau

Die Baugruppen können sowohl vertikal, als auch horizontal aufgebaut werden.

Zulässige Umgebungstemperatur:

0 bis 60 °C



3 Systemübersicht

3.1 Anwendung und Funktionsbeschreibung

Der DP/CAN-Koppler der Systeme Helmholz GmbH ermöglicht es, beliebige CAN-Teilnehmer mit dem PROFIBUS DP zu verbinden.

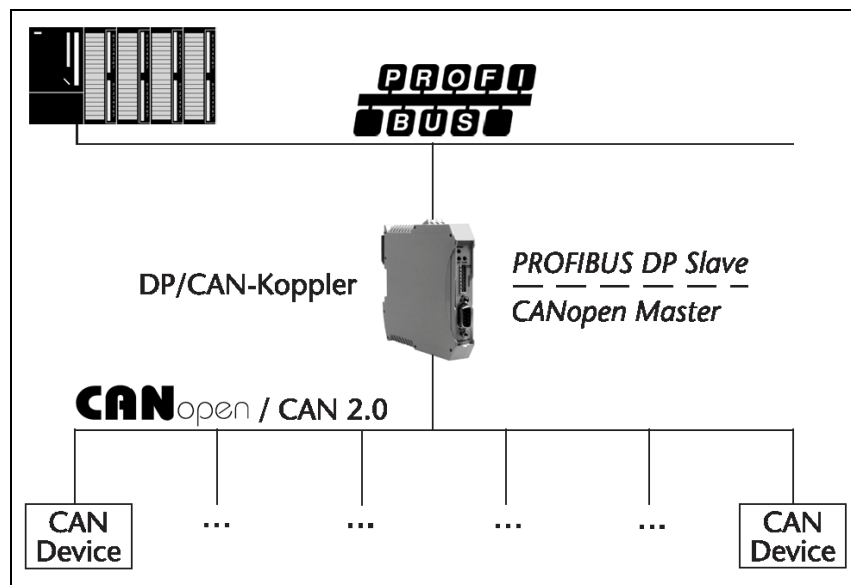
Der DP/CAN-Kopplers ist im Hardwarekonfigurator als ein PROFIBUS Teilnehmer zu parametrieren. Entsprechende GSD-Dateien liegen dem Gerät bei.

Die PROFIBUS-Seite ist als DP-Slave ausgelegt. Die Schnittstellen entsprechen der EN 50170 und sind galvanisch getrennt. Die Baudrate von 9.6kBaude bis 12Mbaude wird automatisch erkannt. Die Größe der Ein- und Ausgangsinformationen beträgt maximal 312 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 4, an einem CANopen[®]-Netzwerk betreiben. Alle parametrisierten Module werden vom CANopen[®]-Master erkannt, gestartet, und deren Betriebszustand überwacht. Für den Datenaustausch können pro Node bis zu 4 Sende-PDOs und 4 Empfangs-PDOs verwaltet werden. Emergency-Telegramme von den CANopen[®] Nodes werden vom DP/CAN-Koppler verarbeitet und an den PROFIBUS Master weitergeleitet. Zur Parametrierung der CANopen[®]-Slave-Module können beliebige SDOs verschickt und empfangen werden.

Alternativ kann der DP/CAN-Koppler auch auf Layer 2 verwendet werden. Hierbei können die CAN-Nachrichten, die in den PROFIBUS eingebündelt werden sollen, frei gewählt werden.

Die CAN-Bus-Schnittstelle entspricht ISO/DIN 11898-2 und ist galvanisch getrennt.



3.2 Anschlüsse

PROFIBUS 9pol. Sub-D Buchse:

Pin	PROFIBUS DP
1	-
2	
3	Datenleitung B
4	-
5	GND
6	VP (Versorgung der Abschlusswiderstände)
7	-
8	Datenleitung A
9	-

CAN-Stecker 3polig (kein Abschlusswiderstand):

1	CAN High
2	CAN-GND
3	CAN-Low

Spannungsversorgung 3polig:

1	GND
2	V-
3	V+

3.3 LED-Anzeigen

Die drei LEDs an der Vorderseite der Baugruppe informieren über den Betriebszustand.

LED Power (Grün): Dauerlicht zeigt an, dass der PROFIBUS läuft und die CPU im Run ist. Langsames Blinklicht zeigt an, dass die CPU im Stop ist.

LED DP (Rot): Es liegt ein Parametrierfehler am PROFIBUS vor.

LED CAN (Gelb): Es werden CAN-Telegramme vom CAN-Bus empfangen.



3.4 DIP-Schalter

Der 8-fach DIP-Schalter an der Gehäusevorderseite ist zur Einstellung der PROFIBUS-Adresse des Gerätes und des CAN Betriebsmodus.

Die Schalter werden von unten nach oben durchgezählt.

Schalter	Funktion		
8	Mode	OFF=Layer 2	ON=CANopen
7	PROFIBUS Adresse	2^6	+ 64
6		2^5	+ 32
5		2^4	+ 16
4		2^3	+ 8
3		2^2	+ 4
2		2^1	+ 2
1		2^0	+ 1



Eine Änderung der Schalterstellung wird nur beim Wiedereinschalten des Gerätes übernommen!

3.5 Lieferumfang

DP/CAN-Koppler 700-650-CAN01
inkl. 2 Stk. 3pol. Stecker für CAN-Bus und 24V Versorgung
CD mit GSD-Dateien und Anleitungen

3.6 Zubehör

Handbuch, deutsch/englisch 900-650-CAN01
CAN-Bus Stecker 700-690-0BA11
CAN-Bus Stecker mit Anschlussstecker 700-690-0BB11
CAN-Bus Stecker mit axialem Kabelabgang 700-690-0CA11

4 Projektierung (CANopen® Master)

Der DP/CAN-Koppler ermöglicht Ihnen, Daten aus einem PROFIBUS-DP Kreis in einen sekundären CAN-Kreis zu übertragen und umgekehrt.

Bevor nun Daten zwischen PROFIBUS und CAN-Bus ausgetauscht werden können, muss in der Hochlaufphase des PROFIBUSes der DP/CAN-Koppler parametrierung werden. Dieses wird durch den Profibus Master (SPS) automatisch beim Busanlauf durchgeführt.

Der Aufbau des CANopen® Busses (Baudrate, Anzahl der Nodes, Verteilung der PDOs) muss bei der Definition des DP/CAN-Kopplers als PROFIBUSTeilnehmer im Parametersatz des DP/CAN-Kopplers hinterlegt werden.

Nach erfolgreicher Parametrierung und nach dem Start der SPS 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.

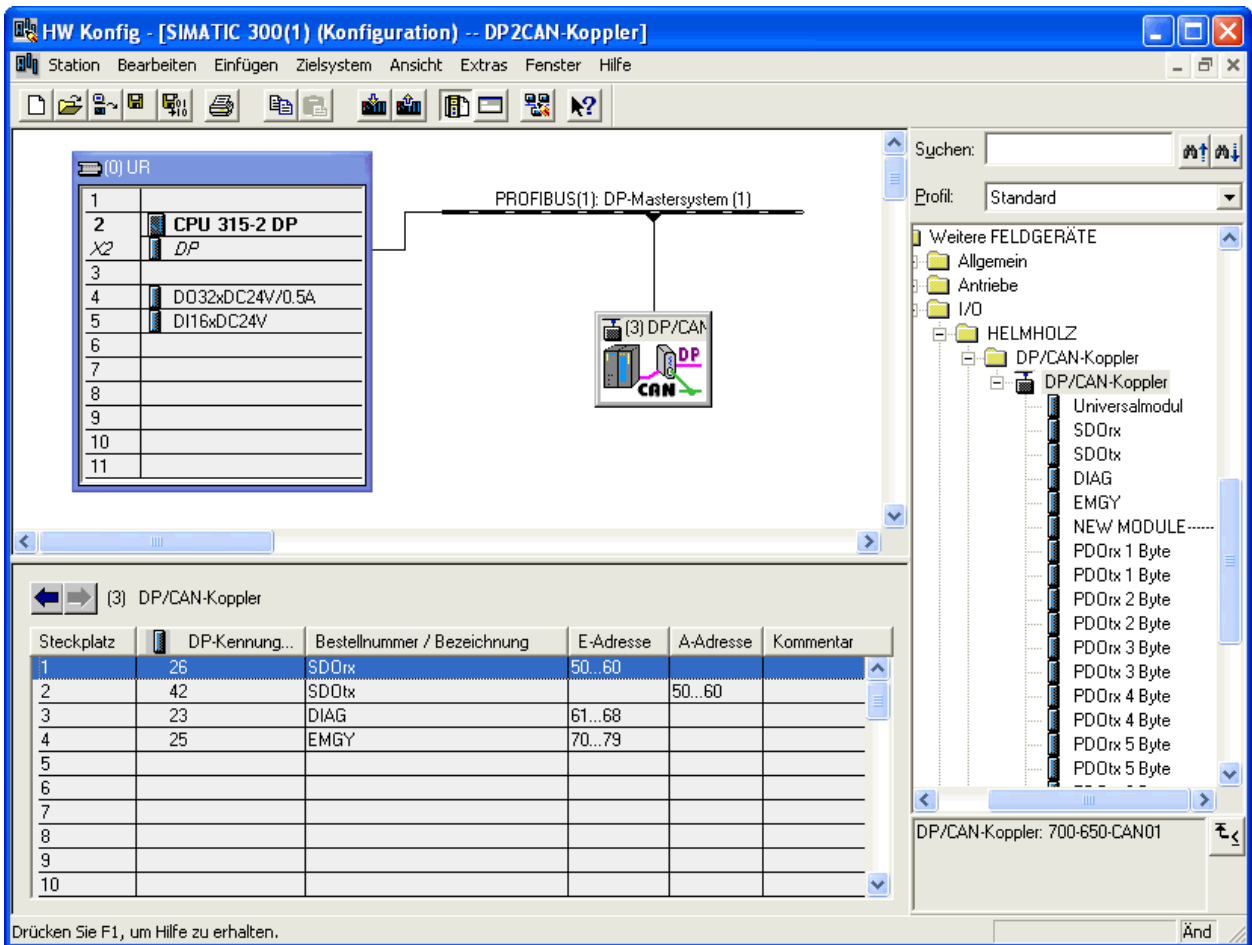
Über ein SDO-Fenster im Prozessabbild können CANopen®-Module mittels SDO-Telegramme parametrierung werden. Die Ein- und Ausgabedaten werden zusätzlich noch bei jedem projektierten Guardingzyklus aktualisiert (Anforderung der PDOs).

