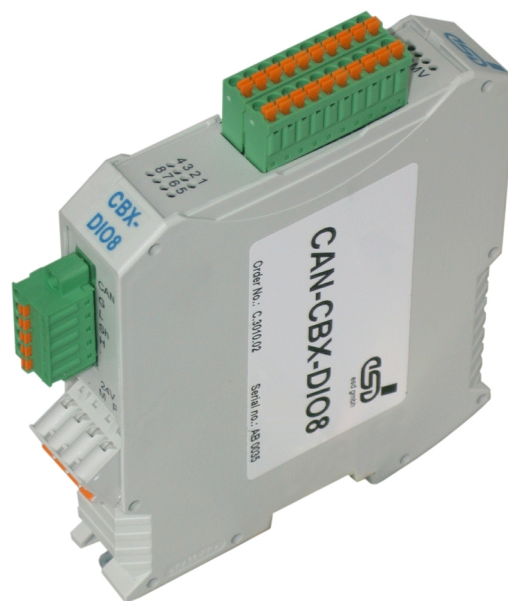




CAN-CBX-DIO8

**Kompaktes I/O-Modul
mit InRailBus**



Handbuch

zu Artikel C.3010.02



Hinweis

Der Inhalt dieses Handbuches wurde mit größter Sorgfalt erarbeitet und geprüft. **esd** übernimmt jedoch keine Verantwortung für Schäden, die aus Fehlern in der Dokumentation resultieren könnten. Insbesondere Beschreibungen und technische Daten sind keine zugesicherten Eigenschaften im rechtlichen Sinne.

esd hat das Recht, Änderungen am beschriebenen Produkt oder an der Dokumentation ohne vorherige Ankündigung vorzunehmen, wenn sie aus Gründen der Zuverlässigkeit oder Qualitätssicherung vorgenommen werden oder dem technischen Fortschritt dienen.

Sämtliche Rechte an der Dokumentation liegen bei **esd**. Die Weitergabe an Dritte und Vervielfältigung jeder Art, auch auszugsweise, sind nur mit schriftlicher Genehmigung durch **esd** gestattet.

© 2015 esd electronic system design gmbH, Hannover

esd electronic system design gmbH

Vahrenwalder Str. 207
30165 Hannover

Tel.: 0511/372 98-0
FAX : 0511/372 98-68
E-Mail: info@esd.eu
Internet: www.esd.eu



Dieses Handbuch enthält wichtige Informationen und Anweisungen für eine sichere und sachgerechte Anwendung des CAN-CBX-Moduls. Lesen Sie das Handbuch sorgfältig durch, bevor sie das CAN-CBX-Modul verwenden und befolgen Sie die Anweisungen.

Dieses Handbuch ist ein Bestandteil des Produktes. Bewahren Sie es für eine spätere Verwendung auf.

Trademark-Hinweise

CiA® und CANopen® sind registrierte Unionsmarken des CAN in Automation e.V.

Alle anderen hier aufgeführten Markenzeichen, Produktnamen, Firmennamen und Firmenlogos sind Eigentum des jeweiligen Rechteinhabers.

Dokument-Datei:	I:\Texte\Doku\MANUALS\CAN\CBX\DIO8\Deutsch\CAN-CBX-DIO8_Handbuch_de_32.wpd
Datum des Ausdrucks:	2016-08-24

Platinenversionen:	ab Rev. 1.3
Firmware-Version:	2.0

Änderungen in den Kapiteln

Die hier aufgeführten Änderungen im Anwenderhandbuch betreffen sowohl Änderungen in der *Hardware* und *Firmware* als auch reine Änderungen in der *Beschreibung* der Sachverhalte.

Version	Kapitel	Änderungen gegenüber Vorversion
3.1	10.	Konformitätserklärung vom 2013-11-19 ersetzt vorhandene.
3.2	-	Sicherheitshinweise überarbeitet und eingefügt, Klassifikation eingefügt
	2	Sicherheitshinweis eingefügt
	3.1	Hinweis eingefügt
	3.2	Hinweis eingefügt
	3.2.3	Hinweis zur Power LED eingefügt
	3.2.6	LED Beschreibung überarbeitet
	3.3.1	Hinweis eingefügt
	3.3.3	Zuordnung der Kodierschalter in dieses Kapitel verschoben
	3.4.2	Hinweis zur Erdumng der Tragschiene aufgenommen
	4.1	Angaben zu Spannungsversorgung, Temperaturbereich und Gehäuse geändert
	4.2	Angabe zum EEPROM geändert
	4.5	Hinweis zur EDS-Datei
	5.2	Bezeichnungen des CAN Steckers (X600) korrigiert, Bild aktualisiert
	5.4.2	Zeichnung korrigiert und Digitale I/O (X400)
	5.4.3	Hinweis zur Leitungslänge aufgenommen
	6., 7.	Kapitel aktualisiert
	8.9	Objekt 1001 _n in Tabelle ergänzt
	8.9.11	Beschreibung ergänzt
	8.9.12	Beschreibung ergänzt
	8.12	Hinweis eingefügt
10.	Konformitätserklärung aktualisiert	
11.	Bestellhinweise der TBUS-Stecker ergänzt	

Weitere technische Änderungen vorbehalten.

Klassifikation der Warn- und Sicherheitshinweise

Dieses Handbuch enthält wichtige Warnhinweise, Sicherheitsanweisungen und Beschreibungen, die Sie befolgen müssen um Personenschäden oder Tod und Sachschäden zu vermeiden.



Dies ist das allgemeine Symbol für Gefahr.
Es warnt Sie vor möglichen Verletzungsgefahren. Befolgen Sie für einen sicheren Gebrauch und Umgang mit dem CAN-CBX-Modul unbedingt alle Sicherheitshinweise und Maßnahmen, die diesem Symbol folgen.

GEFAHR, WARNUNG, VORSICHT

Abhängig vom Risikograd der Gefährdung werden die Signalwörter GEFAHR, WARNUNG oder VORSICHT verwendet, um auf eine Gefährdung von Personen hinzuweisen.



GEFAHR
Hier wird auf eine Gefährdung mit hohem Risikograd hingewiesen, die, wenn sie nicht vermieden wird, zu einer schweren Körperverletzung oder zum Tod führen wird.



WARNUNG
Hier wird auf eine Gefährdung mit mittlerem Risikograd hingewiesen, die, wenn sie nicht vermieden wird, zu einer schweren Körperverletzung oder zum Tod führen kann.



VORSICHT
Hier wird auf eine Gefährdung mit niedrigem Risikograd hingewiesen, die, wenn sie nicht vermieden wird, zu einer geringfügigen oder mäßigen Verletzung von Personen führen kann.

ACHTUNG

Das Signalwort ACHTUNG weist auf eine möglicherweise schädliche Situation hin, die zu einer Schädigung des CAN-CBX-Moduls und/oder seiner Umgebung führen kann, wenn sie nicht vermieden wird.



ACHTUNG
Hier wird darauf hingewiesen, dass angemessene ESD-Schutzmaßnahmen ergriffen werden müssen, um elektronische Bauelemente gegen Schäden, die durch elektrostatische Entladung hervorgerufen werden, zu schützen.



ACHTUNG
Diese Hinweisbox enthält das allgemeine Gebotszeichen und gibt Anweisungen, die für einen sicheren Gebrauch des CAN-CBX-Moduls befolgt und eingehalten werden müssen.

HINWEIS



HINWEIS
Diese Hinweisbox enthält Anwendungstipps und andere besonders nützliche Informationen.



Sicherheitshinweise

- Beachten Sie im Umgang mit dem CAN-CBX-DIO8 die folgenden Sicherheitshinweise und lesen Sie dieses Handbuch aufmerksam durch, um Verletzungen von Personen und Schäden am Gerät zu vermeiden.
- Verwenden Sie keine beschädigten Leitungen für den Anschluss des CAN-CBX-Moduls und beachten Sie die Verdrahtungshinweise zum CAN-Bus.
- Bei Beschädigungen am Gerät, die die Sicherheit betreffen könnten, müssen unverzüglich geeignete Maßnahmen getroffen werden, die eine Gefährdung von Personen oder Sachen verhindern.
- Mit der Einrichtung verbundene Stromkreise müssen gegen gefährliche Spannungen ausreichend geschützt sein (SELV nach EN 60950-1).
- Das CAN-CBX-DIO8 darf nur an Versorgungsstromkreisen betrieben werden, die berührungssicher sind. Ein Netzteil, welches eine Schutzkleinspannung (SELV oder PELV) nach EN 60950-1 zur Verfügung stellt, erfüllt diese Bedingung.
- Das CAN-CBX-Modul darf nicht geöffnet werden.
- Das Gerät muss vor der Inbetriebnahme fest montiert sein.
- Die zugelassene Betriebslage ist wie abgebildet. (Abbildung: 7) Weitere Betriebslagen sind nicht zugelassen.
- Lassen Sie keine Flüssigkeiten in das CAN-CBX-Modul eindringen, da sonst elektrische Schläge oder Kurzschlüsse die Folge sein können.
- Schützen Sie das CAN-CBX-Modul vor Feuchtigkeit und Dämpfen.
- Schützen Sie das CAN-CBX-Modul vor Stößen und Vibrationen.
- Das CAN-CBX-Modul wird möglicherweise während des normalen Betriebs warm. Achten Sie stets auf ausreichende Luftzufuhr, damit die Wärme abgeführt werden kann.
- Betreiben Sie das CAN-CBX-Modul nicht in unmittelbarer Nähe von Wärmequellen und setzen Sie es keiner unnötigen Wärmestrahlung aus. Die zulässige Umgebungstemperatur ist in den technischen Daten festgelegt.



GEFAHR

Gefährliche Spannung - Risiko des elektrischen Schocks durch unabsichtliches direktes Berühren aktiver elektrischer Teile, die mit gefährlichen Spannungen versorgt werden innerhalb des Systems, in das das CAN-CBX-Modul installiert werden soll.

- Das CAN-CBX-DIO8 darf nur an Versorgungsstromkreisen betrieben werden, die berührungssicher sind. Ein Netzteil, welches eine Schutzkleinspannung (SELV oder PELV) nach EN 60950-1 zur Verfügung stellt, erfüllt diese Bedingung.

Qualifiziertes Personal

Diese Dokumentation wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuer- und Automatisierungstechnik. Die Installation und Inbetriebnahme des Produkts darf nur von qualifiziertem Personal vorgenommen werden, das berechtigt ist, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen.

Konformität

Das CAN-CBX-Modul ist ein industrielles Produkt und erfüllt die in der Konformitätserklärung am Ende dieses Handbuchs angegebenen EG-Richtlinien und Normen zur EMV für industrielle Umgebungen.

Warnung: In Wohnbereich, Geschäfts- und Gewerbebereichen sowie Kleinbetrieben kann das CAN-CBX-Modul Funkstörungen verursachen. In diesem Fall ist es erforderlich, dass der Anwender angemessene Maßnahmen ergreift.

Hinweis: Die zugesicherten EMV-Eigenschaften werden eingehalten, wenn:

- für die digitalen I/Os Leitungen mit maximal 30 m Länge eingesetzt werden.
- die Tragschiene an den Funktionserde-Kontakt (FE) angeschlossen ist! Bitte achten Sie darauf, den Widerstand des Kabels so gering wie möglich zu halten.

Bestimmungsgemäßer Gebrauch

Die bestimmungsgemäße Verwendung ist der Einsatz des CAN-CBX-Moduls als CANopen-Slave mit digitalen Ein- und Ausgängen. Der Hersteller haftet nicht für Schäden, die durch unsachgemäßen Gebrauch, nicht bestimmungsgemäße Verwendung oder in Folge von Nichtbeachtung der Sicherheitshinweise und Warnungen verursacht werden.

Jeder Eingriff in das CAN-CBX-Modul durch nicht von esd autorisierte Personen führt zum Verlust aller Garantieansprüche.

- Das CAN-CBX-Modul ist nur für die Anwendung innerhalb von Gebäuden vorgesehen.
- Das CAN-CBX-Modul darf nicht in explosionsgefährdeten Bereichen und Zonen für Gase und Stäube sowie in explosivstoffgefährdeten Bereichen eingesetzt werden.
- Der Einsatz zu medizinischen Zwecken ist nicht zulässig.

Wartungshinweis

Innerhalb und außerhalb des CAN-CBX-Modul befinden sich keine vom Anwender zu wartenden Komponenten. Jeder Eingriff in das Gerät durch nicht von esd autorisierte Personen führt zum Verlust aller Garantieansprüche.

Umwelthinweis

Auf Dauer unbrauchbar gewordene Geräte sind in geeigneter Weise zu entsorgen oder dem Hersteller zur Entsorgung zu übergeben. Bitte leisten auch Sie Ihren Beitrag zum Schutz unserer Umwelt.

Darstellungskonventionen

In diesem Handbuch werden die folgenden Darstellungen zur Unterscheidung und Hervorhebung der aufgelisteten Programmbestandteile verwendet.

Darstellung von	Beispiel
Datei- und Pfadname	/dev/null or <stdio.h>
Funktionsnamen	<i>open()</i>
Konstanten	NULL
Datentypen	uint32_t
Variablenamen	<i>Count</i>

Zahlendarstellung

Alle Zahlenangaben in diesem Dokument sind Dezimalzahlen, sofern nicht anders angegeben. Hexadezimalzahlen sind mit einem vorangestellten 0x gekennzeichnet. Zum Beispiel wird die Dezimalzahl 42 hexadezimal als 0x2A dargestellt.

1. Übersicht	11
2. Schnelle Inbetriebnahme	12
3. Hardware-Installation	14
3.1 Anschlussplan	14
3.2 Frontansicht	15
3.2.1 Blinkzustände	16
3.2.2 Bedeutung der CAN-Error-LED	16
3.2.3 Bedeutung der CANopen-Status-LED	17
3.2.4 Bedeutung der Error-LED	17
3.2.5 VIO-LED	18
3.2.6 Besondere Blinkzustände	18
3.2.7 I/O-LED 1-8	19
3.3 Kodierschalter	20
3.3.1 Einstellung der Node-ID über die Kodierschalter Low und High	20
3.3.2 Einstellung der Bitrate	21
3.3.3 Zuordnung der Kodierschalterbezeichnungen auf dem Gerät zum Schaltplan	21
3.4 Einbau des Moduls bei Verwendung des InRailBus-Verbinders	22
3.4.1 Anschluss über den CBX-InRailBus	23
3.4.2 Anschluss der Versorgungsspannung	24
3.4.3 Anschluss von CAN	25
3.5 Ausbau des CAN-CBX-Moduls vom InRailBus	25
4. Technische Daten	26
4.1 Allgemeine technische Daten	26
4.2 CPU-Baugruppe	27
4.3 CAN-Schnittstelle	27
4.4 Digitale Ein/Ausgänge	28
4.5 Software-Unterstützung	28
5. Steckerbelegungen	29
5.1 Spannungsversorgung 24V (X100)	29
5.2 CAN	30
5.2.1 CAN-Interface-Schaltung	30
5.2.2 CAN-Stecker (X600)	31
5.3 CAN und Versorgungsspannung über InRailBus X101	32
5.4 Digitale Ein/Ausgänge	33
5.4.1 Schaltung Digitale Eingänge	33
5.4.2 Schaltung Digitale Ausgänge	33
5.4.3 Steckerbelegung Digitale Ein-/Ausgänge (X400)	34
5.5 Leiteranschluss/Leiterquerschnitt	35
6. Korrekte Verdrahtung galvanisch getrennter CAN-Netze	36
6.1 Standards zur CAN-Verdrahtung	36
6.2 Leicht störbehaftete Industrieumgebung (zweiadrig verdrehte Leitung)	37

6.2.1 Grundregeln	37
6.2.2 Verkabelung	38
6.2.3 Abschlusswiderstand	38
6.3 Stark störbehaftete Industrieumgebung (vieradrig verdrehte Leitung)	39
6.3.1 Grundregeln	39
6.3.2 Verkabelung	40
6.3.3 Abschlusswiderstand	40
6.4 Erdung	41
6.5 Buslänge	41
6.6 Beispiele für CAN-Kabel	42
6.6.1 Kabel für leicht störbehaftete Industrieumgebung (zweiadrig)	42
6.6.2 Kabel für stark störbehaftete Industrieumgebung (vieradrig)	42
7. CAN-Bus Troubleshooting Guide	43
7.1 Bus-Abschluss	43
7.2 Erdung	44
7.3 Kurzschluss in der CAN-Verdrahtung	44
7.4 CAN_H/CAN_L-Spannungen	44
7.5 CAN-Transceiver Widerstandstest	45
7.6 Support bei esd	45
8. CANopen-Firmware	46
8.1 Begriffsdefinition	46
8.2 NMT-Boot-up	47
8.3 Das CANopen-Objektverzeichnis	47
8.4 Communication Parameter	48
8.4.1 Zugriff auf das Objektverzeichnis über SDOs	48
8.4.2 Nichtflüchtiges Speichern von Parametern im EEPROM	50
8.5 Übersicht der verwendeten CANopen-Identifizierer	51
8.5.1 Einstellung der COB-ID	51
8.6 PDO-Belegung	52
8.7 Setzen und Lesen der Aus- und Eingänge	53
8.7.1 Meldung der digitalen Eingänge	53
8.7.2 Digitale Ausgänge	53
8.7.3 Unterstützte Übertragungsarten nach DS-301	53
8.8 Communication Profile Area	54
8.8.1 Verwendete Bezeichnungen und Abkürzungen	54
8.9 Implementierte CANopen-Objekte	55
8.9.1 Übersicht der 1000er-Objekte mit Default-Werten	55
8.9.2 Device Type (1000 _h)	57
8.9.3 Error Register (1001 _h)	58
8.9.4 Pre-defined Error Field (1003 _h)	59
8.9.5 COB-ID of SYNC-Message (1005 _h)	61
8.9.6 Communication Cycle Period (1006 _h)	62
8.9.7 Manufacturer's Device Name (1008 _h)	63
8.9.8 Manufacturer's Hardware Version (1009 _h)	64
8.9.9 Manufacturer's Software Version 100A _h	64
8.9.10 Guard Time (100C _h) und Life Time Factor (100D _h)	65

8.9.11 Node Guarding Identifier (100E _h)	66
8.9.12 Store Parameters (1010 _h)	67
8.9.13 Restore Default Parameters (1011 _h)	69
8.9.14 COB_ID Emergency Message (1014 _h)	71
8.9.15 Inhibit Time EMCY (1015 _h)	72
8.9.16 Consumer Heartbeat Time (1016 _h)	73
8.9.17 Producer Heartbeat Time (1017 _h)	75
8.9.18 Identity Object (1018 _h)	76
8.9.19 Synchronous Counter Overflow Value (1019 _h)	78
8.9.20 Verify Configuration (1020 _h)	79
8.9.21 Error Behaviour Object (1029 _h)	80
8.9.22 NMT Startup (1F80 _h)	82
8.9.23 Self Starting Nodes Timing Parameters (1F91 _h)	83
8.9.24 Receive PDO Communication Parameter 1400 _h	84
8.9.25 Receive PDO Mapping Parameter 1600 _h	85
8.9.26 Objekt Transmit PDO1 Communication Parameter 1800 _h	88
8.9.27 Objekt Transmit PDO2 Communication Parameter 1801 _h	89
8.9.28 Transmit PDO1 Mapping Parameter 1A00 _h	90
8.9.29 Transmit PDO2 Mapping Parameter 1A01 _h	91
8.10 Device Profile Area	93
8.10.1 Übersicht der implementierten Objekte 6000 _h ... 6207 _h	93
8.10.2 Zusammenhang der implementierten Objekte für die digitalen Eingänge	94
8.10.3 Zusammenhang der implementierten Objekte für die digitalen Ausgänge	95
8.10.4 Read Input 8-bit (6000 _h)	96
8.10.5 Polarity Input 8-bit (6002 _h)	97
8.10.6 Filter Constant Input 8-bit (6003 _h)	98
8.10.7 Global Interrupt Enable Digital (6005 _h)	99
8.10.8 Interrupt Mask Any Change 8-bit (6006 _h)	100
8.10.9 Interrupt Mask Low to High 8-bit (6007 _h)	101
8.10.10 Interrupt Mask High to Low 8-bit (6008 _h)	102
8.10.11 Write Output 8-Bit (6200 _h)	103
8.10.12 Change Polarity Output 8-bit (6202 _h)	104
8.10.13 Error Mode Output 8-bit (6206 _h)	105
8.10.14 Error Value Output 8-bit (6207 _h)	106
8.11 Manufacturer Specific Profile Area	107
8.11.1 Übersicht der Objekte 2210 _h ... 2403 _h	107
8.11.2 Output Errors 8-Bit (2210 _h)	108
8.11.3 Output Error Mask Bit 1 to 8 (2220 _h)	109
8.11.4 I/O Direction_Mask 8-Bit (2250 _h)	110
8.11.5 VIO Voltage 16-Bit (2300 _h)	111
8.11.6 Sample Configuration (2310 _h)	112
8.11.7 Funktionsweise der Counter	113
8.11.8 Counter Enable (2400 _h)	115
8.11.9 Counter Preload (2401 _h)	116
8.11.10 Counter Value 16-Bit (2402 _h)	117
8.11.11 Counter Value 32-Bit (2403 _h)	118
8.12 Programm Download über CiA DSP-302-Objekte	119
8.12.1 Download-Steuerung über Objekt 1F51 _h	120

Inhalt	Seite
8.12.2 Download-Steuerung über Objekt 1F52 _h	121
9. Referenzen	122
10. Konformitätserklärung	123
11. Bestellhinweise	124



1. Übersicht

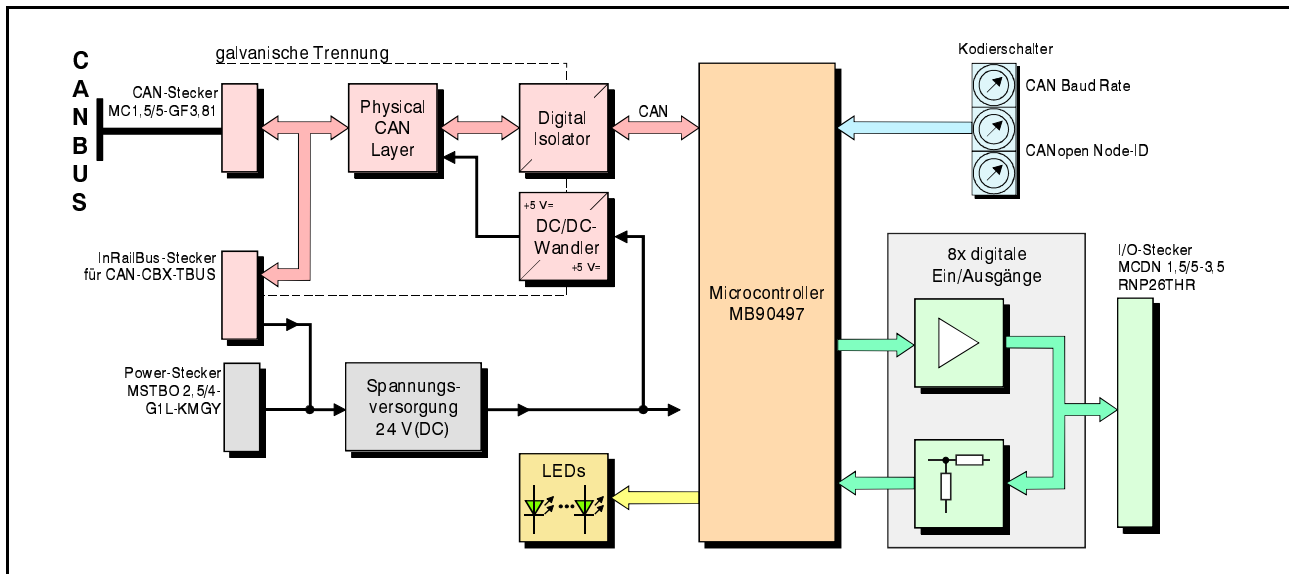


Abb. 1: Blockschaltbild des CAN-CBX-DIO8-Moduls

Das CAN-CBX-DIO8-Modul verfügt über einen MB90F497 Microcontroller, der die CAN-Daten in seinem lokalen SRAM zwischen speichert. Die Firmware wird im Flash gehalten. Zur Speicherung von Parametern dient ein serielles EEPROM.

Die acht digitalen Ein-/Ausgänge sind frei als Ausgang oder Eingang wählbar. Die Versorgungsspannung und der CAN-Bus-Anschluss werden entweder über den in die Modulschiene integrierten InRailBus-Verbinder oder separat über einen 5-poligen Stecker mit Federkraftanschluss-Kontakten zugeführt.

Die zu ISO 11898 kompatible CAN-Schnittstelle gestattet eine maximale Datenübertragungsrate von 1 MBit/s. Das CAN-Interface ist durch einen digitalen Datenkoppler und DC/DC-Wandler galvanisch getrennt.

Die CANopen[®] Knotennummer und die CAN-Bitrate können über drei außen liegende Drehschalter eingestellt werden.

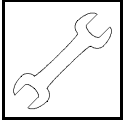


2. Schnelle Inbetriebnahme

Schritt	Aktion	siehe Seite								
	Bevor Sie mit der Inbetriebnahme beginnen lesen Sie die Sicherheitshinweise am Anfang des Handbuches sorgfältig!	4								
	GEFAHR Gefährliche Spannung - Risiko des elektrischen Schocks Das CAN-CBX-DIO8 darf nur an Versorgungsstromkreisen betrieben werden, die berührungssicher sind. Ein Netzteil, welches eine Schutzkleinspannung (SELV oder PELV) nach EN 60950-1 zur Verfügung stellt, erfüllt diese Bedingung.									
	Beachten Sie auch die Kapitel: “Einbau und Verdrahtung des Moduls” sowie “Korrekte Verdrahtung galvanisch getrennter CAN-Netze”!	22, 36								
1	Montieren und verdrahten Sie das CAN-CBX-Modul (Spannungsversorgung, CAN-Bus, Digitale Ein/Ausgänge)	14								
2	Beachten Sie beim Anschluss des CAN-Bus bitte, dass der CAN-Bus an beiden Enden abgeschlossen werden muss. esd bietet hierzu T-Stücke und Terminatoren. Das CAN-GND-Signal ist außerdem an genau einem Punkt im CAN-Netz zu erden. Ein CAN-Teilnehmer, dessen CAN-Interface nicht galvanisch getrennt ist, ist mit einer Erdung des CAN-GND gleichzusetzen.	-								
3	Einstellen der Baudrate (Nur wenn eine andere als die Default-Einstellung gewünscht wird.) Die Default-Bitrate beträgt 1 MBit/s. Die Bitrate kann über den Kodierschalter BAUD, wie im Kapitel: “Einstellung der CAN-Bitrate über den Kodierschalter” beschrieben, gesetzt werden.	21								
4	Einstellen der Modul-Nummer (Node-ID). Die Node-ID wird mit den Kodierschaltern LOW und HIGH, gesetzt. Sie darf Werte zwischen 1 und 127 (01-7F _h) annehmen.	20								
5	Legen sie die Versorgungsspannung an	-								
6	Modul starten mit NMT-Start Kommando <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>CAN-Identifizier</th> <th>Len</th> <th colspan="2">Data</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">2 Byte</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0 (Node-ID, 0 = alle Module starten)</td> </tr> </tbody> </table>	CAN-Identifizier	Len	Data		0	2 Byte	1	0 (Node-ID, 0 = alle Module starten)	-
CAN-Identifizier	Len	Data								
0	2 Byte	1	0 (Node-ID, 0 = alle Module starten)							



Schritt	Aktion	siehe Seite						
7	Daten CAN -> Digitale Ein/Ausgänge senden <table border="1" data-bbox="493 461 1268 642"> <thead> <tr> <th data-bbox="493 461 748 510">CAN-Identifizier</th> <th data-bbox="748 461 954 510">Len</th> <th data-bbox="954 461 1268 510">Data</th> </tr> </thead> <tbody> <tr> <td data-bbox="493 510 748 642">200_h + Node-ID</td> <td data-bbox="748 510 954 642">8 Byte</td> <td data-bbox="954 510 1268 642">0...8 Byte Nutzdaten Setzen der digitalen Ausgänge</td> </tr> </tbody> </table>	CAN-Identifizier	Len	Data	200 _h + Node-ID	8 Byte	0...8 Byte Nutzdaten Setzen der digitalen Ausgänge	-
CAN-Identifizier	Len	Data						
200 _h + Node-ID	8 Byte	0...8 Byte Nutzdaten Setzen der digitalen Ausgänge						
8	Daten Digitale Ein/Ausgänge -> CAN empfangen <table border="1" data-bbox="493 745 1268 927"> <thead> <tr> <th data-bbox="493 745 748 795">CAN-Identifizier</th> <th data-bbox="748 745 954 795">Len</th> <th data-bbox="954 745 1268 795">Data</th> </tr> </thead> <tbody> <tr> <td data-bbox="493 795 748 927">180_h + Node-ID</td> <td data-bbox="748 795 954 927">8 Byte</td> <td data-bbox="954 795 1268 927">0...8 Byte Nutzdaten Zustand der digitalen Eingänge</td> </tr> </tbody> </table>	CAN-Identifizier	Len	Data	180 _h + Node-ID	8 Byte	0...8 Byte Nutzdaten Zustand der digitalen Eingänge	-
CAN-Identifizier	Len	Data						
180 _h + Node-ID	8 Byte	0...8 Byte Nutzdaten Zustand der digitalen Eingänge						



3. Hardware-Installation

3.1 Anschlussplan

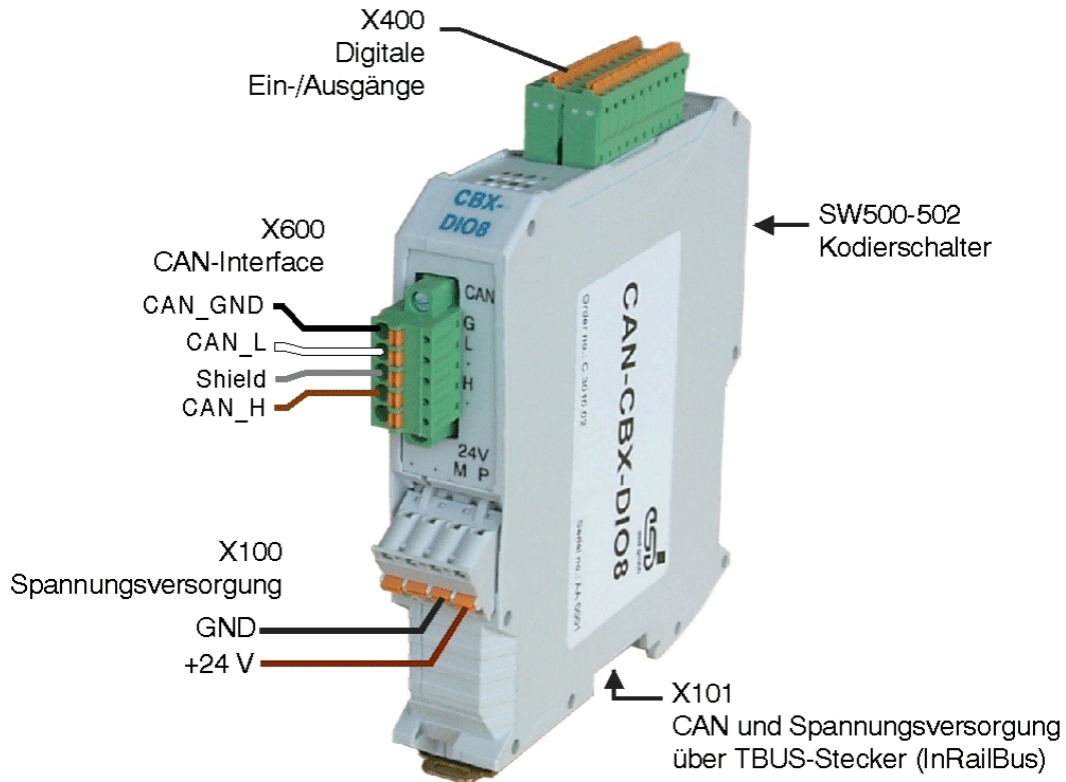


Abb. 2: Anschlüsse des CAN-CBX-DIO8-Moduls



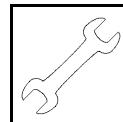
ACHTUNG

Lesen Sie Kapitel "Schnelle Inbetriebnahme" auf Seite 12, bevor sie mit der Installation der Hardware beginnen!



HINWEIS

Hinweise zum Leiteranschluss/Leiterquerschnitt entnehmen Sie bitte Seite 35.
Die Signalbelegung der Steckverbinder in tabellarischer Form ist ab Seite 29 abgedruckt.



3.2 Frontansicht

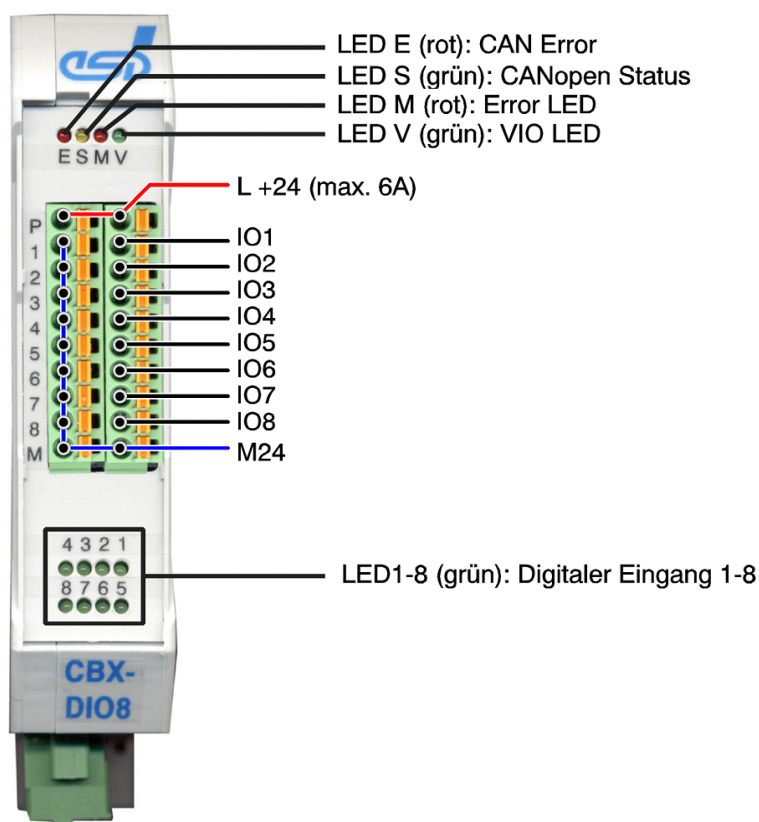


Abb. 3: Position der LEDs und Stecker in der Frontplatte

Das CAN-CBX-DIO8-Modul verfügt über 4 Status-LEDs und 8 LEDs für die I/O-Kanäle.