4.1 Gerät installieren und parametrierung

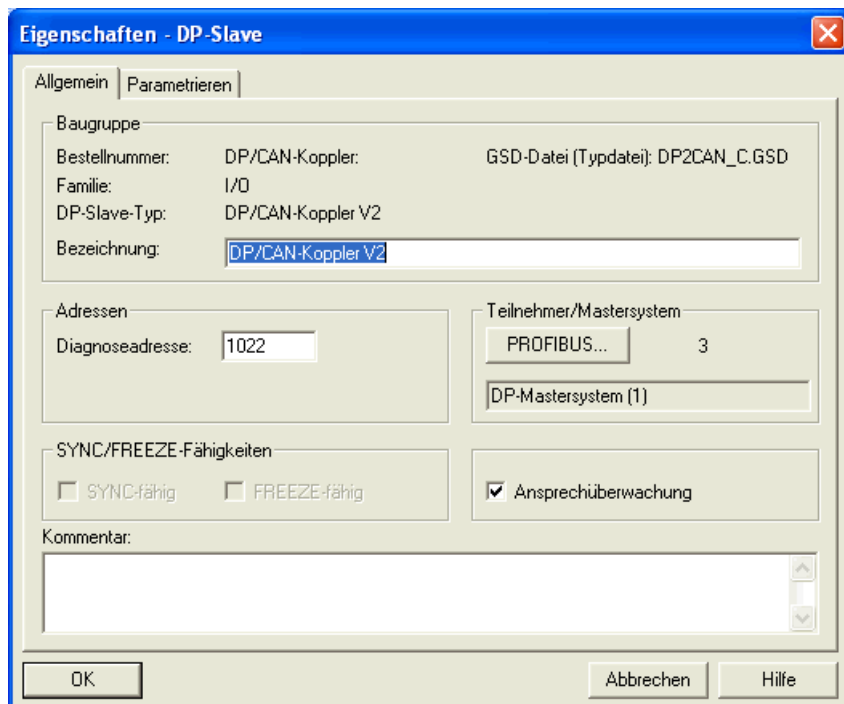
Um den DP/CAN-Koppler im Hardware-Konfigurator verwenden zu können, muss die mitgelieferte GSD-Datei „DP2CAN_M.GSD“ installiert werden. Dieses ist im Hardware-Konfigurator unter dem Menüpunkt „Extras / GSD Dateien installieren“ möglich.

Danach ist der DP/CAN-Koppler im Hardware-Katalog unter ‘Weitere FELDDGERÄTE / IO / Helmholz’ zu finden.

Per Drag & Drop ist nun der „DP/CAN-Koppler CO V2M“ auf einem bereits angelegten PROFIBUS-Netz zu platzieren.



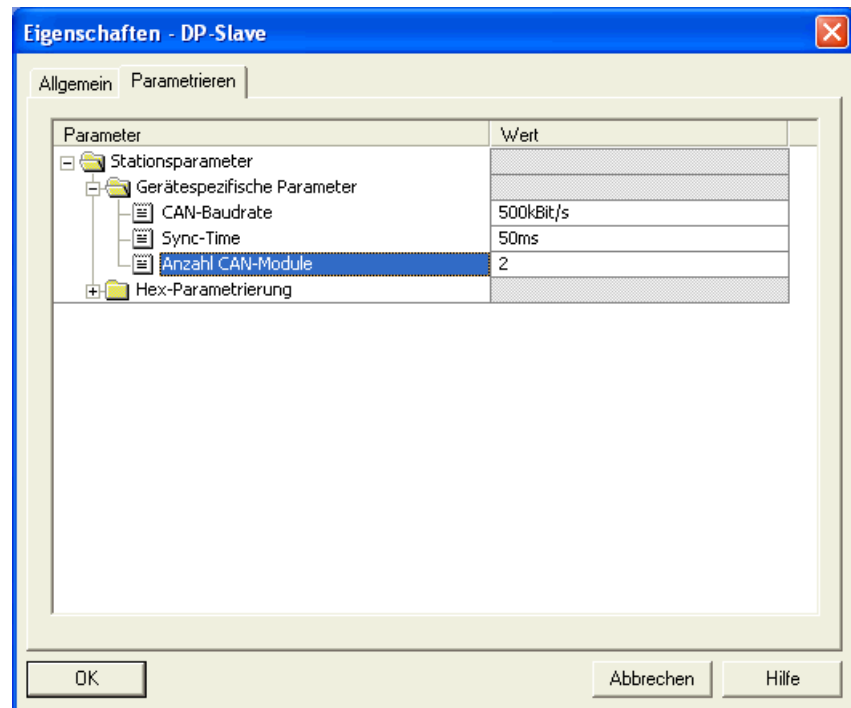
Dem Slave muss dann noch die gewünschte Stationsadresse zugewiesen werden.



Mit dem Parametriertelegramm wird der DP/CAN-Koppler vom Master beim Anlauf mit den notwendigen Informationen über den Aufbau des CANopen® Bus versorgt.

Folgende CAN-Parameter werden hier festgelegt:

- CAN-Baudrate
- Sendezeit für das SYNC Telegramm
- Anzahl der verwendeten CANopen®-Knoten im CAN-Kreis



CAN-Baudrate:

Mögliche Baudraten: 1 MBit/s, 500 KBit/s, 250 KBit/s, 125 KBit/s, 100 KBit/s, 50 KBit/s, 20 KBit/s, 10 KBit/s

Sync-Time:

Hier wird die Zeit für das Senden eines Sync Telegrammes (COB-ID: 80) auf dem CAN-Bus eingestellt. Bereich: 10 ms bis 1000 ms

Anzahl CAN-Module:

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



Die Anzahl der CAN-Module muss mit der Anzahl der definierten „NEW MODULE-----“ Einträge übereinstimmen!

4.2 Festlegung des EA-Adressbereiches in der SPS

Nachdem die Grundparameter des CAN-Busses festgelegt worden sind, müssen alle Datenelemente in den E/A-Bereich der CPU eingebündelt werden.

Die Elemente „SDOrx“, „SDOtx“, „DIAG“, „EMGY“ müssen immer am Anfang der Liste stehen. Die E/A-Adressen können frei gewählt werden.

The screenshot shows the HW Config interface for a SIMATIC 300(1) DP2CAN coupler. The main window displays a rack of modules (UR) and a DP/CAN coupler connected to a PROFIBUS DP-Mastersystem. Below, a table lists the DP/CAN coupler modules with their addresses and data types.

Steckplatz	DP-Kennung...	Bestellnummer / Bezeichnung	E-Adresse	A-Adresse	Kommentar
1	26	SDOrx	50...60		
2	42	SDOtx		50...60	
3	23	DIAG	61...68		
4	25	EMGY	70...79		
5	0	NEW MODULE-----			
6	8DA	PDOtx 1 Byte		80	
7	8DE	PDOrx 1 Byte	80		
8					
9					
10					

4.3 Konsistente Daten

i
Alle CAN-Nachrichten sind als konsistente Datenbereiche im PROFIBUS definiert.

Alle Datenelemente sind als konsistente Daten definiert, somit kann es innerhalb der SDO bzw. PDO-Daten nicht zu Inkonsistenzen kommen. Die Adressen der Datenelemente können im zyklischen Prozessabbild liegen oder außerhalb des zyklischen Prozessabbildes. Wenn die Daten außerhalb des zyklischen Prozessabbildes liegen, so muss mit den Peripheriezugriffsbefehlen „L PEx“ oder „T PAX“ zugegriffen werden.

Wenn 3, 5 oder mehr als 5 Bytes konsistent übertragen werden sollen, so müssen für die Aktualisierung der Daten bei älteren CPUs die SFC 14 „DPRD_DAT - konsistente Daten lesen“ und SFC 15 „DPWR_DAT - konsistente Daten schreiben“ verwendet werden.

Die Bausteine sind im Siemens Handbuch „System- und Standardfunktionen für S7-300/400“, Band 1, Kapitel 16 beschrieben.

4.4 Maximale Parametergrößen und Adressbereiche

Es können maximal 84 Steckplätze (Elemente) definiert werden. Die maximale Anzahl an belegbaren Eingabe- oder Ausgabeadressen sind jeweils 240 Bytes aber zusammen max. 312 Bytes.

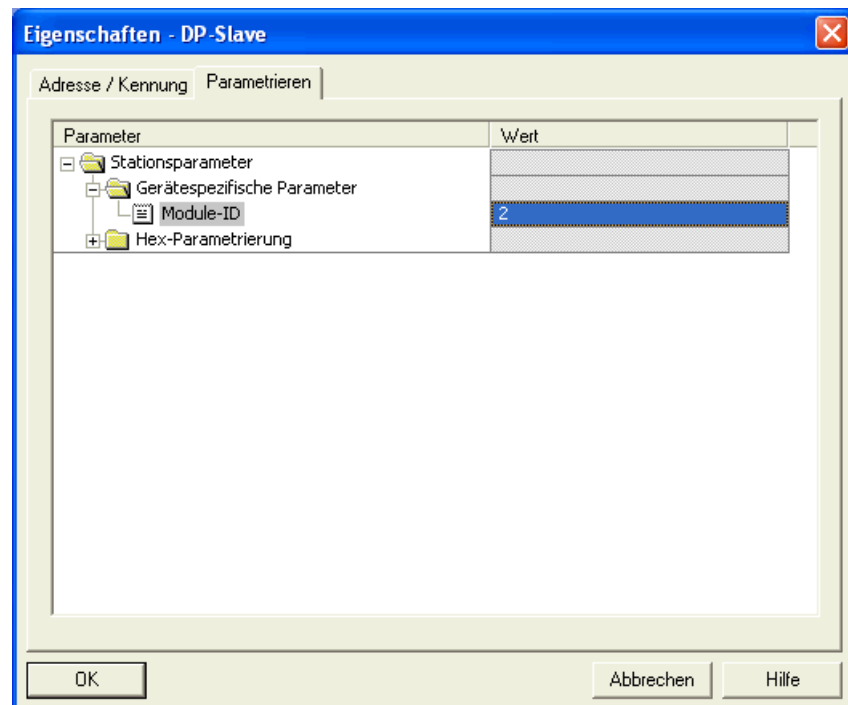
Die Parameter der Elemente dürfen maximal 84 Bytes belegen. Die Grundparameter belegen bereits 4 Bytes, jeder „NEW MODULE“ oder PDO Eintrag belegt ein weiteres Byte.



Beachten Sie die maximalen Parametergrößen!

4.5 Parametrieren der Module

Auf den weiteren Steckplätzen des DP/CAN-Kopplers können nun die CAN-Module und die zum Modul gehörigen PDOs festgelegt werden. Eintrag „NEW MODULE----“.




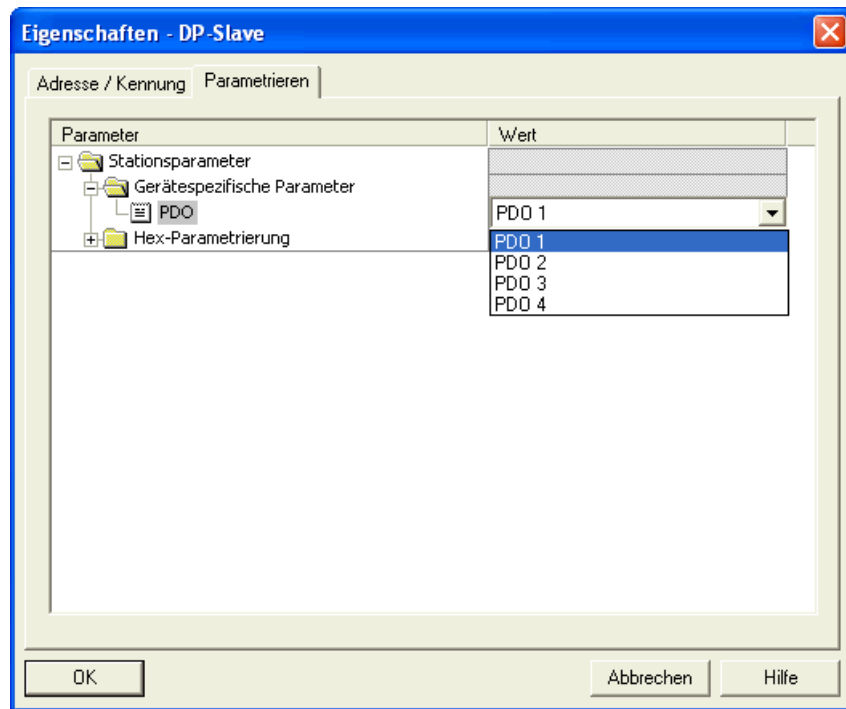
Inkorrekte Anzahl der Moduldefinitionen führen zu Datenverfälschungen oder Busanlauf Fehlern!

Die Anzahl der Module („NEW MODULE“ Einträge) muss unbedingt mit dem in der Geräteparametrierung festgelegten Anzahl übereinstimmen.

Nach jedem „NEW MODULE“ Eintrag folgen die PDO-Elemente. Die PDO-Datenelemente gibt es in 1 Byte bis zu 8 Byte Breite als lesendes („PDOrx“) oder schreibendes PDO („PDOtx“).

Für jedes PDO-Element kann die PDO Nummer gewählt werden.


Als Übertragungsarten werden bei den PDOs nur Event und Sync unterstützt.



Die E/A-Adressen können frei vergeben werden, alle PDO-Elemente sind als konsistente Daten ausgelegt (s. a. Kapitel 4.3).

5 Programmierung (CANopen® Master)

5.1 Datenaustausch

Nach dem der Master im Abschluss der Hochlaufphase die fehlerfreie Parametrierung und Konfigurierung erkannt hat und die SPS gestartet worden ist, sendet er Datenaustauschtelegramme. Dazu sendet der PROFIBUS-Master zyklisch alle Daten der parametrierten Sendeidentifizier an den DP/CAN-Koppler.



Alle CAN-Nachrichten sind als konsistente Datenbereiche im PROFIBUS definiert.

Jede Datenänderung (auf CANopen®-Seite oder DP-Seite) wird sofort beim nächsten Zyklus übertragen. Zusätzlich werden die Eingabedaten der CANopen®-Module pro Guardingzyklus aktualisiert, falls diese nicht schon vorher übertragen worden sind.

5.2 Aufstartvorgang CAN-Netzwerk

Das Aufstarten wird durch einen Stop → Run Zustandsübergang der SPS ausgelöst. Hierbei werden wichtige System SDOs gelesen bzw. geschrieben (PDO-, Nodeguarding- oder Heartbeat Einstellungen, etc.). Ein CAN-Slave wird immer aufgestartet, sobald er projektiert ist und an der SPS ein Stop → Run Zustandsübergang zuvor erfolgte.



Der DP/CAN-Koppler braucht nach einem Netzaus zum Neuanlauf immer einen SPS Stop → Run Übergang!

Meldet sich ein CAN-Modul nach einem Ausfall erneut an, so wird es automatisch parametriert und wieder in den zyklischen Betrieb eingebunden.

5.3 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.

Diagnosebereich:

Byte	Bit	Funktion
0	0	Eintrag 1 im Prozessabbild ungültig
	1	Eintrag 2 im Prozessabbild ungültig
	2	Eintrag 3 im Prozessabbild ungültig
	3	Eintrag 4 im Prozessabbild ungültig
	4	Eintrag 5 im Prozessabbild ungültig
	5	Eintrag 6 im Prozessabbild ungültig
	6	Eintrag 7 im Prozessabbild ungültig
	7	Eintrag 8 im Prozessabbild ungültig
...	...	
5	0	Eintrag 41 im Prozessabbild ungültig
	1	Eintrag 42 im Prozessabbild ungültig
	2	Eintrag 43 im Prozessabbild ungültig
	3	Eintrag 44 im Prozessabbild ungültig
	4	Eintrag 45 im Prozessabbild ungültig
	5	Eintrag 46 im Prozessabbild ungültig
	6	Eintrag 47 im Prozessabbild ungültig
	7	Eintrag 48 im Prozessabbild ungültig
6	0	<i>Reserviert</i>
	1	CAN-Modul 1 ist nicht operational
	2	CAN-Modul 2 ist nicht operational
	3	CAN-Modul 3 ist nicht operational
	4	CAN-Modul 4 ist nicht operational
	5	CAN-Modul 5 ist nicht operational
	6	CAN-Modul 6 ist nicht operational
	7	CAN-Modul 7 ist nicht operational
7	0	CAN-Modul 8 ist nicht operational
	1	CAN-Modul 9 ist nicht operational
	2	CAN-Modul 10 ist nicht operational
	3	CAN-Modul 11 ist nicht operational
	4	CAN-Modul 12 ist nicht operational
	5	CAN-Modul 13 ist nicht operational
	6	CAN-Modul 14 ist nicht operational
	7	CAN-Modul 15 ist nicht operational

Das Prozessabbild ist ungültig wenn:

- das zugehörige CAN-Modul nicht operational ist
- das Bit 31 von der COB ID des entsprechenden PDOs gesetzt ist (Zustand high besagt, dass das PDO inaktiv ist)

Das CAN-Modul ist nicht operational wenn:

- der Status im Heartbeat/Node Guarding nicht 0x05 ist
- das Heartbeat/Node Garding Frame vom Slave ausbleibt
- alle PDOs, die vom Slave mit dem Transmission Type 1-240 gesendet werden (wenn welche vorhanden sind), ausbleiben.