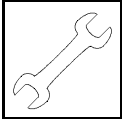
HINWEIS

Hinweise zum Leiteranschluss/Leiterquerschnitt entnehmen Sie bitte Seite 35.
 Die Signalbelegung der Steckverbinder in tabellarischer Form ist ab Seite 29 abgedruckt.



HINWEIS

Rote und grüne LEDs werden entsprechend der CANopen Spezifikation [3] grundsätzlich gegenphasig angesteuert.
 Bei bestimmten Blinkzuständen kann es bei gleichzeitiger Betrachtung aller LEDs zu einer Fehlinterpretation von Blinkzuständen nebeneinanderliegender LEDs kommen. Es wird daher ggf. empfohlen die Blinkzustände nebeneinanderliegender LEDs einzeln zu betrachten, indem die anderen LEDs verdeckt werden.



Hardware-Installation

3.2.1 Blinkzustände

Die Bezeichnungen der Blinkzustände der einzelnen LEDs wurden in Anlehnung an die von der CiA empfohlenen Bezeichnungen [3] gewählt. Die Bedeutung der Blinkzustände ist in den folgenden Kapiteln beschrieben. Es gibt prinzipiell 8 Blinkzustände, die die LEDs annehmen können:

Blinkzustand	Anzeige
an	LED an
aus	LED aus
Blinken	LED blinkt mit 2,5 Hz
Flackern	LED flackert mit 10 Hz
1 Blitz	LED 200 ms an, 1400 ms aus
2 Blitze	LED 200 ms an, 200 ms aus, 200 ms an, 1000 ms aus
3 Blitze	LED 2x (200 ms an, 200 ms aus) + 1x (200 ms an, 1000 ms aus)
4 Blitze	LED 3x (200 ms an, 200 ms aus) + 1x (200 ms an, 1000 ms aus)

Tabelle 1: Anzeigefunktion der LEDs

Die rote und die grüne LED blinken gegenphasig!

3.2.2 Bedeutung der CAN-Error-LED

LED-Kennung				Anzeigefunktion	
Aufdruck	Name	Farbe	Bezeichnung im Schaltplan* ¹	Blinkzustand	Bedeutung
E	CAN-Error	rot	200A	aus	kein Fehler
				1 Blitz	CAN-Controller ist in <i>Error Active</i>
				an	CAN-Controller ist <i>Bus Off</i> (oder Kodierschalterstellung ID-Node > 7F _h beim Einschalten; siehe Besondere Blinkzustände Seite 18)
				2 Blitze	Heartbeat- oder Nodeguard-Fehler aufgetreten. Die LED geht automatisch wieder aus, wenn wieder Nodeguard/Heartbeat-Message empfangen werden.

*¹ Der Schaltplan ist nicht Bestandteil dieses Handbuchs

Tabelle 2: Anzeigefunktion der roten CANopen Error-LED



3.2.3 Bedeutung der CANopen-Status-LED

LED-Kennung				Anzeigefunktion	
Aufdruck	Name	Farbe	Bezeichnung im Schaltplan* ¹	Blinkzustand	Bedeutung
S	CANopen-Status	grün	200B	Blinken	<i>Preoperational</i>
				an	<i>Operational</i>
				1 Blitz	<i>Stopped</i>
				3 Blitze	Modul ist im Bootloader-Mode, die Power LED ist aus (oder Kodierschalterstellung ID-Node > 7F _h beim Einschalten; siehe Seite 18)

*¹ Der Schaltplan ist nicht Bestandteil dieses Handbuchs

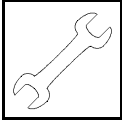
Tabelle 3: Anzeigefunktion der CANopen Status-LED

3.2.4 Bedeutung der Error-LED

LED-Kennung				Anzeigefunktion	
Aufdruck	Name	Farbe	Bezeichnung im Schaltplan* ¹	Blinkzustand	Bedeutung
M	Error	rot	200C	aus	kein Fehler
				an	Fehler an einem der fehlerüberwachten Ausgänge. - Ist das Modul während des Fehlers in den Zustand <i>Stopped</i> gewechselt, so bleibt die LED auch an, wenn der Fehler nicht mehr existiert - Fehler, die erst nach Eintreten des <i>Stopped</i> Zustandes auftreten, werden nicht angezeigt.
				2 Blitze	EEPROM Fehler - gespeicherte Daten hatten eine ungültige Checksumme und Default-Werte wurden geladen. - Blinkzustand bleibt, bis das Modul einen Reset durchführt oder ein Fehler an den Ausgängen auftritt.

*¹ Der Schaltplan ist nicht Bestandteil dieses Handbuchs

Tabelle 4: Anzeigefunktion der Error-LED



3.2.5 VIO-LED

Die in der folgenden Tabelle beschriebenen Anzeigefunktionen gelten nur, wenn sich das Modul im Zustand *Operational* befindet und sowohl Output Port als auch Output Port Config gesetzt sind. Treffen diese Bedingungen nicht zu, ist die LED immer an.

LED-Kennung				Anzeigefunktion wenn das Modul im Zustand <i>Operational</i> ist und Output Port und Output Port Config gesetzt sind	
Aufdruck	Name	Farbe	Bezeichnung im Schaltplan* ¹	Blinkzustand	Bedeutung
V	VIO_Sense	grün	200D	aus	Messwert der Betriebsspannung der Digitalausgänge (Objekt 2300 _h Subindex 1, siehe Seite 111) kleiner als 5 V
				Blinken	Messwert der Betriebsspannung der Digitalausgänge kleiner als der in Objekt 2300 _h Subindex 2 eingestellte Wert (siehe Seite 111) aber größer als 5 V
				Flackern	Messwert der Betriebsspannung der Digitalausgänge höher als der in Objekt 2300 _h Subindex 3 eingestellte Wert (siehe Seite 111)
				an	Messwert der Betriebsspannung der Digitalausgänge innerhalb der Grenzwerte (Objekt 2300 _h Subindex 2 und 3, siehe Seite 111)

*¹ Der Schaltplan ist nicht Bestandteil dieses Handbuchs

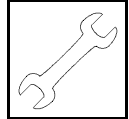
Tabelle 5: Anzeigefunktion der VIO-LED

3.2.6 Besondere Blinkzustände

Dieser Zustand wird von der CANopen-Status-LED und der CAN-Error-LED gemeinsam angezeigt:

LED-Anzeige	Bedeutung
<ul style="list-style-type: none"> - rote CAN-Error-LED: an und - CANopen-Status-LED: 3 Blitze 	Ungültige Node-ID: Die Kodierschalter für die Node-ID stehen auf einer ungültigen ID, die Firmware-Anwendung wird angehalten.

Tabelle 6: Besondere Blinkzustände



3.2.7 I/O-LED 1-8

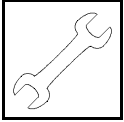
LED	Bezeichnung im Schaltplan* ¹	Anzeigefunktion = Zustand des Kanals
1	LED400D	IO1
2	LED400C	IO2
3	LED400B	IO3
4	LED400A	IO4
5	LED410D	IO5
6	LED410C	IO6
7	LED410B	IO7
8	LED410A	IO8

*¹ Der Schaltplan ist nicht Bestandteil dieses Handbuchs

Tabelle 7: Anzeigefunktion der LEDs 1-8

LED	Zustand des Kanals IO1-8
aus	Eingangsspannung unterhalb der unteren Schaltschwelle, bzw. Ausgang ausgeschaltet
an	Eingangsspannung oberhalb der oberen Schaltschwelle, bzw. Ausgang eingeschaltet

Tabelle 8: Zustand des Kanals IO1-8



3.3 Kodierschalter

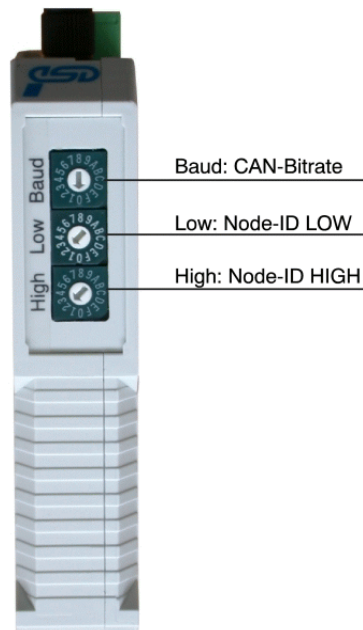


Abb. 4: Position der Kodierschalter



ACHTUNG

Die Auswertung der Kodierschalter durch die Firmware erfolgt beim Einschalten des Moduls. Änderungen der Einstellungen müssen daher **vor dem Einschalten** durchgeführt werden, da Änderungen während des Betriebes keine Auswirkungen haben.

Auch nach einem CANopen-Reset (z.B. NMT-Reset) werden die Einstellungen erneut eingelesen.

3.3.1 Einstellung der Node-ID über die Kodierschalter Low und High

Der Adress-Bereich des CAN-CBX-DIO8-Moduls ist *dezimal* von 1 bis 127 bzw. *hexadezimal* von 01_h bis $7F_h$ einstellbar.

Der Kodierschalter HIGH dient zur Einstellung der 3 höherwertigen Bits (höherwertiges Nibble), der Kodierschalter LOW zur Einstellung der 4 niederwertigen Bits.

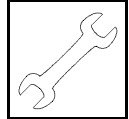


HINWEIS

Die folgenden Einstellungen sollten vermieden werden:

Einstellungen der Kodierschalter höher als $7F_h$ führen zu Fehlermeldungen, die rote CAN-Error-LED leuchtet dauerhaft.

Bei Einstellung der Kodierschalter auf 00_h befindet sich das Modul im Bootlader-Modus.



3.3.2 Einstellung der Bitrate

Die CAN-Bitrate wird mit dem Kodierschalter **Baud** eingestellt.

Über den Kodierschalter können Werte von 0_h bis F_h eingestellt werden. Die Einstellung der Bitrate erfolgt entsprechend der folgenden Tabelle:

Kodierschalter-Stellung [Hex]	Bitrate [kBit/s]
0	1000
1	666,6
2	500
3	333,3
4	250
5	166
6	125
7	100
8	66,6
9	50
A	33,3
B	20
C	12,5
D	10
E	800
F	83,3 ^{*2}

^{*2} ab FW 2.03

Tabelle 9: Index der Bitrate

3.3.3 Zuordnung der Kodierschalterbezeichnungen auf dem Gerät zum Schaltplan

Bezeichnung auf Gerät	Bezeichnung im Schaltplan ^{*1}
Baud	SW301
Low	SW300
High	SW302

^{*1} Der Schaltplan ist nicht Bestandteil dieses Handbuchs



3.4 Einbau des Moduls bei Verwendung des InRailBus-Verbinders

Sollen der CAN-Bus und die Versorgungsspannung über den InRailBus angeschlossen werden, gehen Sie bitte wie folgt vor:

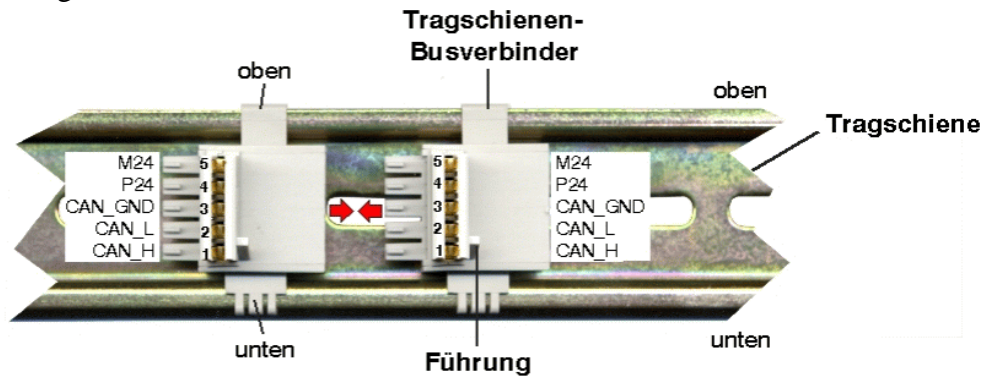


Abb. 5: Tragschiene mit Busverbinder

1. Der Tragschienen-Busverbinder des InRailBus wird an die Tragschiene angelegt und durch leichtes Andrücken auf der Tragschiene aufgerastet. Durch Zusammenstecken der einzelnen Busverbinder werden die Kommunikations- und Leistungssignale untereinander kontaktiert. Die Busverbinder können beliebig vor oder nach dem Aufstecken des CAN-CBX-Moduls verbunden werden.
2. Halten Sie das Modul leicht schräg nach hinten gekippt und setzen Sie das CBX-Modul auf den Busverbinder, so dass der obere Teil der Tragschiene dabei in die Einkerbung greift.

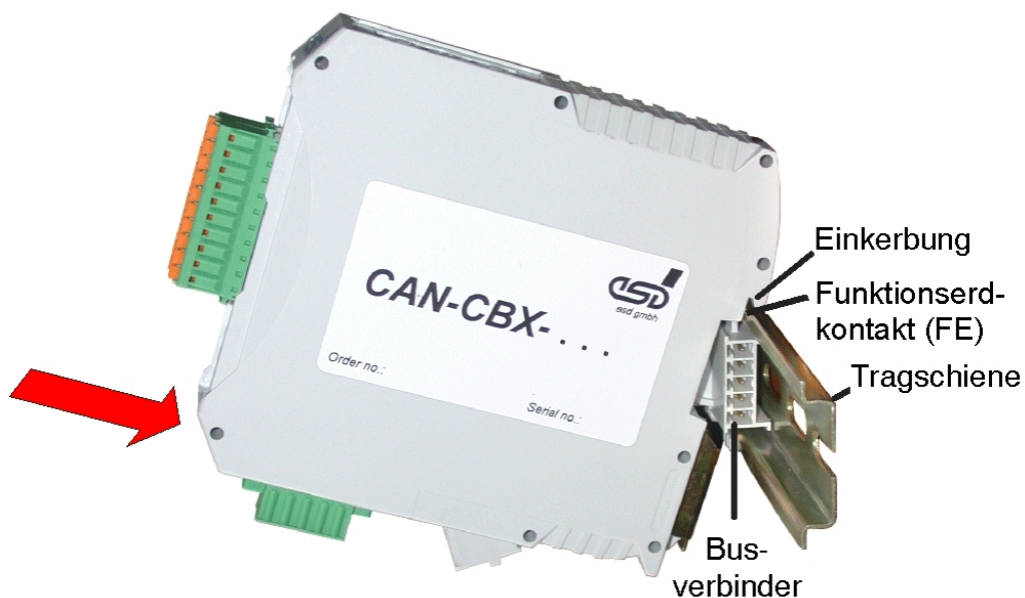
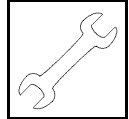


Abb. 6 : Einsetzen des CAN-CBX-Moduls



3. Schwenken Sie nun das CBX-Modul auf die Tragschiene auf, indem sie das Modul entsprechend der Pfeilrichtung in Abbildung 6 nach unten an die Tragschiene heran drücken. Dabei wird das Gehäuse durch die Führungsschiene des Busverbinders mechanisch geführt.
4. Beim Aufschwenken des CAN-CBX-Moduls rastet der untere Fußriegel auf dem unteren Teil der Tragschiene ein. Das Modul sitzt nun fest auf der Tragschiene und ist über den Busverbinder mit dem InRailBus verbunden. Verbinden Sie ggf. noch die Busverbinder untereinander und schließen Sie die +24 V Versorgungsspannung und das CAN-Interface an den InRailBus an.

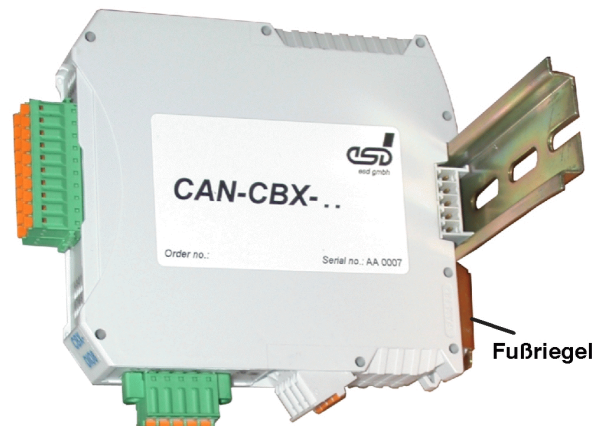


Abb. 7: Eingebautes CAN-CBX-Modul

3.4.1 Anschluss über den CBX-InRailBus

Um die Versorgungsspannung und die CAN-Signale über den InRailBus anschließen zu können, benötigen sie einen Anschlussstecker. Dieser Stecker ist nicht im Lieferumfang des CAN-CBX-Moduls enthalten. Er muss gesondert bestellt werden (Bestell-Nr. C.3000.02, siehe auch Bestellhinweise).