Die COB-ID des Emergency-Telegrammes ist nach CANopen® Standard vorgegeben. Änderungen der COB-ID in den Slaves werden vom DP/CAN-Koppler nicht unterstützt!

5.4 Emergency Telegramme empfangen

In dem Emergency-Empfangsfach werden die EMERGENCY-Telegramme vom CAN-Bus kommend eingeblendet. Das erste vom CAN-Bus kommende EMERGENCY-Telegramm wird direkt in das Emergency-Empfangsfach kopiert, alle weiteren EMERGENCY-Telegramme werden in einem Ringpuffer im DP/CAN-Koppler zwischengespeichert und erst dann in das Emergencyempfangsfach geschrieben, wenn das vorhergehende EMERGENCY-Telegramm vom Anwender gelesen wurde. Es ist aus diesem Grunde ein Handshaking notwendig.

5.4.1 Emergency Empfangsfach

Das Emergency-Empfangsfach hat folgenden Aufbau:

Byte	Funktion
0	CAN Modul Node-ID
1	Datenlänge des Emcy-Telegrammes (0-8 Bytes)
2	Datenbyte 1
3	Datenbyte 2
4	Datenbyte 3
5	Datenbyte 4
6	Datenbyte 5
7	Datenbyte 6
8	Datenbyte 7
9	Datenbyte 8

5.4.2 Handshaking für Emergency-Telegramme

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

Bit 4 (im Byte 0 Empfangs-SDO) gesetzt → neues Emergency-Telegramm im Emergencybereich.

Bit 5 (im Byte 0 Empfangs-SDO) gesetzt → Daten im Emergencybereich sind gültig.

Bit 6 (im Byte 0 Sende-SDO) gesetzt → Daten sind vom SPS-Programm verarbeitet worden.

Ablauf:

1. Wenn Bit 4 = 0 und Bit 6 = 0 und neue EMERGENCY-Telegramme empfangen wurden, wird das erste EMERGENCY-Telegramm in den Emergency-Bereich geschrieben.
2. Bit 4 und Bit 5 werden gesetzt
3. Wenn die SPS das Emergency-Telegramm verarbeitet hat, muss Bit 6 = 1 gesetzt werden, um das Telegramm zu quittieren
4. Wenn Bit 6 = 1, löscht der DP/CAN-Koppler den Emergency-Bereich und löscht Bit 4 und Bit 5
5. Die SPS wartet auf gelöschte Bits 4 und Bit 5 und löscht dann Bit 6

5.5 CAN-Module parametrieren (SDO-Transfer)

Die SDO-Kommunikation (SDO = Service Data Object) ist ein bestätigter bzw. quittierter Dienst des CANopen[®]-Protokolls. Jede SDO-Anfrage (Request) wird mit einer Antwort (Response) vom angesprochenen Module quittiert. Sollte das Module nicht antworten, so wird vom DP/CAN-Koppler eine Timeoutmeldung an die SPS gesendet.

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 (Process Data Objects) abgewickelt wird, nicht geeignet.

Die gesamte SDO-Kommunikation wird von dem DP/CAN-Koppler über ein 11 Byte großes Fenster im Prozessabbild abgewickelt.

5.5.1 Expedited SDO-Transfers (bis zu 4 Bytes Daten)

Neue Aufträge werden in den E/A-Bereich von **SDOtx** geschrieben. Die Antworten können aus dem Bereich **SDOrx** gelesen werden.

Es kann immer nur ein Auftrag auf einmal abgearbeitet werden. Nach jedem Auftrag muss auf eine Antwort gewartet werden!



Es wird zur Zeit nur der Expedited SDO-Transfer mit bis zu 4 Bytes unterstützt.

Eine Unterstützung von SDO-Transfers mit mehr als 4 Bytes ist auf Anfrage erhältlich.

CAL WRITE REQ (SDO Write senden):

SDOtx	Funktion
0	Handshake Byte
1	Telegrammtyp: 36
2	Node-ID (1..127)
3	SDO Index (High Byte)
4	SDO Index (Low Byte)
5	SDO Subindex
6	Anzahl gültige Datenbyte (SDO-Größe, 1-4)
7	Datenbyte 1
8	Datenbyte 2
9	Datenbyte 3
10	Datenbyte 4

CAL READ REQ (SDO Read senden):

SDOtx	Funktion
0	Handshake Byte
1	Telegrammtyp: 39
2	Node-ID (1..127)
3	SDO Index (High Byte)
4	SDO Index (Low Byte)
5	SDO Subindex



Nach jeder Anforderung muss die Antwort des Slaves abgewartet werden! SDO-Transfers können nur nacheinander abgearbeitet werden!

CAL READ CNF P (SDO Read Response):

SDOrx	Funktion
0	Handshake Byte
1	Telegrammtyp: 40
2	Node-ID (1..127)
3	Datenlänge 4-7 Bytes
4	SDO Index (High Byte)
5	SDO Index (Low Byte)
6	SDO Subindex
7	Datenbyte 1
8	Datenbyte 2
9	Datenbyte 3
10	Datenbyte 4

CAL READ CNF N (SDO Read negative Response):

SDOrx	Funktion
0	Handshake Byte
1	Telegrammtyp: 41
2	Node-ID (1..127)
3	Datenlänge 7 Bytes
4	SDO Index (High Byte)
5	SDO Index (Low Byte)
6	SDO Subindex
7	Error_Class
8	Error_Class
9	Additional_Class (High Byte)
10	Additional_Class (Low Byte)

CAL WRITE CNF P (SDO Write Response):

SDOrx	Funktion
0	Handshake Byte
1	Telegrammtyp: 37
2	Node-ID (1..127)
3	Datenlänge 3 Bytes
4	SDO Index (High Byte)
5	SDO Index (Low Byte)
6	SDO Subindex

CAL WRITE CNF N (SDO Write negative Response):

SDOrx	Funktion
0	Handshake Byte
1	Telegrammtyp: 38
2	Node-ID (1..127)
3	Datenlänge 7 Bytes
4	SDO Index (High Byte)
5	SDO Index (Low Byte)
6	SDO Subindex
7	Error_Class
8	Error_Class
9	Additional_Class (High Byte)
10	Additional_Class (Low Byte)

5.5.2 SDO-Timeout

Dieses Telegramm erscheint im Empfangsbereich (SDOrx) wenn auf eine SDO-Anfrage nach 700 ms keine Antwort vom Slave gekommen ist.

SDOrx	Funktion
0	Handshake Byte
1	Telegrammtyp: 240
2	Node-ID (1..127)

5.5.3 Handshaking SDOtx (Sende-SDO)

Das Handshaking zum Versenden von SDO-Aufträgen wird im ersten Byte des **SDOtx** Bereiches durchgeführt. Das SPS Programm setzt und löscht diese Bits.

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

Bit 0 gelöscht: Daten sind ungültig, warte auf nächstes SDO-Telegramm. Zwischen zwei SDO-Telegrammen muss dieses Bit auf 0 gesetzt werden.

5.5.4 Handshaking SDOrx (Empfangs-SDO)

Das Handshaking zum Empfangen von SDO-Antworten auf laufende SDO-Aufträge wird im ersten Byte des **SDOrx** Bereiches durchgeführt. Das SPS Programm wertet diese Bits aus.

Bit 0 gesetzt: SDO-Transfer ist aktiv

Bit 1 gesetzt: gültige Daten im Empfangspuffer

Bit 2 gesetzt: SDO-Telegramm wurde gesendet

Wenn das Bit 1 gesetzt wurde, kann das Empfangstelegramm verarbeitet werden. Danach muss das Bit 0 vom SDOtx Handshaking Byte wieder zurückgesetzt werden, damit ein neuer Auftrag gestartet werden kann.

6 Projektierung (CAN Layer 2)

Im Layer 2 Modus kann der DP/CAN-Koppler beliebige CAN-Nachrichten (CAN 2.0A, 11Bit) versenden und empfangen werden.

Es werden grundsätzlich zwei Übertragungsarten unterschieden. Bei den Empfangs und Sendeobjekten wird der Identifier und die Größe der CAN-Nachricht fest parametrierbar und nur die Daten werden über den PROFIBUS übertragen. Jedes Sende- bzw. Empfangsobjekt entspricht somit einem CAN-Telegramm.

Zusätzlich gibt es noch ein Receive Objekt und ein Transmit Objekt. Das Receive Objekt kann mehrere Nachrichten gefiltert durch eine parametrierbare Akzeptanzmaske empfangen. Hierbei wird neben den Daten auch noch der Identifier und die Länge des CAN-Telegramms über den PROFIBUS an die SPS Applikation übertragen.