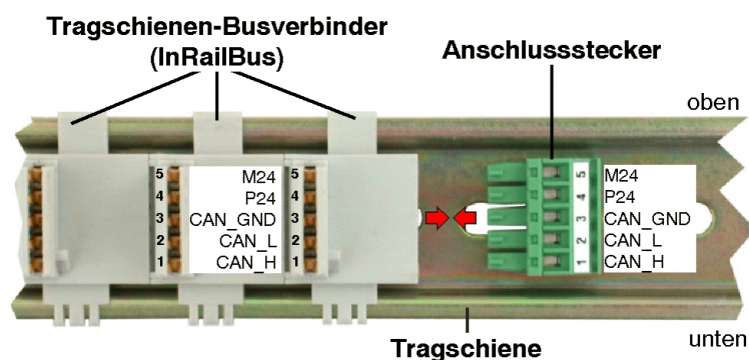
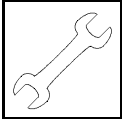


Abb. 8: Tragschiene mit InRailBus-Verbindern und Anschlussstecker

Stecken Sie den Anschlussstecker, wie in Abb. 8 beschrieben, von rechts in die Buchsenseite des äußeren Tragschienen-Busverbinders des InRailBus. Schließen sie nun das CAN-Interface und die Versorgungsspannung über den Anschlussstecker an.



Hardware-Installation

3.4.2 Anschluss der Versorgungsspannung

Der Anschluss der Versorgungsspannung erfolgt über den +24V-Stecker oder über den InRailBus-Stecker.



ACHTUNG

Bitte beachten Sie die Sicherheitshinweise und Anforderungen bezüglich der Versorgungsstromkreise (siehe Seite 5)!



ACHTUNG

Die Anschlüsse der 24 V Spannungsversorgung sind intern verbunden und dürfen **nicht** von zwei unabhängigen Stromquellen gleichzeitig versorgt werden!

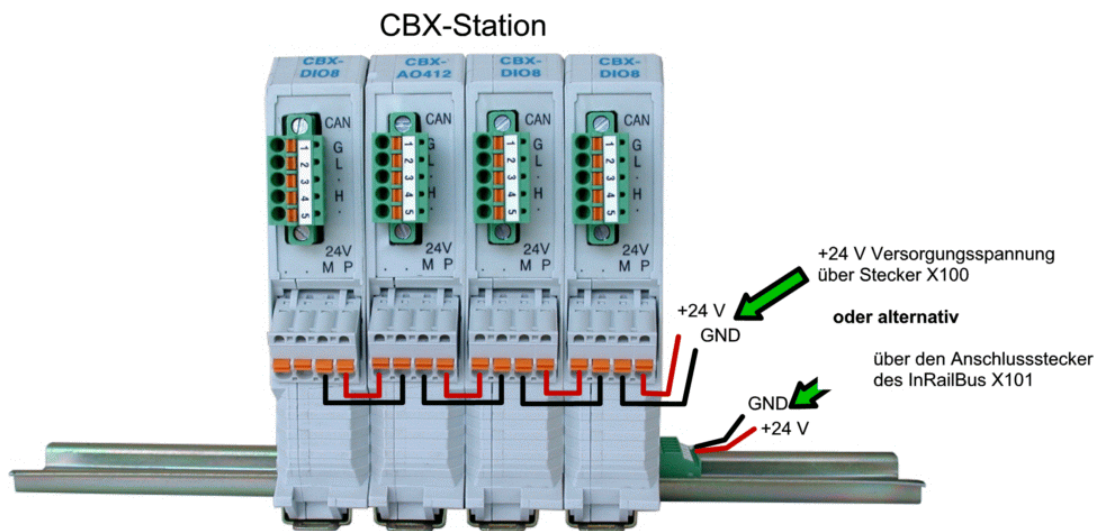


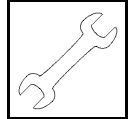
Abb. 9: CAN-CBX-Station mit Anschluss der Versorgungsspannung

Erdung der Tragschiene



HINWEIS

Das Modul ist über seinen Funktionserde-Kontakt mit der Tragschiene verbunden. Dies verbessert die Stabilität gegen elektromagnetische Störungen. Die Tragschiene ist daher mit einer geeigneten Funktionserde in der Umgebung bzw. in der Anlage zu verbinden. Dabei ist zu beachten, dass die Impedanz der Verbindung gering gehalten wird. Der Funktionserde-Kontakt des Moduls dient nicht der elektrischen Sicherheit.



3.4.3 Anschluss von CAN

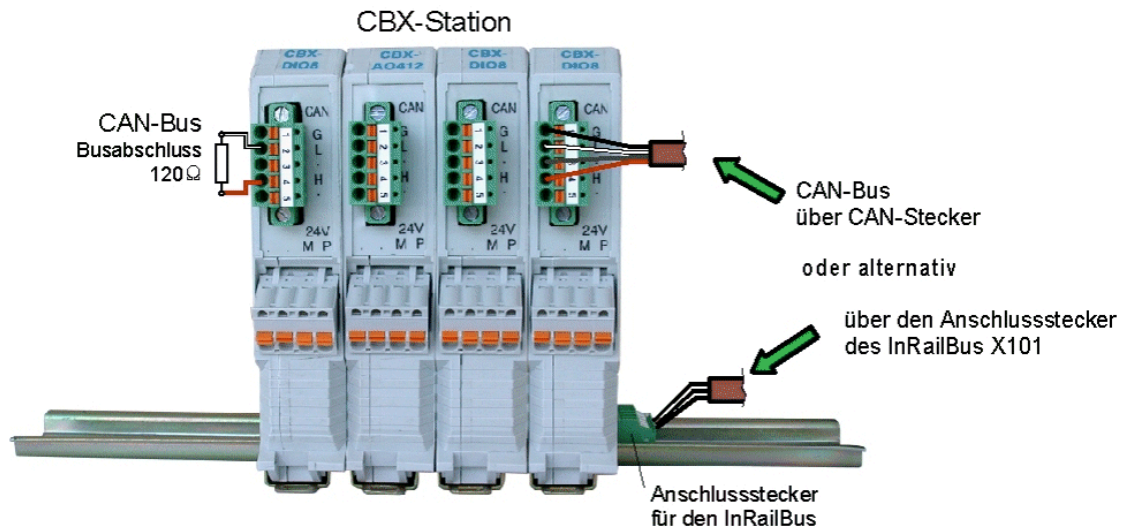


Abb. 10: Anschluss von CAN an die CBX-Station

Generell besteht die Möglichkeit die CAN-Signale über den InRailBus oder über den CAN-Stecker eines äußeren CAN-CBX-Moduls der CBX-Station einzuspeisen. Die Signale werden dann über den InRailBus an die folgenden CAN-CBX-Module der CAN-CBX-Station weitergeleitet. Die CAN-Signale dürfen über den CAN-Stecker des CAN-CBX-Moduls, das am anderen Ende der CBX-Station montiert ist, weitergeführt werden. Über die CAN-Stecker der mittleren CAN-CBX-Module der CBX-Station dürfen die CAN-Signale jedoch nicht weitergeführt werden, da dies zu unzulässigen Verzweigungen führt.

Bitte beachten Sie, dass an das CAN-CBX-Modul, das sich am Ende des InRailBus befindet ein Bus-Abschlusswiderstand angeschlossen werden muss, wenn der CAN-Bus dort endet (siehe Abb. 10).

3.5 Ausbau des CAN-CBX-Moduls vom InRailBus

Ist das CAN-CBX-Modul über den InRailBus verbunden, gehen Sie beim Ausbau wie folgt vor:

Lösen Sie das Modul von der Tragschiene indem Sie den Fußriegel (siehe Abb. 7) nach unten ziehen (z.B. mit einem Schraubendreher). Dabei löst sich das Modul unten von der Tragschiene und kann abgezogen werden.



HINWEIS

Es ist möglich, einzelne Gehäuse aus dem Verbund zu lösen, ohne die InRailBus-Verbindung zu unterbrechen, da beim Ziehen einzelner Module aus dem Verbund die Signalkette nicht unterbrochen wird.

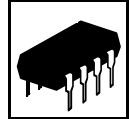


4. Technische Daten

4.1 Allgemeine technische Daten

Versorgungsspannung	Nennspannung: 24 V/DC \pm 20% Stromaufnahme (24 V, 20 °C): typ.: 30 mA (ohne VIO) max.: 40 mA
Steckverbinder	24V (4-pol. Leiterplattensteckverbinder, X100) - 24V-Spannungsversorgung CAN (5-pol. Leiterplattensteckverbinder, X600) - CAN-Schnittstelle 1 - 8, P, M (20-pol. Leiterplattensteckverbinder, X400) - Digitale Ein-/Ausgänge, 24 V, GND InRailBus (5-pol. TBUS-Verbnder, X101) - CAN-Schnittstelle und Spannungsversorgung über InRailBus Nur für werksseiteige Test und Programmierzwecke (interner Stecker): X200 (6-pol. Buchsenleiste)
Temperaturbereich	-20 °C ... +60 °C Umgebungstemperatur
IP-Schutzklasse	IP20
Verschmutzungsgrad	maximal zulässig nach DIN EN 61131-2: Verschmutzungsgrad 2
Gehäuse	Phoenix Contact MEMAX Kunststoffgehäuse für Tragschienenmontage NS35/7,5 DIN EN 60715
Luftfeuchtigkeit	max. 90%, nicht kondensierend
Abmessungen	Breite: 22,5 mm, Höhe: 99 mm, Tiefe: 114,5 mm (ohne Stecker)
Gewicht	ca. 125 g

Tabelle 9: Allgemeine Daten des Moduls



4.2 CPU-Baugruppe

CPU	16 Bit μ C MB90F497
RAM	2 kByte integriert
Flash	64 kByte integriert
EEPROM	1 kByte via I ² C

Tabelle 10: Microcontroller

4.3 CAN-Schnittstelle

Anzahl der CAN-Schnittstellen	1 x CAN
Anschluss	5-pol. Leiterplattensteckverbinder mit Federkraftanschluss-Kontakten oder über TBUS-Verbinder (InRailBus)
CAN-Controller	MB90F497, gemäß ISO 11898-1, (Software unterstützt zur Zeit nur 11-Bit CAN-Identifizier)
Galvanische Trennung des CAN-Interfaces gegenüber den anderen Baugruppen	über Datenkoppler und DC/DC-Wandler
Physical CAN Layer	Physical Layer gemäß ISO 11898-2, Übertragungsrate programmierbar von 10 kBit/s bis 1 MBit/s
Busabschluss	Abschlusswiderstand ist bei Bedarf extern anzuschließen

Tabelle 11: Daten der CAN-Schnittstelle



4.4 Digitale Ein/Ausgänge

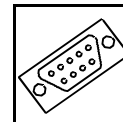
Anzahl	8 I/O-Kanäle, wahlweise Ein- oder Ausgang Überspannungsfest (max. 30V)
Digitale Ausgangsschaltung	
Ausführung	Plus-schaltend
Nennspannung	24 V
Nennbelastbarkeit	bis zu 1 A pro Ausgang (bei 50°C Umgebungstemperatur, alle Kanäle geschaltet)
Digitale Eingangsschaltung	
Nennspannung	24 V
Schutzschaltungen	20-pol. Mini-Combicon
Eingangsschaltsschwellen	$U_{INAus} = 0 \dots 8 \text{ V},$ $U_{INEin} = \geq 12 \text{ V},$ $U_{INMax} = 30 \text{ V}$

Tabelle 12: Daten der digitalen Ein/Ausgänge

4.5 Software-Unterstützung

Die Firmware des Moduls unterstützt CANopen[®] nach der CiA[®] CANopen[®] Spezifikation CiA 301 [1].

Die EDS-Datei des CAN-CBX-DIO8 kann von der esd-Webseite unter www.esd.eu heruntergeladen werden.

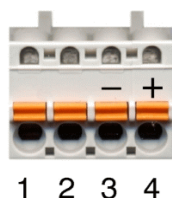


5. Steckerbelegungen

5.1 Spannungsversorgung 24V (X100)

Gerätebuchse: Phoenix Contact MSTBO 2,5/4-G1L-KMGY
 Leitungsstecker: Phoenix Contact FKCT 2,5/4-ST, 5.0 mm Raster,
 Federkraftanschluss-Kontakte,
 Phoenix Contact Bestell-Nr.: 19 21 90 0 (im Lieferumfang enthalten)
 Zu Leiteranschluss und Leiterquerschnitt siehe Seite 35.

Pin-Zuordnung:



Pin-Belegung:

Gehäusebezeichnung	24V	
	M	P
Stecker-Aufdruck	-	+

Pin-Nr.	1	2	3	4
Signal	P24 (+ 24 V)	M24 (GND)	M24 (GND)	P24 (+ 24 V)

Siehe auch Anschlussplan Seite 14.



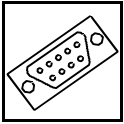
HINWEIS

Die Pins 1 und 4 sind intern miteinander verbunden.
 Die Pins 2 und 3 sind intern miteinander verbunden.

Signalbeschreibung:

P24... Versorgungsspannung +24 V

M24... Bezugspotenzial



Steckerbelegung

5.2 CAN

5.2.1 CAN-Interface-Schaltung

Das CAN Physical Layer ist gemäß ISO 11898-2 ausgelegt. Die differentiellen CAN-Bus-Signale sind von den anderen Potenzialen über einen magnetischen Datenkoppler und DC/DC-Wandler galvanisch getrennt.

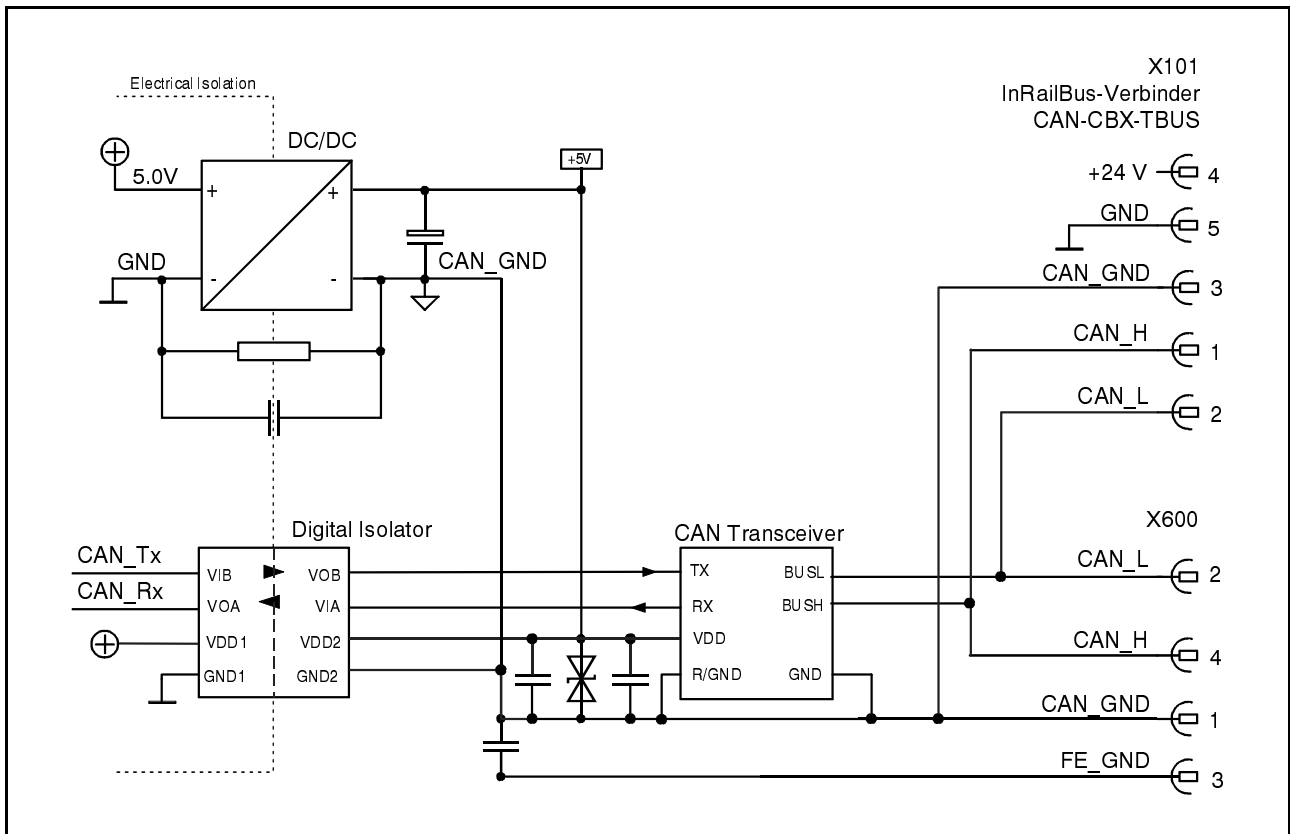
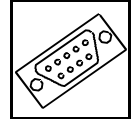


Abb. 11: Schaltung des CAN-Interface

Die CAN Schnittstelle kann entweder über den CAN-Stecker oder optional über den InRailBus angeschlossen werden. Benutzen Sie für den Anschluss über den InRailBus den Tragschienenbusverbinder des CBX-InRailBus (CAN-CBX-TBUS), siehe Bestellhinweise (Seite 124).