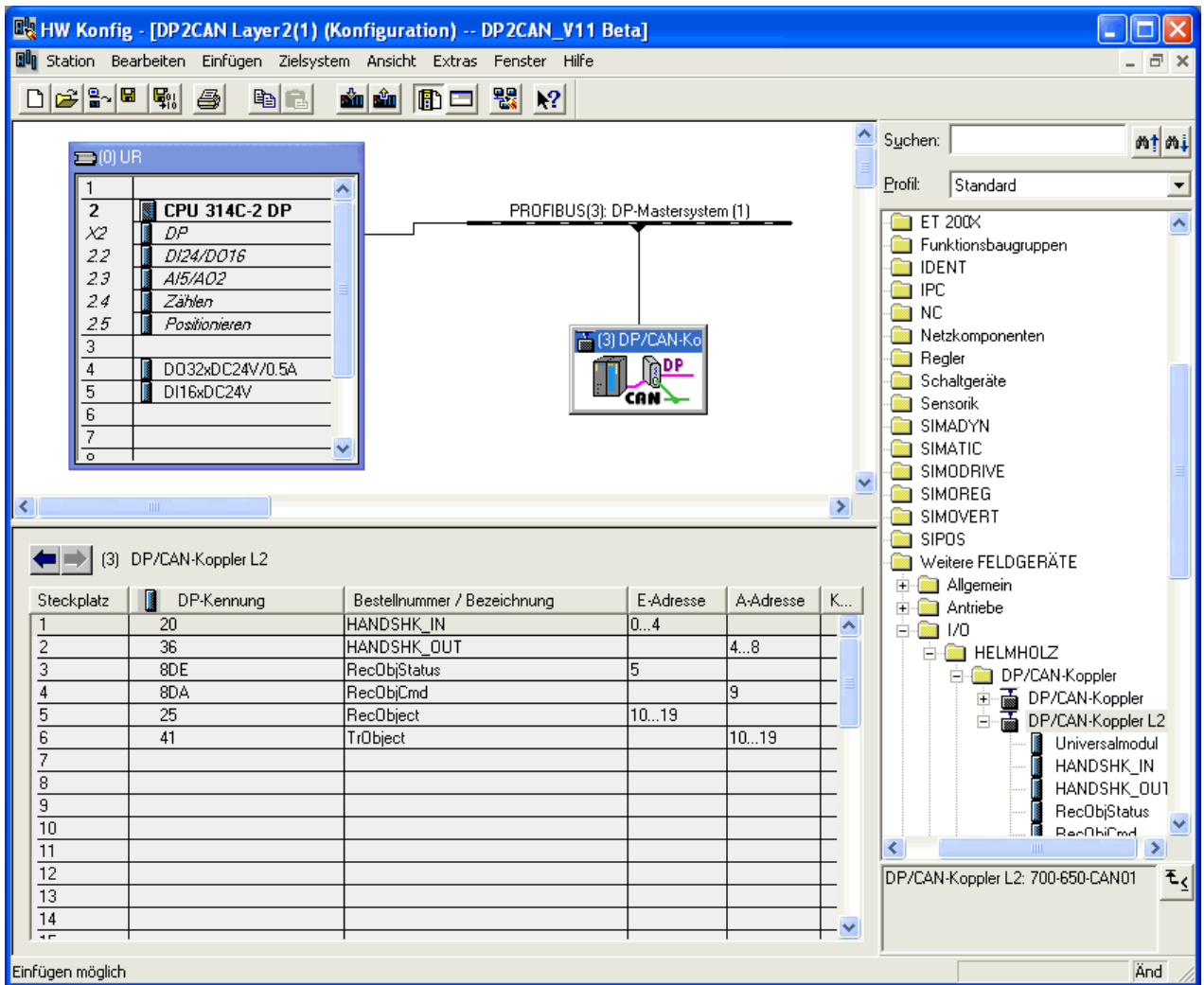
Mit dem Transmit Objekt können beliebige CAN-Nachrichten versendet werden. Zusätzlich kann das Transmit Objekt die Nachricht auch zyklisch mit einer fest eingestellten Zeit versenden.

6.1 Gerät installieren und parametrieren

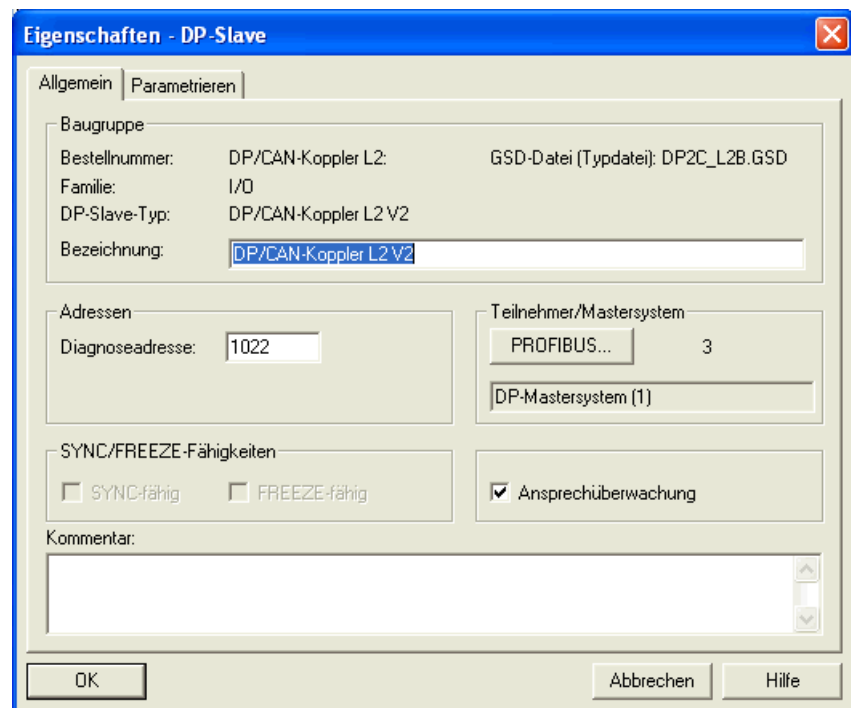
Um den DP/CAN-Koppler mit Layer 2 im Hardware-Konfigurator verwenden zu können, muss die mitgelieferte GSD-Datei „DP2C_L2h.GSD“ installiert werden. Dieses ist im Hardware-Konfigurator unter dem Menüpunkt „Extras / GSD Dateien installieren“ möglich.

Danach ist der DP/CAN-Koppler im Hardware-Katalog unter 'Weitere FELDDGERÄTE / IO / Helmholz' zu finden.

Per Drag & Drop ist nun der „DP/CAN-Koppler L2 V2h“ auf einem bereits angelegtem PROFIBUS-Netz zu platzieren.



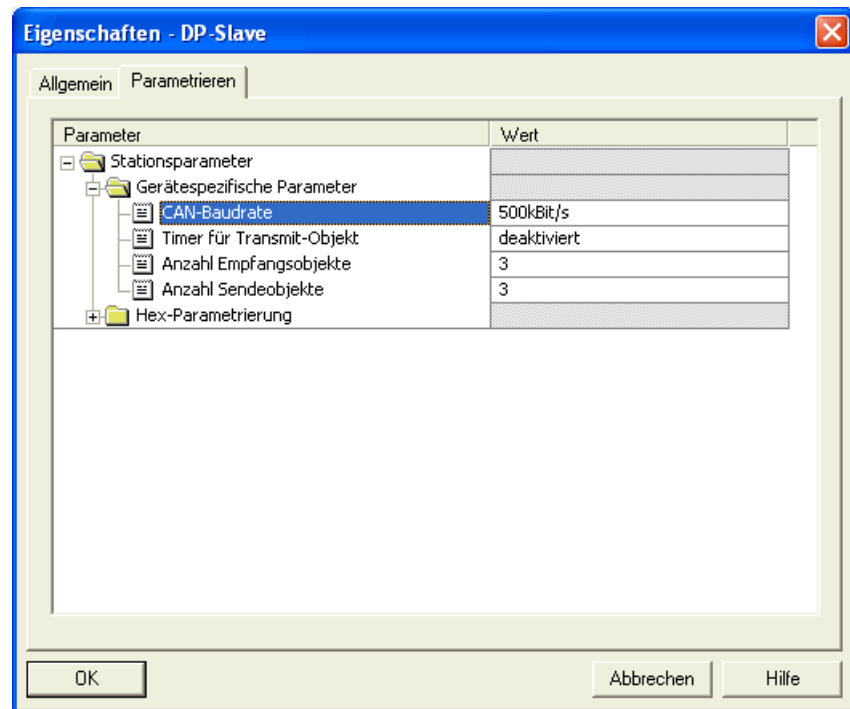
Dem Slave kann hier die gewünschte Stationsadresse zugewiesen werden.



Mit dem Parametriertelegramm wird der DP/CAN-Koppler vom Master beim Anlauf mit den notwendigen Informationen über die zu verarbeitenden CAN-Telegramme versorgt.

Folgende CAN-Parameter werden hier festgelegt:

- CAN-Baudrate
- Sendezeit für das zyklische Versenden des Transmit Objekts (falls gewünscht)
- Anzahl der verwendeten Empfangs- und Sendenachrichten



CAN-Baudrate:

Mögliche Baudraten: 1 MBit/s, 500 KBit/s, 250 KBit/s, 125 KBit/s, 100 KBit/s, 50 KBit/s, 20 KBit/s, 10 KBit/s

Timer für Transmit Objekt:

Hier wird die Zeit für das zyklische Senden des Transmit Objektes eingestellt (10 ms bis 1000 ms). Mit der Option „Deaktiviert“ wird das Transmit Objekt immer sofort und nur einmal versendet.

Anzahl Empfangsobjekte:

Die Anzahl der verwendeten Empfangs Objekte (DI).

Anzahl Sendeobjekte:

Die Anzahl der verwendeten Sendeobjekte (DO).



Inkorrekte Angabe der Anzahl der Empfangs-/Sendeobjekte führen zu Datenverfälschungen oder Busanlauffehlern!

6.2 Festlegung des EA-Adressbereiches in der SPS

Nachdem die Grundparameter des CAN-Busses festgelegt worden sind, müssen alle Datenelemente in den E/A-Bereich der SPS eingeblendet werden.