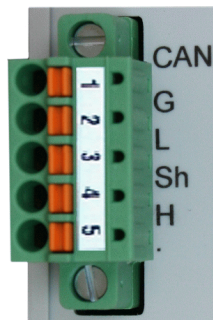


5.2.2 CAN-Stecker (X600)

Gerätebuchse: Phoenix Contact MC 1,5/5-GF-3,81

Leitungsstecker: Phoenix Contact FK-MCP 1,5/5-STF-3,81, Federkraftanschluss-Kontakte
 Phoenix Contact Bestell-Nr.: 1851261 (im Lieferumfang enthalten)
 Zu Leiteranschluss und Leiterquerschnitt siehe Seite 35.

Pin-Zuordnung: (Ansicht Leitungsstecker)



Pin-Belegung:

Labelling	Signal	Pin
G	CAN_GND	1
L	CAN_L	2
Sh	Shield	3
H	CAN_H	4
•	-	5

Signalbeschreibung:

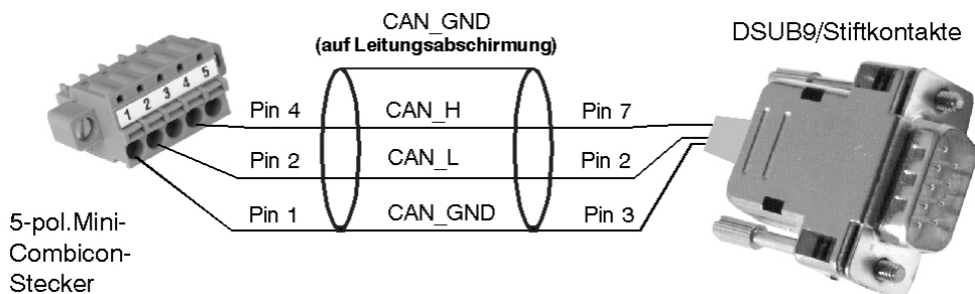
CAN_L, CAN_H ... CAN-Signale

CAN_GND ... Bezugspotenzial des lokalen CAN-Physical Layers

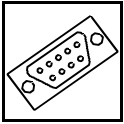
Shield ... Anschluss für Leitungsabschirmung (bei Hutschienenmontage direkter Kontakt zu Hutschienenpotenzial)

- ... nicht angeschlossen

Empfehlung eines Adapterkabels 5-pol-COMBICON (hier Leitungsstecker FK-MCP1,5/5-STF-3,81 mit Federkraftanschluss-Kontakten) auf 9-pol-DSUB:



Die Belegung des 9-poligen DSUB-Steckers erfolgt nach CiA DS 102.

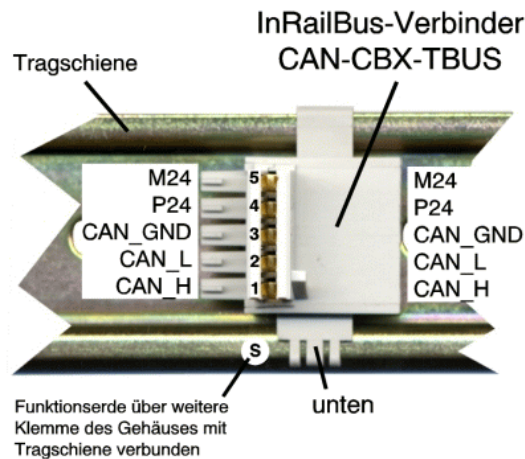


Steckerbelegung

5.3 CAN und Versorgungsspannung über InRailBus X101

Steckertyp: Tragschienensteckverbinder CAN-CBX-TBUS
(Phoenix Contact ME 22,5 TBUS 1,5/5-ST-3,81 KMGY)

Steckeraufsicht:

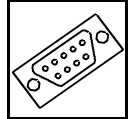


Pin-Belegung:

Pin	Signal
5	M24 (GND)
4	P24 (+24 V)
3	CAN_GND
2	CAN_L
1	CAN_H
S	FE (PE_GND)

Signalbeschreibung:

CAN_L,
CAN_H ... CAN-Signale
CAN_GND ... Bezugspotenzial des lokalen CAN-Physical Layers
P24... Versorgungsspannung +24 V
M24... Bezugspotenzial
FE... direkt am Gehäuse des Moduls: Funktionserde (mit Hutschienenpotenzial verbunden)



5.4 Digitale Ein/Ausgänge

5.4.1 Schaltung Digitale Eingänge

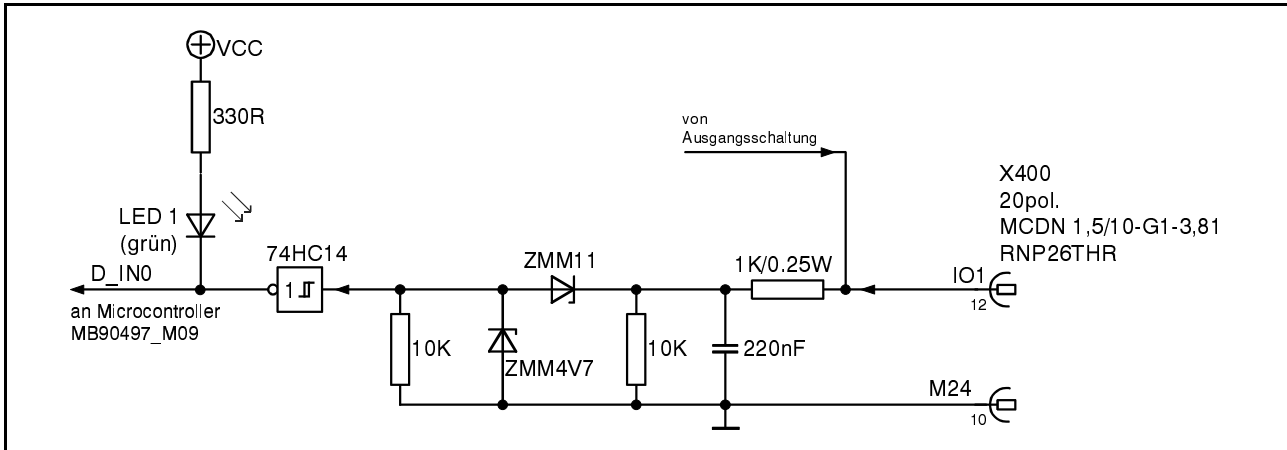


Abb. 12: Schaltung des digitalen Eingangs IO1

5.4.2 Schaltung Digitale Ausgänge

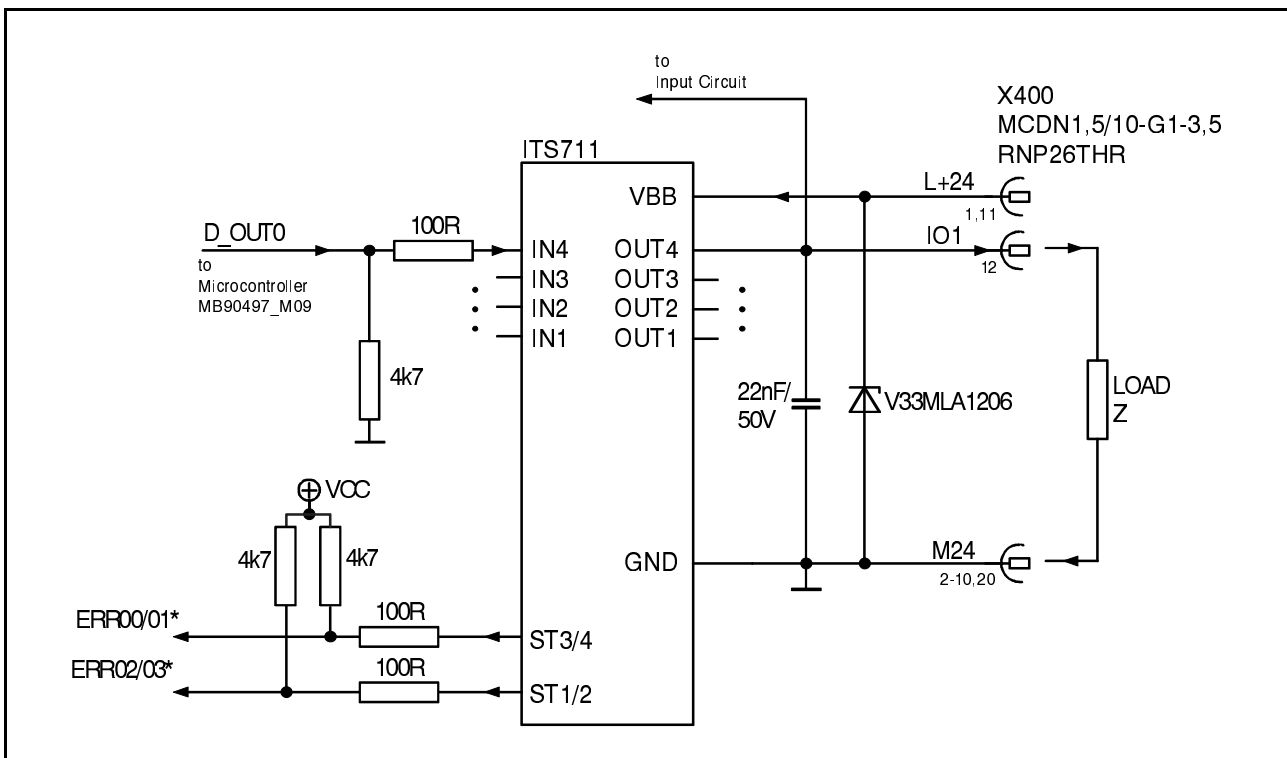
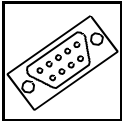


Abb. 13: Schaltung des digitalen Ausgangs IO1



Steckerbelegung

5.4.3 Steckerbelegung Digitale Ein-/Ausgänge (X400)

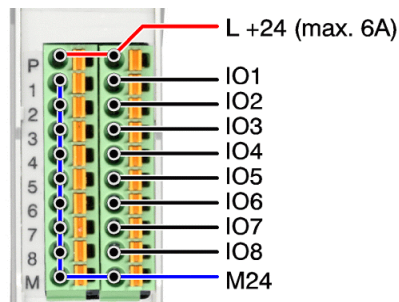
Gerätebuchse: Phoenix Contact MCDN 1,5/10-G1-3,5 RNP26THR

Leitungsstecker: 2x Phoenix Contact FMC 1,5/10-ST-3,5 (Federkraftanschluss-Kontakte)

Phoenix Contact Bestell-Nr.: 1952348 (im Lieferumfang enthalten)

Zu Leiteranschluss und Leiterquerschnitt siehe Seite 35.

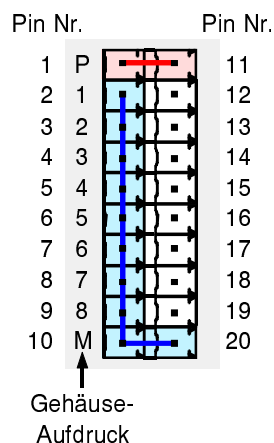
Steckeraufsicht:



Pin Belegung:

Signal	Pin
L + 24 V	1
M24	2
	3
	4
	5
	6
	7
	8
	9
	10

Pin-Zuordnung:



Pin Belegung:

Pin	Signal
11	L + 24 V
12	IO1
13	IO2
14	IO3
15	IO4
16	IO5
17	IO6
18	IO7
19	IO8
20	M24

Signalbeschreibung:

L +24 V... Versorgungsspannung der digitalen Ausgänge

M24... Bezugspotenzial

IO1-8... Signale der digitalen I/Os 1-8



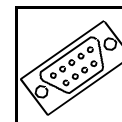
ACHTUNG

Die maximale Strombelastung der Steckerkontakte beträgt 6A/Pin. Sollen alle 8 Ausgänge mit der maximal zulässigen Last betrieben werden, so sind für die Spannungszuführung die Pins 1 und 11 zu verwenden. Die Pins sind auf der Leiterplatte miteinander verbunden.



ACHTUNG

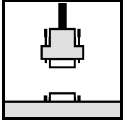
Die zugesicherten EMV-Eigenschaften werden eingehalten, wenn eine Kabellänge von maximal 30 m verwendet wird.



5.5 Leiteranschluss/Leiterquerschnitt

Die folgende Tabelle enthält einen Auszug aus den technischen Daten der verwendeten Stecker.

Schnittstelle	Spannungsversorgung 24 V ^[4]	CAN-Stecker [5]	Digitale Ein-/Ausgänge ^[6]
Steckertyp Leitungsstecker (Artikelfamilie)	FKCT 2,5/...-ST KMGY	FK-MCP 1,5/...- STF-3,81	FMC 1,5/10-ST- 3,5
Anschlussart	Zugfederanschluss	Federkraftanschluss	Federkraftanschluss
Abisolierlänge	10 mm	9 mm	10mm
Leiterquerschnitt starr min	0,2 mm ²	0,14 mm ²	0,2 mm ²
Leiterquerschnitt starr max	2,5 mm ²	1,5 mm ²	1,5 mm ²
Leiterquerschnitt flexibel min	0,2 mm ²	0,14 mm ²	0,2 mm ²
Leiterquerschnitt flexibel max	2,5 mm ²	1,5 mm ²	1,5 mm ²
Leiterquerschnitt flexibel mit Aderendhülse ohne Kunststoffhülse min	0,25 mm ²	0,25 mm ²	0,25 mm ²
Leiterquerschnitt flexibel mit Aderendhülse ohne Kunststoffhülse max	2,5 mm ²	1,5 mm ²	1,5 mm ²
Leiterquerschnitt flexibel mit Aderendhülse mit Kunststoffhülse min	0,25 mm ²	0,25 mm ²	0,25 mm ²
Leiterquerschnitt flexibel mit Aderendhülse mit Kunststoffhülse max	2,5 mm ²	0,5 mm ²	0,75 mm ²
Leiterquerschnitt AWG/kcmil min	24	26	24
Leiterquerschnitt AWG/kcmil max	12	16	16
2 Leiter gleichen Querschnitts starr min	unzulässig	unzulässig	unzulässig
2 Leiter gleichen Querschnitts starr max	unzulässig	unzulässig	unzulässig
2 Leiter gleichen Querschnitts flexibel min	unzulässig	unzulässig	unzulässig
2 Leiter gleichen Querschnitts flexibel max	unzulässig	unzulässig	unzulässig
2 Leiter gleichen Querschnitts flexibel mit Aderendhülse ohne Kunststoffhülse min	unzulässig	unzulässig	unzulässig
2 Leiter gleichen Querschnitts flexibel mit Aderendhülse ohne Kunststoffhülse max	unzulässig	unzulässig	unzulässig
2 Leiter gleichen Querschnitts flexibel mit Zwillingsaderendhülsen (TWIN-AEH) mit Kunststoffhülse min	0,5 mm ²	unzulässig	unzulässig
2 Leiter gleichen Querschnitts flexibel mit Zwillingsaderendhülsen (TWIN-AEH) mit Kunststoffhülse max	1,0 mm ²	unzulässig	unzulässig
AWG nach UL/CUL min	26	28	24
AWG nach UL/CUL max	12	16	16



6. Korrekte Verdrahtung galvanisch getrennter CAN-Netze

Generell sind bei der CAN-Verdrahtung sämtliche gültigen Richtlinien (DIN, VDE) bzgl. EMV-gerechtem Aufbau, Leitungsführung, Leiterquerschnitt, zu verwendende Materialien und Mindestabstand zu beachten.

6.1 Standards zur CAN-Verdrahtung

Die Flexibilität beim Entwurf von CAN-Netzwerken ist eine wesentliche Stärke der verschiedenen, auf dem CAN-Standard ISO11898-2 aufbauenden Erweiterungen wie z.B. CANopen, ARINC825, DeviceNet und NMEA2000. Die Nutzung dieser Flexibilität erfordert jedoch zwingend ein Netzwerk-Design, das die Wechselwirkungen aller Netzwerkparameter berücksichtigt.

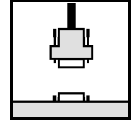
Zum Teil haben die CAN-Organisationen in ihren Spezifikationen den Einsatzbereich von CAN angepasst, um Anwendungen außerhalb des ISO 11898-Standards zu ermöglichen. Sie haben Einschränkungen auf Systemebene bei der Datenrate, der Leitungslänge und bei parasitären Bus-Lasten vorgenommen. Beim Entwurf von CAN-Netzwerken muss jedoch immer ein Spielraum für Signalverluste über das Gesamtsystem und die Verdrahtung, parasitäre Lasten, Netzwerk-Unsymmetrien, Potenzialunterschiede gegen Erde und Signal-Integritäten eingeplant werden. **Daher kann die maximal erreichbare Anzahl an Knoten, Buslängen und Stichleitungslängen von der theoretisch möglichen Anzahl abweichen!**

esd hat sich bei seinen Empfehlungen zur CAN-Verdrahtung auf die Vorgaben der ISO 11898-2 beschränkt. Auf die Beschreibung der Besonderheiten der abgeleiteten Spezifikationen CANopen, ARINC825, DeviceNet und NMEA2000 wird an dieser Stelle verzichtet.