Die ersten 6 Elemente müssen immer am Anfang der Liste in der festgelegten Reihenfolge stehen. Die E/A-Adressen können frei gewählt werden.

Die „DI“ Elemente müssen vor den „DO“ Elementen definiert werden!

The screenshot shows the HW Konfig software interface for configuring a DP/CAN-Koppler L2. The main window displays a hardware rack configuration on the left, a network diagram in the center, and a detailed data point table at the bottom. The data point table is as follows:

Steckplatz	DP-Kennung	Bestellnummer / Bezeichnung	E-Adresse	A-Adresse	K...
1	20	HANDSHK_IN	50...54		
2	36	HANDSHK_OUT		50...54	
3	8DE	RecObjStatus	59		
4	8DA	RecObjCmd		59	
5	25	RecObject	60...69		
6	41	TrObject		70...79	
7	16DE	DI 2 Byte	80...81		
8	16DE	DI 2 Byte	82...83		
9	16DE	DI 2 Byte	84...85		
10	16DA	DO 2 Byte		80...81	
11	16DA	DO 2 Byte		82...83	
12	16DA	DO 2 Byte		84...85	
13					
14					

The right-hand side of the interface shows a tree view of the configuration structure, including folders for 'Allgemein', 'Antriebe', 'I/O', and 'HELMHOLZ', with a sub-tree for 'DP/CAN-Koppler L2' containing various data point types like 'DI 1 Byte' through 'DI 8 Byte' and 'DO 1 Byte' through 'DO 6 Byte'.



Alle CAN-Nachrichten sind als konsistente Datenbereiche im PROFIBUS definiert.

6.3 Konsistente Daten

Alle Datenelemente sind als konsistente Daten definiert, somit kann es innerhalb der DI/DO- und RecObject/TrObject-Daten nicht zu Inkonsistenzen kommen. Die Adressen der Datenelemente können im zyklischen Prozessabbild liegen oder außerhalb des zyklischen Prozessabbildes. Wenn die Daten außerhalb des zyklischen Prozessabbildes liegen, so muss mit den Peripheriezugriffsbefehlen „L PEx“ oder „T PAX“ zugegriffen werden.

Wenn 3, 5 oder mehr als 5 Bytes konsistent übertragen werden sollen, so müssen für die Aktualisierung der Daten bei älteren CPUs die SFC 14 „DPRD_DAT - konsistente Daten lesen“ und SFC 15 „DPWR_DAT – konsistente Daten schreiben“ verwendet werden.

Die Bausteine sind im Siemens Handbuch „System- und Standardfunktionen für S7-300/400“, Band 1, Kapitel 16 beschrieben.

6.4 Maximale Parametergrößen und Adressbereiche

Es können maximal 44 Steckplätze (Elemente) definiert werden. Davon sind 6 Elemente bereits fest belegt. Die maximale Anzahl an belegbaren Eingabe- oder Ausgabeadressen sind jeweils 240 Bytes aber zusammen max. 312 Bytes.

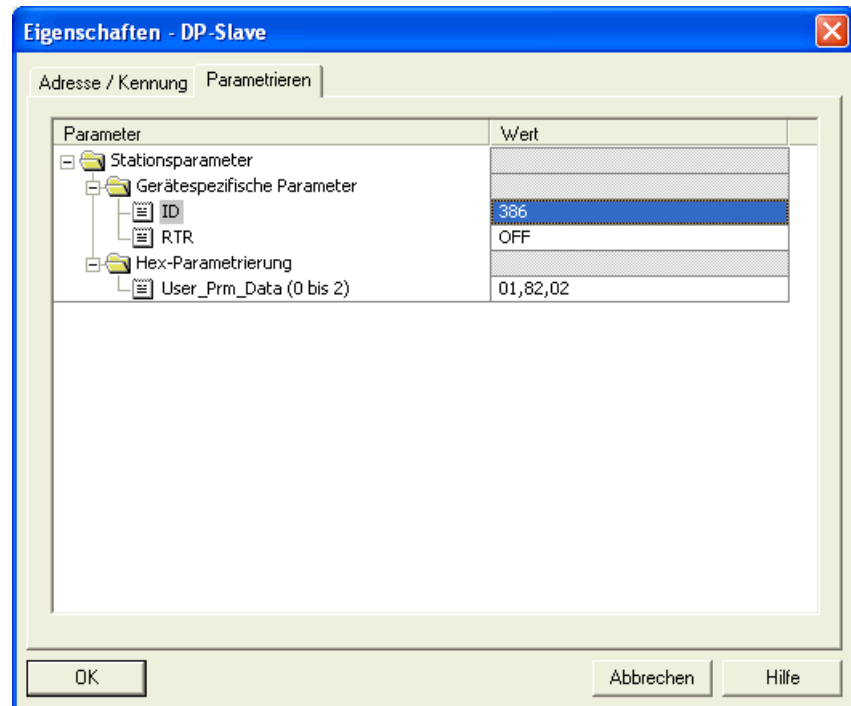
Die Parameter der Elemente dürfen maximal 120 Bytes belegen. Die Grundparameter belegen bereits 7 Bytes, jeder weitere „DI“ oder „DO“ Eintrag belegt 3 weitere Bytes.



Beachten Sie die maximalen Parametergrößen!

6.5 Sende und Empfangsnachrichten parametrieren

Auf den weiteren Steckplätzen des DP/CAN-Kopplers können nun die Sende- und Empfangsnachrichten („DI x Byte“, „DO x Byte“) definiert werden.



Jedes Sende- oder Empfangsobjekt kann für genau eine CAN-Nachricht festgelegt werden. Die Länge der CAN-Nachricht entspricht der Größe des DI-/DO-Objektes. Die CAN-ID wird im Parametersatz festgelegt.

6.6 Receive Objekt parametrieren

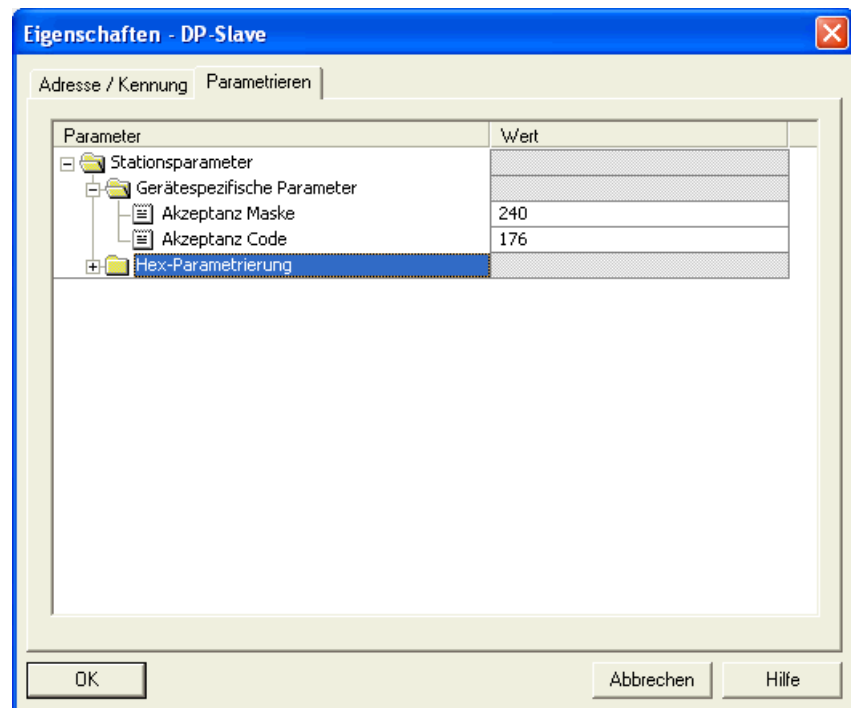
Das Receive Objekt muss ebenfalls parametrieren werden, wenn es verwendet werden soll.

Um beliebige CAN-Telegramme mit dem Receive Objekt zu empfangen werden die oberen 8 Bit des CAN-Identifiers erst mit einer Maske gefiltert (Akzeptanzmaske) und dann mit einem vorgegebenen Wert verglichen (AkzeptanzCode). Wenn dieser Vergleich positiv ist, wird das CAN-Telegramm im Receive FIFO eingetragen und der SPS zur Verfügung gestellt.

10	9	8	7	6	5	4	3	2	1	0
Empfangener CAN-Identifizier										
AkzeptanzMaske (z.B. 11110000)								<i>nicht berücksichtigt</i>		
AkzeptanzCode (z.B. 1011xxxx)										

Die Akzeptanzmaske 11110000 (= 240) filtert die oberen 4 Bits des CAN-Identifiers heraus. Der Akzeptanzcode 1011xxxx (= 176) legt fest, welche Telegramme nach der Filterung angenommen werden sollen.

In diesem Beispiel ergeben sich die CAN-Telegramme mit den Identifiern 0x580 bis 0x5FF.



Mit dem Wert „0“ für die Akzeptanz Maske und den Akzeptanz Code werden alle nicht in einem DI Objekt projizierten CAN-Telegramme zum Receive Objekt zugeordnet (keine Filterung).

7 Programmierung (CAN Layer 2)

7.1 Datenaustausch

Nach dem der Master im Abschluss der Hochlaufphase die fehlerfreie Parametrierung und Konfigurierung erkannt hat und die SPS gestartet worden ist, kann der DP/CAN-Koppler Telegrammes über CAN empfangen bzw. senden.

7.2 Handshake Bits

Die 5 Bytes des HANDSHK_IN Bereiches signalisieren den Empfang von CAN-Nachrichten über das Receive Objekt und die DI-Objekte. Die Bits werden bei Empfang einer neuen Nachricht immer invertiert (Toggle-Bit).

Byte	Bit	Funktion
0	0	Neues Telegramm ReceiveObjekt
	1	Neues Telegramm EmpfangsObjekt 1
	2	Neues Telegramm EmpfangsObjekt 2
	3	Neues Telegramm EmpfangsObjekt 3
	4	Neues Telegramm EmpfangsObjekt 4
	5	Neues Telegramm EmpfangsObjekt 5
	6	Neues Telegramm EmpfangsObjekt 6
	7	Neues Telegramm EmpfangsObjekt 7
...	...	
4	0	Neues Telegramm EmpfangsObjekt 32
	1	Neues Telegramm EmpfangsObjekt 33
	2	Neues Telegramm EmpfangsObjekt 34
	3	Neues Telegramm EmpfangsObjekt 35
	4	Neues Telegramm EmpfangsObjekt 36
	5	Neues Telegramm EmpfangsObjekt 37
	6	Neues Telegramm EmpfangsObjekt 38
	7	Neues Telegramm EmpfangsObjekt 39

Bei der Interpretation des Receive Objektes sind auch die Objekte „RecObjStatus“ und „RecObjCmd“ zu beachten.

Die 5 Bytes des HANDSHK_OUT Bereiches werden zum Versenden der Sende Objekte und des Transmit Objektes verwendet. Die Bits lösen das Versenden der Nachricht immer dann aus, wenn das Bit invertiert wird (Toggle-Bit).

Byte	Bit	Funktion
0	0	Telegramm für TransmitObjekt versenden
	1	Telegramm für SendeObjekt 1 versenden
	2	Telegramm für SendeObjekt 2 versenden
	3	Telegramm für SendeObjekt 3 versenden
	4	Telegramm für SendeObjekt 4 versenden
	5	Telegramm für SendeObjekt 5 versenden
	6	Telegramm für SendeObjekt 6 versenden
	7	Telegramm für SendeObjekt 7 versenden
...	...	
4	0	Telegramm für SendeObjekt 32 versenden
	1	Telegramm für SendeObjekt 33 versenden
	2	Telegramm für SendeObjekt 34 versenden
	3	Telegramm für SendeObjekt 35 versenden
	4	Telegramm für SendeObjekt 36 versenden
	5	Telegramm für SendeObjekt 37 versenden
	6	Telegramm für SendeObjekt 38 versenden
	7	Telegramm für SendeObjekt 39 versenden

7.3 Empfangs- und Sende-Objekte

Die Empfangs-Objekte („DI x Bytes“) enthalten je nach der parametrisierten Größe des Objektes immer die Daten des letzten empfangenen CAN-Telegrammes mit dem entsprechenden CAN-Identifizier. Über die 5 Bytes des HANDSHK_IN Bereiches kann der empfang eines neuen Telegrammes erkannt werden.

Die Sende-Objekte („DO x Bytes“) können mit den Werten für das parametrisierte CAN-Telegramm beschrieben werden. Ändern sich die Werte des Sende Objektes, so wird das Telegramm automatisch versenden.

Soll das Telegramm ohne Änderung der Daten versendet werden, so kann das über die 5 Bytes des HANDSHK_OUT Bereiches ausgelöst werden.

Die Reihenfolge der Objekte ist festgelegt: Es müssen erst alle Empfangsobjekte „DI“ und dann alle Sendeobjekte „DO“ definiert werden.



Alle Empfangs- und Sende-Objekte sind als konsistente Datenbereiche im PROFIBUS definiert.



Die Reihenfolge der Empfangs- und Sende-Objekte ist festgelegt!

7.4 Receive Objekt

Wird ein Telegramm empfangen, welches der parametrierten Akzeptanzmaske des Receive Objektes entspricht, wird das Telegramm in den Bereich des Receive Objektes übertragen.

Byte	Inhalt ReceiveObjekt
0	Len + RTR + HighByte CAN-Identifizier
1	LowByte CAN-Identifizier
2	Datenbyte 1
3	Datenbyte 2
4	Datenbyte 3
5	Datenbyte 4
6	Datenbyte 5
7	Datenbyte 6
8	Datenbyte 7
9	Datenbyte 8

In den ersten beiden Bytes (0+1) ist der gesamte CAN-Nachrichten Kopf hinterlegt:

Byte 0						Byte 1									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Datenlänge				RTR	CAN-Identifizier (11 Bit)										

Der Empfang eines neuen Telegrammes ist an der Invertierung des Bits 0 im Byte 0 des **HANDSHK_IN** Bereiches zu erkennen.

Wenn die Nachricht durch das SPS Programm verarbeitet worden ist, muss diese dem DP/CAN-Koppler quittiert werden. Die Quittierung wird über das **RecObjCmd** Byte abgewickelt.

Auch hier sind alle Bits als toggle-Bits zu verwenden, d.h. wenn das Bit invertiert wird, wird die Funktion ausgeführt.

Bit	Funktion RecieveObjCmd
0	Letztes Recieve Objekt Telegramm quittieren
1	<i>reserviert</i>
2	Lösche RecObjStatus Overflow Fehlerbit
3	Reset Receive Objekt FIFO
4	<i>reserviert</i>
5	<i>reserviert</i>
6	<i>reserviert</i>
7	<i>reserviert</i>

Für das Recieve Objekt ist im DP/CAN-Koppler ein FIFO implementiert, der bis zu 24 Nachrichten aufnehmen kann. Werden mehr Nachrichten empfangen und diese nicht durch das SPS-Programm abgeholt werden, dann werden die ältesten Telegramme verworfen und ein Overflow-Fehler im **RecObjStatus** Byte angezeigt. Die Bits des RecieveObjStatus Bytes sind als Zustandsanzeigen zu verarbeiten.

Bit	Funktion ReceiveObjStatus
0	FIFO Ok
1	Overflow ist bei letztem Receive aufgetreten
2	Overflow Fehlermerker
3	Anzahl der noch im FIFO vorhandenen Telegramme
4	
5	
6	
7	

Das Bit 2 des **RecObjStatus** kann mit dem Bit 2 im **RecObjCmd** gelöscht werden.

7.5 Transmit Objekt

Über das Transmit Objekt (**TrObjekt**) können beliebige Nachrichten versendet werden.

Byte	Inhalt Transmit Objekt
0	Len + RTR + HighByte CAN-Identifizier
1	LowByte CAN-Identifizier
2	Datenbyte 1
3	Datenbyte 2
4	Datenbyte 3
5	Datenbyte 4
6	Datenbyte 5
7	Datenbyte 6
8	Datenbyte 7
9	Datenbyte 8

In den ersten beiden Bytes (0+1) wird der gesamte CAN-Nachrichten Kopf hinterlegt:

Byte 0							Byte 1								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Datenlänge				RTR	CAN-Identifizier (11 Bit)										

Das Versenden der Nachricht wird durch das invertieren des Bits 0 im Byte 0 den **HANDSHK_OUT** Bereiches angestoßen.

7.6 Zyklisches Senden des Transmit Objektes

Das Transmit Objekt kann über die Geräteparameter des DP/CAN-Kopplers über einen Timer zyklisch versendet werden.

Das zyklische Versenden über den Timer wird erst mit dem invertieren des Bits 0 im Byte 0 den **HANDSHK_OUT** Bereiches angestoßen.

Eine Änderung der Sendedaten wird beim zyklischen Senden erst durch invertieren des Bits 0 im Byte 0 den **HANDSHK_OUT** Bereiches aktiviert.

Das zyklische Versenden kann nicht gestoppt werden.



CIA® = CAN in Auto-
mation e.V.,
www.can-cia.org

8 CANopen® Kommunikation

8.1 Allgemein

Das CANopen® Protokoll ist ein Schicht 7-Protokoll (Application Layer), das auf den CAN-Bus (ISO 11898) aufsetzt. Die Schichten 1 & 2 (Physical Layer/Data Link Layer) vom CAN-Bus bleiben unberührt.

Die CANopen® Kommunikationsprofile für die verschiedenen Anwendungen werden von der CIA verwaltet.

Die von der Anwendungsschicht bereitgestellten Dienstelemente ermöglichen die Realisierung einer über das Netzwerk verteilten Applikation. Diese Dienstelemente sind in der „CAN Application Layer (CAL) for Industrial Applications“ beschrieben.

Der 11 Bit Identifier und die 8 Datenbytes eines CAN-Layer 2 Telegrammes bekommen eine feste Bedeutung.

Jedes Gerät in einem CANopen®-Netz hat eine feste Node-ID (Modulnummer, 1-127).

8.2 Objekte

Der Datenaustausch mit einem CANopen® Slave erfolgt entweder über fest definierte Servicedaten-Objekte (SDO) oder über frei konfigurierbare Prozessdaten-Objekte (PDO).

Jeder CANopen® Slave besitzt ein festes Verzeichnis von SDOs, die über eine Objektnummer (16 Bit) und einen Index (8 Bit) angesprochen werden.

Beispiel: Objekt 0x1000 / Index 0 = Device Type, 32Bit Unsigned

SDOs mit 8/16/32 Bit Breite können mit einem CANopen®-Telegramm gelesen und geschrieben werden. SDOs die länger sind, werden über mehrere Telegramme übertragen. Für sehr grosse Datenmengen ist die SDO-Blockübertragung vorgesehen.