Die konsequente Einhaltung der ISO 11898-2-Vorgaben bietet wesentliche Vorteile:

- Zuverlässiger Betrieb durch bewährte Design-Vorgaben
- Minimieren von Fehlerquellen durch ausreichend Abstand zu den physikalischen Grenzwerten.
- Unproblematische Wartung, weil bei zukünftigen Anpassungen und bei Fehlersuche keine „Sonderfälle“ zu berücksichtigen sind.

Selbstverständlich lassen sich auch zuverlässige Netzwerke nach den Spezifikationen von CANopen, ARINC825, DeviceNet und NMEA2000 aufbauen. **Zu beachten ist jedoch, dass die Verdrahtungsvorgaben der verschiedenen Spezifikationen nicht bedenkenlos miteinander vermischt werden dürfen!**



6.2 Leicht störbehaftete Industrieumgebung (zweiadrig verdrehte Leitung)

6.2.1 Grundregeln



ACHTUNG

esd garantiert die EU-Konformität des Produkts, wenn für die CAN-Verdrahtung mindestens Kabel mit einfach abgeschirmten **zweiadrig** verdrehten Leitungen verwendet werden, die die Anforderungen der Norm ISO 11898-2 erfüllen. Einfach abgeschirmte vieradrig verdrehte Leitungen, wie in Kapitel 6.3 beschrieben, stellen die EU-Konformität ebenfalls sicher.

Die folgenden Grundregeln für die CAN-Bus Verdrahtung mit einfach abgeschirmten zweiadrig verdrehten Leitungen sollten unbedingt beachtet werden:

1	Es ist ein geeigneter Leitungstyp (Wellenwiderstand ca. $120 \Omega \pm 10\%$) mit ausreichendem Leiterquerschnitt ($\geq 0.22 \text{ mm}^2$) zu verwenden. Der Spannungsabfall auf der Leitung ist zu berücksichtigen.
2	Für den Einsatz in leicht störbehafteter Industrieumgebung ist mindestens ein zweiadriges CAN-Kabel zu verwenden, dessen Adern wie folgt zu belegen sind: <ul style="list-style-type: none"> ● Zwei verdrehte Adern sind mit den CAN-Signalleitungen (CAN_H, CAN_L) zu belegen. ● Die Kabel-Abschirmung ist mit dem Bezugspotenzial (CAN_GND) zu belegen.
3	Das Bezugspotenzial CAN_GND muss an genau einem Punkt mit Funktionserde (FE) verbunden sein.
4	Ein CAN-Netz darf sich nicht verzweigen (Ausnahme: kurze Stichleitungen) und muss an beiden Enden mit dem Wellenwiderstand der Leitung (in der Regel $120 \Omega \pm 10\%$) abgeschlossen werden (zwischen den Signalen CAN_L und CAN_H und nicht an CAN_GND).
5	Stichleitungen sind so kurz wie möglich zu halten ($l < 0,3 \text{ m}$).
6	Die Baudrate muss an die Leitungslänge angepasst werden.
7	Die CAN-Leitungen sollten nicht in unmittelbarer Nähe von Störquellen verlegt werden. Lässt sich dies nicht vermeiden, so sind doppelt abgeschirmte Leitungen vorzuziehen.

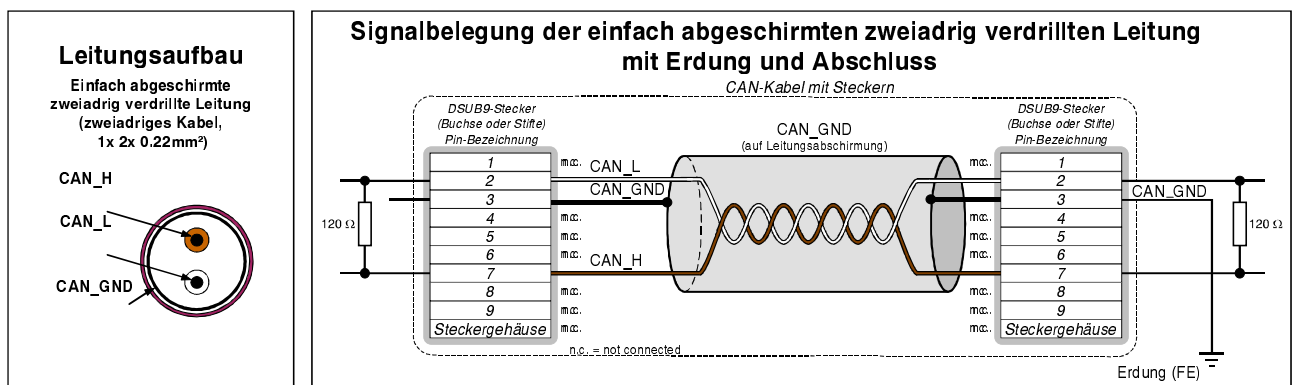
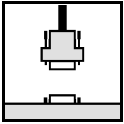


Abb. 14: CAN-Verdrahtung in leicht störbehafteter Industrieumgebung



Verdrahtungshinweise

6.2.2 Verkabelung

- Geräte, die pro CAN-Netz nur einen CAN-Stecker besitzen, sind über eine kurze Stichleitung ($< 0,3\text{ m}$) und ein T-Stück (als Zubehör lieferbar) anzuschließen. Werden sie am Ende des CAN-Netzes angeschlossen, kann auch der CAN-Abschlussstecker „CAN-Termination DSUB“ verwendet werden.

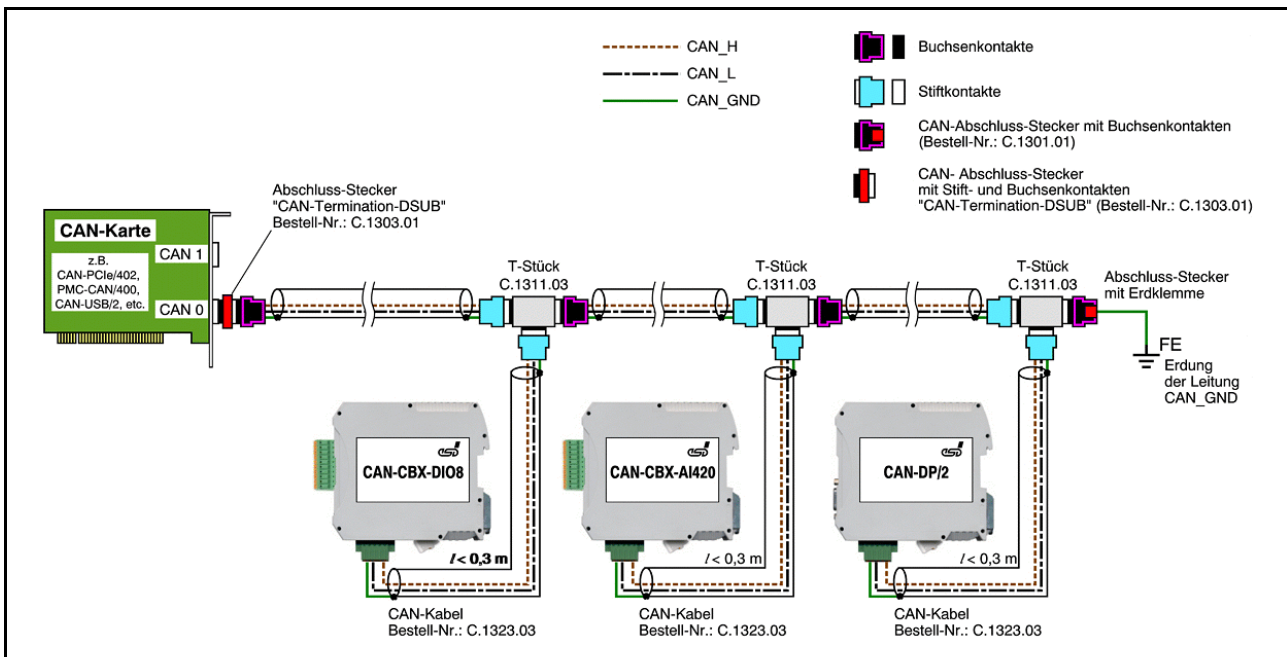
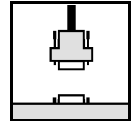


Abb. 15: Beispiel für korrekte Verdrahtung einfach abgeschirmter zweiadrig verdrehter Leitung

6.2.3 Abschlusswiderstand

- An beiden Enden des CAN-Bus ist ein Abschlusswiderstand anzuschließen. Ist auf dem CAN-Interface am Ende des CAN-Busses ein integrierter CAN-Busabschluss angeschlossen, so ist dieser an Stelle eines externen Abschlusswiderstands zu verwenden.
- 9-polige DSUB-Verbinder mit integriertem Abschlusswiderstand und Stift- und Buchsenkontakten sind unter der Artikelnummer C.1303.01 lieferbar.
- Für den Abschluss des CAN-Bus und Erdung des CAN_GND sind DSUB-Abschlussstecker mit Stiftkontakten (C.1302.01) oder Buchsenkontakten (C.1301.01) mit Erdungsklemme erhältlich.



6.3 Stark störbehaftete Industrieumgebung (vieradrig verdrillte Leitung)

6.3.1 Grundregeln

Die folgenden **Grundregeln** für die CAN-Bus Verdrahtung mit einfach abgeschirmten vieradrig verdrillten Leitungen sollten unbedingt beachtet werden:

1	Es ist ein geeigneter Leitungstyp (Wellenwiderstand ca. $120 \Omega \pm 10\%$) mit ausreichendem Leiterquerschnitt ($\geq 0.22 \text{ mm}^2$) zu verwenden. Der Spannungsabfall auf der Leitung ist zu berücksichtigen!
2	Für den Einsatz in stark störbehafteter Industrieumgebung ist ein vieradriges CAN-Kabel zu verwenden, dessen Adern wie folgt zu belegen sind: <ul style="list-style-type: none"> ● Zwei verdrillte Adern sind mit den CAN-Signalleitungen (CAN_H, CAN_L) zu belegen. ● Die anderen beiden verdrillten Adern sind mit dem Bezugspotenzial (CAN_GND) zu belegen. ● Die Leitungsabschirmung ist an mindestens einem Punkt an Funktionserde (FE) anzuschließen!
3	Das Bezugspotenzial CAN_GND muss an genau einem Punkt mit Funktionserde (FE) verbunden sein.
4	Ein CAN-Netz darf sich nicht verzweigen (Ausnahme: kurze Stichleitungen) und muss an beiden Enden mit dem Wellenwiderstand der Leitung (in der Regel $120 \Omega \pm 10\%$) abgeschlossen werden (zwischen den Signalen CAN_L und CAN_H und nicht an CAN_GND)!
5	Stichleitungen sind so kurz wie möglich zu halten ($l < 0,3 \text{ m}$)!
6	Die Baudrate muss an die Leitungslänge angepasst werden.
7	Die CAN-Leitungen sollten nicht in unmittelbarer Nähe von Störquellen verlegt werden. Lässt sich dies nicht vermeiden, so sind doppelt abgeschirmte Leitungen vorzuziehen.

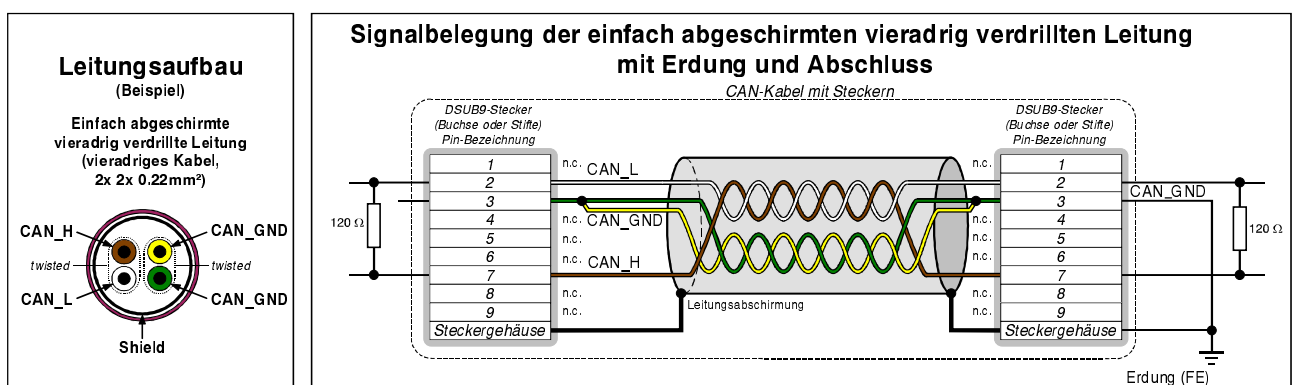
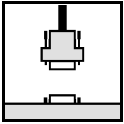


Abb. 16: CAN-Verdrahtung in stark störbehafteter Industrieumgebung



Verdrahtungshinweise

6.3.2 Verkabelung



ACHTUNG

Werden einfach abgeschirmte vieradrig verdrehte Leitungen verwendet, ist für den CAN-Bus Steckverbinder ein T-Verbindungsstecker zu verwenden, der den Anschluss zweier CAN-Kabel gestattet, und bei dem die Kabel-Abschirmung (Shield) durchgeführt wird, z.B. DSUB9-Stecker von ERNI (ERBIC CAN BUS MAX, Bestell-Nr.:154039).

Die Verwendung des esd T-Connectors (Bestell-Nr: C.1311.03) wird für einfach abgeschirmte vieradrig verdrehte Leitungen nicht empfohlen, da das Schirm-Potenzial des leitenden DSUB-Gehäuses nicht durch diesen T-Connector-Typ durchgeführt wird. Ist eine gemischte Verwendung von zwei- und vieradrig verdrehten Leitungen nicht zu vermeiden, ist auf eine durchgehende CAN-GND-Verbindung zu achten!

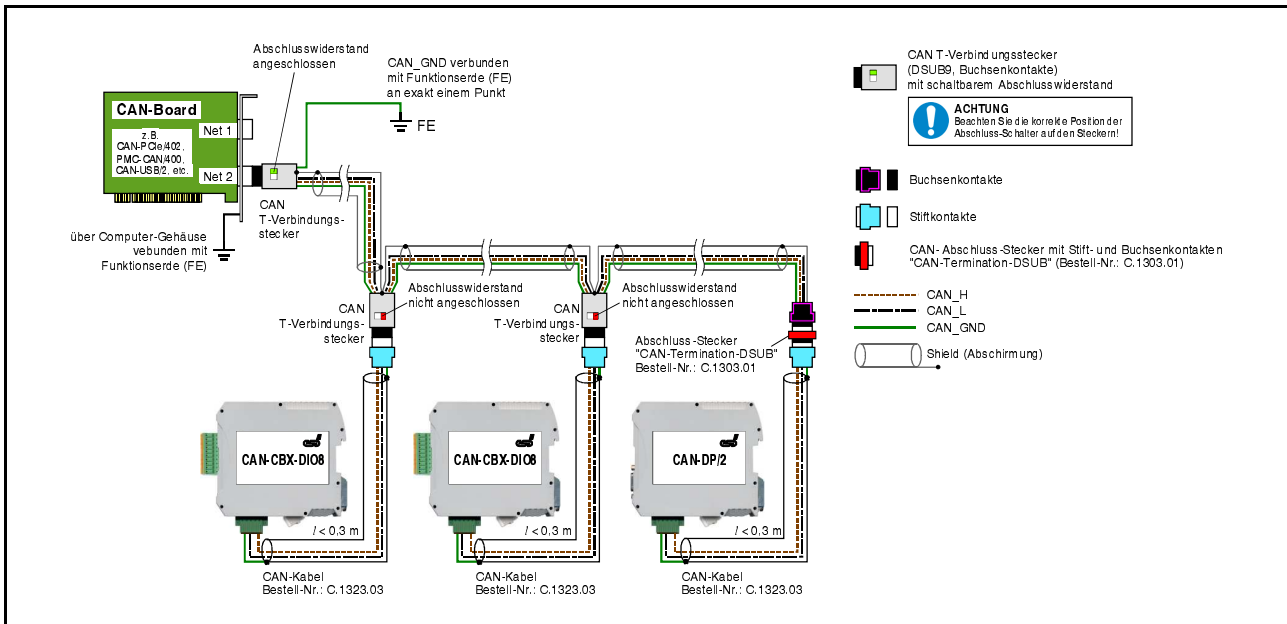
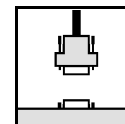


Abb. 17: Beispiel für korrekte Verdrahtung einfach abgeschirmter, vieradrig verdrehter Leitungen

6.3.3 Abschlusswiderstand

- An beiden Enden des CAN-Bus ist ein Abschlusswiderstand anzuschließen. Ist auf dem CAN-Interface am Ende des CAN-Busses ein integrierter CAN-Busabschluss angeschlossen, so ist dieser an Stelle eines externen Abschlusswiderstands zu verwenden.
- 9-polige DSUB-Verbinder mit integriertem Abschlusswiderstand und Stift- und Buchsenkontakten sind unter der Artikelnummer C.1303.01 lieferbar.
- 9-polige DSUB-Stecker mit integriertem, umschaltbarem Abschlusswiderstand können z.B. von ERNI (ERBIC CAN BUS MAX, Buchsenkontakte, Bestell-Nr.:154039) bezogen werden.



6.4 Erdung

- Bei CAN-Modulen mit galvanischer Trennung muss CAN_GND zwischen den CAN-Modulen verbunden werden.
- CAN_GND sollte an **exakt einem** Punkt im Netz mit dem Erdpotenzial (FE) verbunden sein.
- Jedes CAN-Modul mit galvanischer Verbindung zum Erdpotenzial wirkt wie eine Erdung. Aus diesem Grund sollte nicht mehr als ein CAN-Modul mit galvanischer Verbindung zum Erdpotenzial angeschlossen werden.
- Die Erdung kann z.B. an einem Abschlussstecker vorgenommen werden.

6.5 Buslänge



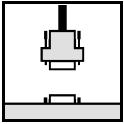
ACHTUNG

Beachten Sie, dass die Kabel, Stecker und Abschlusswiderstände gemäß ISO11898-2 auszulegen sind. Weitere Empfehlungen der CiA dazu, wie Standardwerte der Leitungsquerschnitte in Abhängigkeit von der Kabellänge, sind in der CiA Spezifikation CiA 303-1 beschrieben (siehe CiA 303 CANopen Recommendation - Part 1: „Cabling and connector pin assignment“, Version 1.8.0, Table 2).

Bit-Rate [kBit/s]	Theoretische Werte der erreichbaren Leitungslänge mit esd-Interface l_{\max} [m]	CiA-Empfehlungen (07/95) für erreichbare Leitungslängen l_{\min} [m]	Standardwerte der Leitungsquerschnitte nach CiA 303-1 [mm ²]
1000	37	25	0,25 bis 0,34
800	59	50	0,34 bis 0,6
666. $\bar{6}$	80	-	
500	130	100	
333. $\bar{3}$	180	-	
250	270	250	
166	420	-	0,5 bis 0,6
125	570	500	0,75 bis 0,8
100	710	650	
83. $\bar{3}$	850	-	
66. $\bar{6}$	1000	-	
50	1400	1000	
33. $\bar{3}$	2000	-	in CiA 303-1 nicht definiert
20	3600	2500	
12.5	5400	-	
10	7300	5000	

Tabelle 13: Erreichbare Leitungslängen in Abhängigkeit von der Bitrate (mit esd-CAN-Interfaces)

- Optokoppler verzögern die CAN-Signale. esd-Module erreichen typischerweise eine Leitungslänge von 37 m bei 1 MBit/s. Voraussetzung hierfür ist ein abgeschlossenes Netz ohne Impedanzstörungen, wie z.B. längere Stichleitungen > 0.3 m.



Verdrahtungshinweise

6.6 Beispiele für CAN-Kabel

Die folgenden zwei- und vieradrigen Kabel empfiehlt esd für den Aufbau von CAN-Netzwerken. Diese Kabeltypen werden auch für die bei esd erhältlichen CAN-Kabel verwendet.

6.6.1 Kabel für leicht störbehaftete Industrieumgebung (zweiadrig)

Hersteller	Leitungstyp
U.I. LAPP GmbH Schulze-Delitzsch-Straße 25 70565 Stuttgart Germany www.lappkabel.de	z.B. UNITRONIC ®-BUS CAN UL/CSA (1x 2x 0.22) (UL/CSA approved) Bestell-Nr.: 2170260
	UNITRONIC ®-BUS-FD P CAN UL/CSA (1x 2x 0.25) (UL/CSA approved) Bestell-Nr.: 2170272
ConCab GmbH Äußerer Eichwald 74535 Mainhardt Germany www.concab.de	z.B. BUS-PVC-C (1x 2x 0,22 mm ²) Bestell-Nr.: 93 022 016 (UL appr.)
	BUS-Schleppflex-PUR-C (1x 2x 0,25 mm ²) Bestell-Nr.: 94 025 016 (UL appr.)

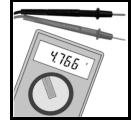
6.6.2 Kabel für stark störbehaftete Industrieumgebung (vieradrig)

Hersteller	Leitungstyp
U.I. LAPP GmbH Schulze-Delitzsch-Straße 25 70565 Stuttgart Germany www.lappkabel.de	z.B. UNITRONIC ®-BUS CAN UL/CSA (2x 2x 0.22) (UL/CSA approved) Bestell-Nr.: 2170261
	UNITRONIC ®-BUS-FD P CAN UL/CSA (2x 2x 0.25) (UL/CSA approved) Bestell-Nr.: 2170273
ConCab GmbH Äußerer Eichwald 74535 Mainhardt Germany www.concab.de	z.B. BUS-PVC-C (2x 2x 0,22 mm ²) Bestell-Nr.: 93 022 026 (UL appr.)
	BUS-Schleppflex-PUR-C (2x 2x 0,25 mm ²) Bestell-Nr.: 94 025 026 (UL appr.)



HINWEIS

Fertig konfektionierte Kabel in Standard- und Sonderlängen können bei **esd** bezogen werden.



7. CAN-Bus Troubleshooting Guide

Der CAN-Bus Troubleshooting Guide ist eine Anleitung zum Auffinden und Beseitigen der häufigsten Hardware-Fehlerursachen in der CAN-Bus-Verdrahtung.

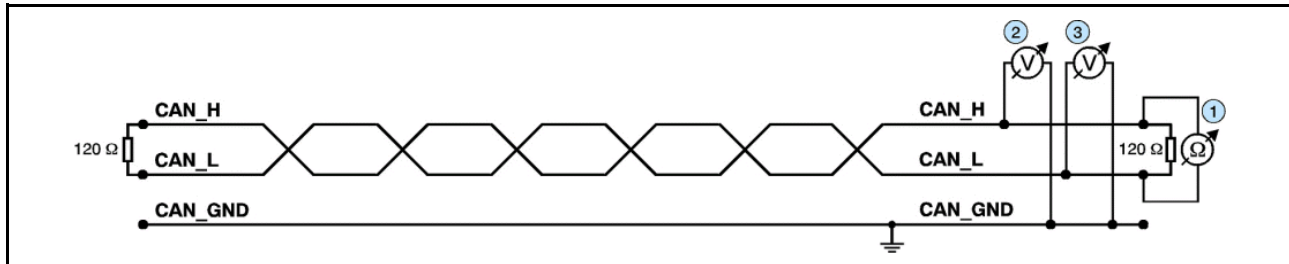


Abb. 18: Vereinfachtes Schaltbild eines CAN-Netzwerks

7.1 Bus-Abschluss

Der Bus-Abschluss wird verwendet, um den Widerstand eines Knotens an den Widerstand der verwendeten Busleitung anzupassen. Ist die Impedanz falsch angepasst, wird das gesendete Signal nicht ganz von der Last aufgenommen und zum Teil in die Übertragungsleitung zurück reflektiert. Sind die Quellen-, Übertragungsleitungs- und Last-Impedanz gleich groß, so werden die Reflexionen vermieden. Dieser Test misst den Gesamtwiderstand der beiden CAN-Datenleitungen und des angeschlossenen Abschlusswiderstandes.

Zum Testen verfahren Sie bitte wie folgt:

1. Schalten Sie die Versorgungsspannungen aller angeschlossenen CAN-Knoten aus.
2. Messen Sie den DC-Widerstand zwischen CAN_H und CAN_L an einem Ende des Netzwerks (1) (siehe obere Abbildung).

Der gemessene Wert sollte zwischen 50 Ω und 70 Ω liegen.

Liegt der ermittelte Wert unter 50 Ω , stellen Sie bitte sicher, dass:

- kein **Kurzschluss** zwischen den CAN_H- und CAN_L-Leitungen besteht
- **nicht mehr als zwei** Abschlusswiderstände angeschlossen sind
- die Transceiver der einzelnen Knoten nicht defekt sind.

Liegt der ermittelte Wert über 70 Ω , stellen Sie bitte sicher, dass:

- alle CAN_H- und CAN_L- Leitungen korrekt angeschlossen sind
- zwei Abschlusswiderstände von **je 120 Ω** an Ihr CAN-Netzwerk angeschlossen sind (einer an jedem Ende).



7.2 Erdung

CAN_GND des CAN-Netzwerks sollte nur an einer einzigen Stelle mit dem Funktionserde-Potenzial (FE) verbunden sein. Dieser Test zeigt an, ob die Abschirmung an mehreren Stellen geerdet ist. Zum Testen verfahren Sie bitte wie folgt:

1. Trennen Sie die CAN_GND vom Erdpotenzial (FE).
2. Messen Sie den DC-Widerstand zwischen CAN_GND und Erdpotenzial (siehe nebenstehende Abbildung).
3. Verbinden Sie CAN_GND wieder mit dem Erdpotenzial.

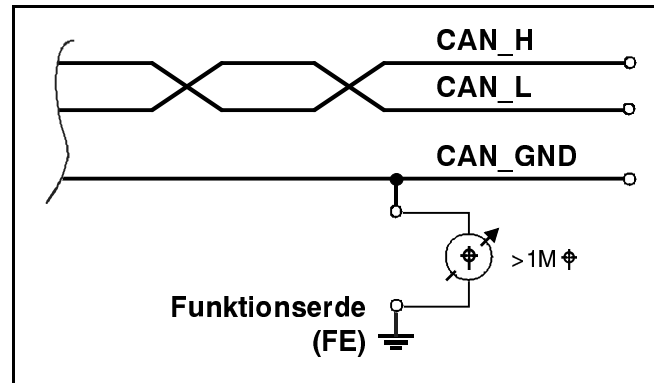


Abb. 19: Vereinfachtes Schaltbild Erdungsmessung

Der gemessene Widerstand sollte größer als ein $1\text{ M}\Omega$ sein. Ist er kleiner, suchen Sie bitte nach zusätzlichen Erdungen der CAN_GND-Leitung.

7.3 Kurzschluss in der CAN-Verdrahtung

Ein CAN-Bus kann möglicherweise auch dann noch Daten übertragen, wenn CAN_GND und CAN_L kurzgeschlossen sind. Dadurch wird aber in der Regel die Fehlerrate stark ansteigen. Stellen Sie sicher, dass zwischen CAN_GND und CAN_L kein Kurzschluss besteht!

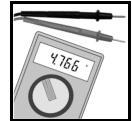
7.4 CAN_H/CAN_L-Spannungen

Jeder Knoten verfügt über einen CAN-Transceiver, der differenzielle Signale auf den Datenleitungen generiert. Ruht die Netzwerk-Kommunikation, betragen die CAN_H- und CAN_L-Spannungen etwa 2,5 V zu CAN_GND. Defekte Transceiver können diese Ruhespannungen verändern und die Netzwerk-Kommunikation unterbrechen.

Um auf defekte Transceiver zu testen, verfahren Sie bitte wie folgt:

1. Schalten Sie alle Versorgungsspannungen an.
2. Beenden sie jegliche Netzwerk-Kommunikation.
3. Messen Sie die DC-Spannung zwischen CAN_H und GND ② (siehe Abbildung auf vorhergehender Seite).
4. Messen Sie die DC-Spannung zwischen CAN_L und GND ③ (siehe Abbildung auf vorhergehender Seite).

Die gemessene Spannung sollte zwischen 2,0 V und 3,0 V liegen.



Ist die Spannung kleiner als 2,0 V oder größer als 3,0 V, ist es möglich, dass ein oder mehrere Knoten defekte Transceiver haben. Bei einer Spannung, die unter 2,0 V liegt, überprüfen Sie bitte den Anschluss der CAN_H- und CAN_L-Leitungen.

Um in einem Netzwerk einen Knoten mit einem defekten Transceiver zu finden, überprüfen Sie bitte einzeln die Widerstände der CAN-Transceiver der Knoten (siehe folgendes Kapitel).

7.5 CAN-Transceiver Widerstandstest

CAN-Transceiver verfügen über Schaltkreise, die CAN_H und CAN_L kontrollieren. Die Erfahrung zeigt, dass elektrische Beschädigungen den Leckstrom in diesen Schaltkreisen erhöhen können. Um den Leckstrom durch die CAN-Schaltungen zu messen, benutzen Sie bitte ein Widerstandsmessgerät und:

1. Schalten Sie den Knoten aus und trennen Sie ihn vom Netzwerk ④ (siehe untere Abbildung).
2. Messen Sie den DC-Widerstand zwischen CAN_H und CAN_GND ⑤ (siehe untere Abbildung).
3. Messen Sie den DC-Widerstand zwischen CAN_L und CAN_GND ⑥ (siehe untere Abbildung).

Der gemessene Widerstand sollte bei jeder Messung etwa $500\text{ k}\Omega$ betragen. Liegt der Widerstand deutlich niedriger, ist der CAN-Transceiver möglicherweise defekt.

Ein weiterer Hinweis auf einen fehlerhaften CAN-Transceiver ist eine sehr hohe Abweichung der beiden gemessenen Eingangswiderstände ($\gg 200\%$).

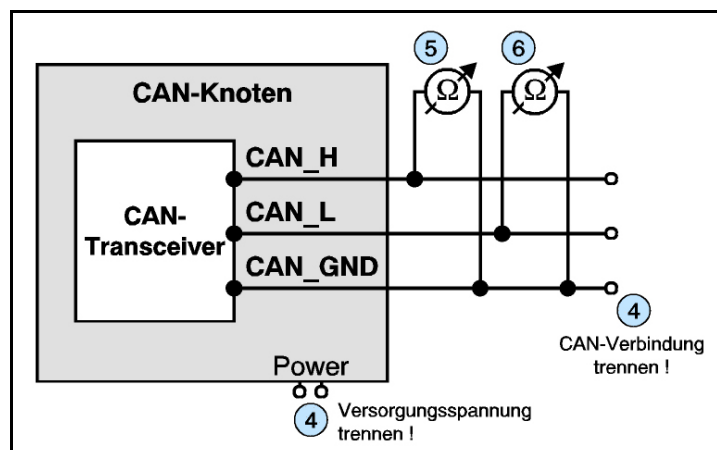


Abb. 20: Messung des Eingangswiderstandes des CAN-Transceivers

7.6 Support bei esd

Sollten Sie trotz Anwendung des CAN-Bus Troubleshooting Guides zu keiner Lösung kommen, wenden Sie sich bitte an unseren Support unter support@esd.eu oder Tel. **0511-37298-130**



8. CANopen-Firmware

Dieses Kapitel enthält neben allgemeinen Grundlagen zum Thema CANopen die wichtigsten Informationen über die implementierten Funktionen.

Eine komplette CANopen-Beschreibung läge außerhalb des Rahmens dieses Handbuchs. Weitergehende Informationen sind daher der CANopen - Dokumentation [1] zu entnehmen.

8.1 Begriffsdefinition

COB ...	Communication Object	
Emergency-Id...	Emergency Data Object	Notfalldaten
NMT...	Network Management (Master)	Netzwerkmanagement
SDO...	Service Data Object	Service-Daten (Parameter)
Sync...	Sync(frame) Telegramm	Synchronisations-Identifizier

PDOs (Process Data Objects)

Die PDOs dienen zur Übertragung der Prozessdaten.

Im 'Transmit'-PDO (TPDO) sendet das CAN-CBX-Modul Daten auf dem CANopen-Netz.

Im 'Receive'-PDO (RPDO) empfängt das CAN-CBX-Modul Daten aus dem CANopen-Netz.

SDOs (Service Data Objects)

Die SDOs dienen zur Übertragung von modulinternen Konfigurations- und Parameterdaten. Im Gegensatz zu den PDOs werden die SDO-Nachrichten bestätigt. Einer Schreib- oder Leseanforderung auf ein Daten-Objekt folgt immer ein Bestätigungstelegramm mit einem Fehlerindex.



8.2 NMT-Boot-up

Das CAN-CBX-Modul kann mit dem in [1] beschriebenen ‘Minimal - Boot-up’ initialisiert werden. (Ab Firmware-Version 1.02 oder aktueller)

Im einfachsten Fall reicht nach dem Einschalten ein Telegramm zum Umschalten aus dem Zustand *Pre-Operational* in den Zustand *Operational*. Dazu ist an den CAN-Identifizier ‘0000_h’ z. B. das 2-Byte-Telegramm ‘01_h’, ‘00_h’ zu senden (= Start Remote Node all Devices).

8.3 Das CANopen-Objektverzeichnis

Das Objektverzeichnis ist eine (sortierte) Gruppierung von Objekten, auf die über das CAN-Netzwerk zugegriffen werden kann. Jedes Objekt in diesem Verzeichnis wird über einen 16-Bit-Index adressiert. In den Objektverzeichnissen wird der Index in hexadezimaler Form angegeben.

Der Index kann 16-Bit-Parameter nach der CANopen-Spezifikation [1] oder ein herstellerspezifischer Code sein. Anhand der höherwertigen Bits des Index wird festgelegt, zu welcher Objektklasse der Parameter gehört.

Zum Objektverzeichnis gehören unter anderem:

Index	Objekt
0001 _h ... 009F _h	Definition von Datentypen
1000 _h ... 1FFF _h	Communication Profile Area
2000 _h ... 5FFF _h	Manufacturer Specific Profile Area
6000 _h ... 9FFF _h	Standardised Device Profile Area
A000 _h ... FFFF _h	reserviert



8.4 Communication Parameter

Die Communication Parameter der PDOs (gemäß [1]) werden als SDOs (Service Data Objects) auf der ID '600_h + NodeID' übertragen (Request). Der Empfänger quittiert die Parameter auf der ID '580_h + NodeID' (Response).