SDOs können bearbeitet werden, sobald ein CANopen® Slave betriebsbereit ist. Für die SDOs stehen nur die COB-ID Funktionen „SDO Anforderung“ oder „SDO Antwort“ bereit. Die Objekt-nummer, der Zugriffsmodus und Typ werden in den ersten 4 Bytes des CAN-Telegrammes hinterlegt.

Die letzten 4 Bytes des CAN-Telegrammes enthalten dann den Wert für das SDO.



Jedem CANopen® Slave sollte ein Verzeichnis der von ihm unterstützten Objekte beiliegen.

PDOs enthalten die „Arbeitswerte“ eines CANopen® Slaves für den zyklischen Prozessbetrieb. Jeder CANopen® Slave kann mehrere PDOs verwalten (im Normalfall bis zu 4 zum Senden und 4 zum Empfangen).

Jedes der vorhandenen PDOs hat eine eigene COB-ID. In den 8 Datenbytes des Telegrammes können beliebige Informationen des CANopen® Slaves zum Lesen und Beschreiben „gemapped“ werden. Dieses können sowohl bereits vorhandene SDOs sein, als auch Aktualwerte des Slaves (z.B. Analogwert eines Einganges).

Die PDOs werden von den meisten CANopen® Slaves automatisch beim Anlauf gemapped. Die Zuordnung kann dann über bestimmte SDOs geändert werden.

8.3 Funktionen

Die CANopen® Funktionen teilen sich in drei Grundarten auf:

- SDO Lesen und Schreiben
- PDO Lesen und Schreiben
- Netzmanagement

Der Funktionscode wird in den oberen 4 Bit des Identifiers hinterlegt. Zusammen mit der Node-ID ergeben sie den COB-Identifier.

COB-Identifier (COB-ID):

10	9	8	7	6	5	4	3	2	1	0
Function				Node-ID						

Broadcast-Funktionen:

Funktion	Function code (binary)	Resulting COB-ID
NMT	0000	0h
SYNC	0001	80h
TIME STAMP	0010	100h

Node Funktionen:

Funktion	Function code (binary)	Resulting COB-ID
EMERGENCY	0001	81h – FFh
PDO1 (tx)	0011	181h – 1FFh
PDO1 (rx)	0100	201h – 27Fh
PDO2 (tx)	0101	281h – 2FFh
PDO2 (rx)	0110	301h – 37Fh
PDO3 (tx)	0111	381h – 3FFh
PDO3 (rx)	1000	401h – 47Fh
PDO4 (tx)	1001	481h – 4FFh
PDO4 (rx)	1010	501h – 57Fh
SDO (tx)	1011	581h – 5FFh
SDO (rx)	1100	601h – 67Fh
NMT Error Control	1110	701h – 77Fh



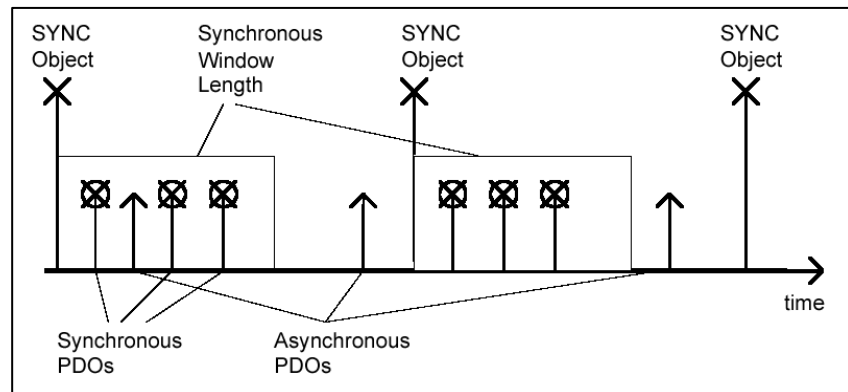
„Tx“ = wird vom Slave gesendet
 „Rx“ = wird vom Slave empfangen

8.4 Netzmanagement

SYNC:

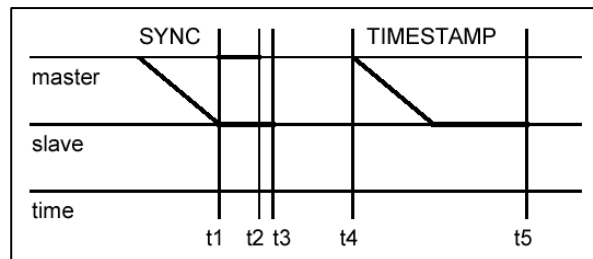
Das SYNC-Telegramm ist ein periodisches "Broadcast"-Telegramm und gibt den Basis-Bustakt vor. Um einen zeitlich äquidistanten Abstand zu ermöglichen, besitzt das SYNC-Telegramm eine hohe Priorität.

[COB-ID: 80h]



Time Stamp:

Das Time Stamp-Telegramm ist ein periodisches "Broadcast"-Telegramm und gibt die Systemzeit vor. Das Time Stamp-Telegramm wird üblicherweise direkt nach einem SYNC-Telegramm übertragen und gibt dann die Systemzeit des SYNC-Telegrammes an.



Um einen zeitlich genauen Übertragung zu ermöglichen, besitzt das Time Stamp-Telegramm eine hohe Priorität.

[COB-ID: 100h]

Nodeguarding:

Beim Nodeguarding überwacht der Master die CANopen® Slave Baugruppen durch zyklisch gesendete Telegramme an jeden Slave. Auf das Nodeguarding-Telegramm muss jeder CANopen® Slave mit einem Status-Telegramm antworten.

Mittels Nodeguarding kann die Steuerung den Ausfall eines CANopen Slaves erkennen.

[COB-ID: 700h + Node-ID]

Lifeguarding:

Beim Lifeguarding überwacht jeder CANopen® Slave, ob der Master das einmal gestartete Nodeguarding kontinuierlich innerhalb bestimmter Zeitlimits durchführt.

Wenn das Nodeguarding Telegramm des Masters ausbleibt, kann die dezentrale Peripheriebaugruppe dieses mittels Lifeguarding feststellen und z.B. alle Ausgänge in den sicheren Zustand versetzen.

Heartbeat:

Die Heartbeat Überwachung entspricht dem Nodeguarding, wobei aber keine Anforderungstelegramme vom CANopen® Master generiert werden. Das Heartbeat Telegramm wird vom Node selbsttätig gesendet und kann im Master ausgewertet werden (Producer Heartbeat). Der Master sendet seinerseits ein Heartbeat Telegramm im 600ms Intervall, welches dann von den Nodes ausgewertet werden kann (Consumer Heartbeat).



Einige CANopen® Slave Baugruppen erzeugen beim Ein- oder Ausschalten spezielle Emergency-Messages.

Emergency-Message:

Sollte bei einem CANopen® Slave ein Störfall auftreten, z.B. der Ablauf des Lifeguarding Timers, so sendet er eine Emergency-Message auf den Bus.

[COB-ID: 80h + Node-ID]

Auf ein Emergency-Telegramm können alle Teilnehmer z.B. einen Notstop ausführen.

BootUp-Message:

CANopen® Slaves erzeugen nach dem Einschalten eine BootUp-Meldung, die der Master erkennen kann, um diesen neuen Teilnehmer zu initialisieren.

[COB-ID: 700h + Node-ID + 1 Byte Daten: 00h]

9 Anhang

9.1 Technische Daten

Bestellnummer	DP/CAN-Koppler	700-650-CAN01
Abmessungen	114 x 18 x 108 mm (LxBxH)	
Gewicht	ca. 120 g	

CAN Schnittstelle

Typ:	ISO/DIN 11898-2, CAN High Speed physical Layer
Übertragungsrate:	10 kbit/s bis 1Mbit/s
Protokoll:	CANopen [®] Master CAN 2.0A (11Bit)
Anschluss:	3-polig Schraubklemme

PROFIBUS-DP Schnittstelle

Typ:	PROFIBUS -DP nach EN 50 170
Übertragungsrate:	19,2 kbit/s bis 12Mbit/s
Anschluss:	Sub-D Stecker 9-polig

Versorgung

Spannung:	DC +24V
Stromaufnahme:	180 mA (typ.)

Zulässige Umgebungsbedingungen

Betriebstemperatur:	0°C ... 60°C
Lagertemperatur:	-25°C ... 75°C
Schutzart:	IP 20

Besonderheiten

Qualitätssicherung:	nach ISO 9001:2000
Wartung:	Wartungsfrei (keine Batterie oder Akku)

9.2 Steckerbelegung

PROFIBUS 9pol. Sub-D Buchse:

Pin	PROFIBUS DP
1	-
2	
3	Datenleitung B
4	-
5	GND
6	VP (Versorgung der Abschlusswiderstände)
7	-
8	Datenleitung A
9	-

CAN-Stecker 3polig (kein Abschlusswiderstand):

1	CAN High
2	CAN-GND
3	CAN-Low

Spannungsversorgung 3polig:

1	GND
2	V-
3	V+

9.3 Weiterführende Dokumentation

Internet: www.can-cia.org

CAN Specification 2.0, Part A & Part B

High Layer Protocol CANopen[®]

Holger Zeltwanger: "CANopen[®]", VDE Verlag, ISBN 3-8007-2448-0

Notizen