Die **Node-ID** (Modul-Nr.) wird über die Kodierschalter Low und High konfiguriert. Die möglichen Einstellungen sind in Kapitel "Einstellung der Node-ID über die Kodierschalter" auf Seite 20 beschrieben.

8.4.1 Zugriff auf das Objektverzeichnis über SDOs

Die SDOs (Service Data Objects) dienen zum Zugriff auf das Objektverzeichnis eines Gerätes. Ein SDO stellt daher einen 'Kanal' zum Zugriff auf die Parameter des Gerätes dar. Der Zugriff über diesen Kanal ist im Zustand *operational* und *pre-operational* möglich.

Dieses Kapitel beschreibt nicht alle möglichen sondern nur einige wichtige Arten des Zugriffs im CAN-CBX-Modul.

Definitionen der Zugriffsmodi können [1] entnommen werden.

Ein SDOCAN Frame ist wie folgt aufgebaut:

Identifizier	Befehls-code	Index		Sub-Index	LSB	Datenfeld	MSB
		(low)	(high)				

Beispiel:

600 _h + NodeID	23 _h (write)	00 _h	14 _h	01 _h	7F _h	04 _h	00 _h	00 _h
		(Index=1400 _h) (Receive-PDO-Comm-Para)		(COB-def.)	COB Node ID = 047F _h			

Beschreibung der SDOs:

Identifizier

Die Parameter werden auf der ID '600_h + NodeID' übertragen (Request). Der Empfänger quittiert die Parameter auf der ID '580_h + NodeID' (Response).

Befehlscode

Der gesendete Befehlscode setzt sich unter anderem aus dem Command Specifier und der Länge zusammen. Häufig benötigte Kombinationen sind z.B.:

40_h = 64_{dez} : Read Request, d.h. ein Parameter soll gelesen werden

23_h = 35_{dez} : Write Request mit 32 Bit Daten, d.h. ein Parameter soll gesetzt werden



Das CAN-CBX-Modul antwortet auf jedes empfangene Telegramm mit einem Antworttelegramm. Das Antworttelegramm kann folgende Befehlscodes enthalten:

- $43_{\text{h}} = 67_{\text{dez}}$: Read Response mit 32 Bit Daten, dieses Telegramm enthält den gewünschten Parameter
 $60_{\text{h}} = 96_{\text{dez}}$: Write Response, d.h. ein Parameter wurde erfolgreich gesetzt
 $80_{\text{h}} = 128_{\text{dez}}$: Error Response, d.h. das CAN-CBX-Modul meldet einen Fehler.

Häufig verwendete Befehlscodes

Die folgende Tabelle gibt eine Übersicht über häufig verwendete Befehlscodes. Die Kommando-Frames müssen immer 8 Bytes beinhalten. Hinweise zur Syntax und weitere Befehlscodes sind in [1] nachzulesen.

Kommando	für Anzahl Datenbytes	Befehlscode
Write Request (Initiate Domain Download)	1	$2F_{\text{h}}$
	2	$2B_{\text{h}}$
	3	27_{h}
	4	23_{h}
Write Response (Initiate Domain Download)	-	60_{h}
Read Request (Initiate Domain Upload)	-	40_{h}
Read Response (Initiate Domain Upload)	1	$4F_{\text{h}}$
	2	$4B_{\text{h}}$
	3	47_{h}
	4	43_{h}
Error Response (Abort Domain Transfer)	-	80_{h}

Index, Sub-Index

Der Index und der Sub-Index werden in den Kapiteln "Device Profile Area" und "Manufacturer Specific Profile Area" dieses Handbuches beschrieben.

Datenfeld

Das maximal 4 Byte lange Datenfeld ist grundsätzlich nach der Regel 'niederwertiges Byte zuerst, höherwertiges Byte zuletzt' aufgebaut. Dabei steht das niederwertige Byte immer in 'Data 1', bei 16-Bit-Werten steht das höchstwertige Byte (Bits 8...15) in 'Data 2', und bei 32-Bit-Werten steht das MSB (Bits 24...31) in 'Data 4'.



Fehlercodes des SDO-Transfers

Die folgenden SDO Abort Codes (Fehlercodes) können zur Anwendung kommen (gemäß [1]):

SDO Abort Codes	Erläuterung
05040001 _h	falscher Command Specifier
06010002 _h	Schreibzugriff ist hier falsch
06020000 _h	falscher Index
06040041 _h	Objekt kann nicht auf PDO abgebildet werden
06060000 _h	kein Zugriff wegen Hardware Fehler
06070010 _h	falsche Anzahl Daten-Bytes
06070012 _h	Länge des Service-Parameters ist zu groß
06070013 _h	Länge des Service-Parameters ist zu klein
06090011 _h	falscher Sub-Index
06090030 _h	gesendeter Parameter außerhalb des zulässigen Wertebereiches
08000000 _h	nicht definierte Fehlerursache
08000020 _h	Daten können nicht übertragen oder gespeichert werden
08000022 _h	wegen des aktuellen Device States können Daten nicht übertragen oder gespeichert werden
08000024 _h	Zugriff auf Flash fehlgeschlagen

8.4.2 Nichtflüchtiges Speichern von Parametern im EEPROM

Die eingestellten Parameter sind sofort nach der Übergabe aktiv. Das ‘nichtflüchtige’ Speichern der Parameter erfolgt jedoch nicht automatisch. Es muss über einen Schreibzugriff auf das Objekt 1010_h veranlasst werden und sollte nur erfolgen, wenn sich das Modul im Zustand *pre-operational* befindet.

Der Speicher-Modus wird über den Inhalt des Objektes 1010_h angezeigt:

Bit 1 des Objekts 1010_h, Subindex 1 ist nicht gesetzt, d.h. das CAN-CBX-DIO8-Modul speichert die Konfiguration nicht ‘automatisch’, sondern muss durch Schreiben der Zeichenkette ‘save’ (73_h 61_h 76_h 65_h, Reihenfolge aus CAN-Telegramm) in das Objekt 1010_h, Subindex 1 extra dazu veranlasst werden.



8.5 Übersicht der verwendeten CANopen-Identifizier

Funktion	Identifizier	Bemerkungen
Netzwerkmanagement	0 _h	NMT
SYNC	80 _h	Sync an alle, (konfigurierbar über Objekt 1005 _h)
Emergency Message	80 _h + <i>NodeID</i>	konfigurierbar über Objekt 1014 _h
Client-PDO	180 _h + <i>NodeID</i>	PDO vom CAN-CBX-DIO8 (Sende-PDO) (Objekt 1800 _h)
Server-PDO	200 _h + <i>NodeID</i>	PDO zum CAN-CBX-DIO8 (Empfangs-PDO) (Objekt 1400 _h)
Client-SDO	580 _h + <i>Node-ID</i>	SDO vom CAN-CBX-DIO8
Server-SDO	600 _h + <i>Node-ID</i>	SDO zum CAN-CBX-DIO8
Node Guarding	700 _h + <i>NodeID</i>	konfigurierbar über Objekt 100E _h

NodeID: Eingestellte CANopen-Adresse [1_h...7F_h]

8.5.1 Einstellung der COB-ID

Die COB-IDs die einstellbar sind (außer der von SYNC), werden zunächst von der Einstellung der Node-ID über die Kodierschalter (siehe Seite 20) abgeleitet. Wurden die COB-IDs über SDO beschrieben, gilt diese Einstellung, auch wenn der Schalter danach auf eine andere Node-ID eingestellt wird.

Um die Default-COB-ID wieder vom Schalter zu übernehmen müssen die *Comm defaults* oder alle Defaults geladen werden (Objekt 1011_h)



8.6 PDO-Belegung

Die PDOs (Process Data Objects) dienen zur Übertragung der Prozessdaten:

CAN-Identifizier	Länge	Übertragungsrichtung	Belegung
200 _h + Node-ID	1 Byte	zum CAN-CBX-DIO8 (Empfangs-PDO)	Setzen der Ausgänge
180 _h + Node-ID	1 Byte	vom CAN-CBX-DIO8 (Sende-PDO)	Abfragen des Status der Eingänge

RPDO1 (-> CAN-CBX-DIO8)

CAN-Identifizier: 200_h + Node-ID

Byte	0	1	2	3	4	5	6	7
Parameter	<i>Write_Output_ DO8-DO1</i>	-	-	-	-	-	-	-

Parameter-Beschreibung:

Bezeichner	Beschreibung	Datentyp	siehe Seite
<i>Write_Output_DO8-DO1</i>	Setzen der digitalen Ausgänge	Byte	103

TPDO1 (CAN-CBX-DIO8->)

CAN-Identifizier: 180_h + Node-ID

Byte	0	1	2	3	4	5	6	7
Parameter	<i>Read_Input_ DI8-DI1</i>	-	-	-	-	-	-	-

Parameter-Beschreibung:

Bezeichner	Beschreibung	Datentyp	siehe Seite
<i>Read_Input_DI8-DI1</i>	Zustand der digitalen Eingänge	Byte	96



8.7 Setzen und Lesen der Aus- und Eingänge

8.7.1 Meldung der digitalen Eingänge

Die Übertragungsarten für die digitalen Eingänge können unterschieden werden in:

- *azyklisch, synchron*: Die Sendung erfolgt nach dem Empfang einer SYNC-Message (PDO - Übertragungstyp 0), wenn sich die Daten geändert haben.
- *zyklisch, synchron*: Die Sendung erfolgt, nachdem jeweils eine bestimmte Anzahl SYNC-Message empfangen wurden (PDO-Übertragungstyp 1...240).
- *synchron, remote request*: Der Zustand der Eingänge wird mit jeder SYNC-Message gespeichert und nach dem Empfang eines RTR-Frames gesendet (PDO-Übertragungstyp 252).
- *asynchron, remote request*: Nach dem Empfang eines RTR-Frames wird der letzte ermittelte Zustand der Eingänge gesendet (PDO-Übertragungstyp 253).
- *ereignisgesteuert, asynchron*: Die Sendung erfolgt, wenn sich der Zustand bestimmter digitaler Eingänge ändert (PDO-Übertragungstyp 254, 255).

8.7.2 Digitale Ausgänge

Die digitalen Ausgänge werden gesetzt, sobald ein Objekt zum Setzen der Ausgänge von der CAN-CBX-DIO8 empfangen wurde (z.B. Objekt 6200_n über RPDO).

8.7.3 Unterstützte Übertragungsarten nach DS-301

Transmission Type	PDO-Transmission					unterstützt von CAN-CBX-DIO8
	cyclic	acyclic	synchronous	asynchronous	RTR	
0		X	X			x
1...240	X		X			x
241...251	reserviert					-
252			X		X	x
253				X	X	x
254				X	X	x
255				X	X	x

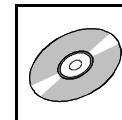


8.8 Communication Profile Area

8.8.1 Verwendete Bezeichnungen und Abkürzungen

Folgende Bezeichnungen werden in den Tabellen zur Beschreibung der Kommunikationsparameter verwendet:

PDO-Mappable	Für diesen Sub-Index des PDOs ist PDO-Mapping möglich
Save to EEPROM	Der Wert dieses Parameters wird im lokalen EEPROM gespeichert, wenn das Kommando 'save' aufgerufen wird (siehe Seite 67)
Datentyp	Datentyp (z.B. unsigned 8, unsigned 32)
Zugriff	Zugelassene Zugriffsarten auf diesen Parameter ro... read only Dieser Parameter kann nur gelesen werden. Schreibzugriffe führen zu einer Fehlermeldung. const... constant Dieser Parameter kann vom Anwender nicht verändert werden. Er ist Lesbar. Schreibzugriffe führen zu einer Fehlermeldung. rw... read / write Dieser Parameter kann gelesen oder gesetzt werden.
Wertebereich	Wertebereich des Parameters
Default-Wert	Grundeinstellung des Parameters
Name	Name und Kurzbeschreibung des Parameters

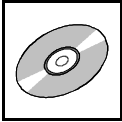


8.9 Implementierte CANopen-Objekte

Eine detaillierte Beschreibung der Objekte ist in CiA 301 [1] nachzulesen.

8.9.1 Übersicht der 1000er-Objekte mit Default-Werten

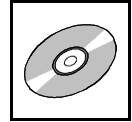
Index	Sub-index (max.)	Name	Datentyp	Zugriff	Produktspezifische Eigenschaften
1000 _h	-	Device Type	unsigned 32	ro	default: 00030191 _h
1001 _h	-	Error Register	unsigned 8	ro	Unterstützte Error Bits: 0: generic 1: current 2: voltage 4: communication error
1003 _h	10	Pre-Defined-Error-Field	unsigned32	rw	default: 00 _h
1005 _h	-	COB-ID-Sync	unsigned32	rw	default: 80 _h
1006 _h	-	Communication Cycle Period	unsigned32	rw	default: 00 _h
1008 _h	-	Manufacturer Device Name	visible string	ro	default: "CAN-CBX-DIO8"
1009 _h	-	Manufacturer Hardware Version	visible string	ro	default ist Versionsabhängig
100A _h	-	Manufacturer Software Version	visible string	ro	default ist Versionsabhängig
100C _h	-	Guard Time	unsigned 16	rw	default: 0000 _h
100D _h	-	Life Time Factor	unsigned 8	rw	default: 00 _h
100E _h	-	Node Guarding Identifier	unsigned 32	rw	Node-ID + 700 _h
1010 _h	4	Store Parameter	unsigned 32	rw	n.a.
1011 _h	3	Restore Parameter	unsigned 32	rw	n.a.
1014 _h	-	COB-ID Emergency Object	unsigned 32	rw	default: 80 _h + Node-ID
1015 _h	-	Inhibit Time EMCY	unsigned 16	rw	default: 00 _h
1016 _h	1	Consumer Heartbeat Time	unsigned 32	rw	default: 00 _h
1017 _h	-	Producer Heartbeat Time	unsigned 16	rw	default: 00 _h
1018 _h	4	Identity Object	unsigned 32	ro	default: Vendor Id: 00000017 _h Prod. Code: 23010002 _h
1019 _h	-	Synchronous Counter Overflow	unsigned 8	rw	default: 00 _h
1020 _h	2	Verify Configuration	unsigned 32	ro	n.a.
1029 _h	6	Error Behaviour	unsigned 8	ro	default: 00 _h



Implementierte CANopen Objekte

Index	Sub-index	Beschreibung	Datentyp	Zugriff
1400 _h	2	1. Receive PDO-Parameter	PDO CommPar (20 _h)	rw
1600 _h	1	1. Receive PDO-Mapping	PDO Mapping (21 _h)	ro
1800 _h	5	1. Transmit PDO-Parameter	PDO CommPar (20 _h)	rw
1801 _h	5	2. Transmit PDO-Parameter	PDO CommPar (20 _h)	rw
1A00 _h	1	1. Transmit PDO-Mapping	PDO Mapping (21 _h)	ro
1A01 _h	4	2. Transmit PDO-Mapping	PDO Mapping (21 _h)	ro

Index	Sub-index (max.)	Beschreibung	Datentyp	Zugriff	Produktspezifische Eigenschaften
1F80 _h	-	NMT startup	unsigned 32	rw	default: 2 (autostart disabled)
1F91 _h	1	Self starting nodes timing parameters	unsigned 16	rw	default: 64 _h (= 100 ms)



8.9.2 Device Type (1000_h)

INDEX	1000_h
Name	<i>device type</i>
Datentyp	unsigned 32
Zugriff	ro
Default-Wert	0003 0191

Der Wert des *device type* beträgt: 0003.0191_h (Digital Input Output: 0003_h
 Digital Profile Number: 0191_h)

Beispiel: Auslesen des Device Type

Der CANopen Master sendet unter dem Identifier '603_h' (600_h + Node-ID) den Read Request an das CAN-CBX-DIO8-Modul mit der Modul-Nr. 3 (Node-ID=3_h):

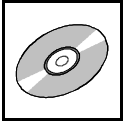
ID	RTR	LEN	DATA								
			1	2	3	4	5	6	7	8	
603 _h	0 _h	8 _h	40 _h Read Request	00 _h	10 _h Index=1000 _h	00 _h	00 _h Sub Index	00 _h	00 _h	00 _h	00 _h

Das CAN-CBX-DIO8-Modul Nr. 3 antwortet dem Master anhand der Read Response unter dem Identifier '583_h' (580_h + Node-ID) mit dem Wert des Device Type:

ID	RTR	LEN	DATA							
			1	2	3	4	5	6	7	8
583 _h	0 _h	8 _h	43 _h Read Response	00 _h	10 _h Index=1000 _h	00 _h	91 _h	01 _h dig. Profile Nr.191	03 _h	00 _h Digital Output

ausgebener Wert des Device Type: 0003.0191_h

Das Datenfeld ist grundsätzlich nach der Regel 'niederwertiges Byte zuerst, höherwertiges Byte zuletzt' aufgebaut (siehe Seite 49, Datenfeld).



Implementierte CANopen Objekte

8.9.3 Error Register (1001_h)

Das CAN-CBX-DIO8-Modul nutzt das Error-Register, um Fehlermeldungen anzuzeigen.

INDEX	1001_h
Name	<i>error register</i>
Datentyp	unsigned 8
Zugriff	ro
Default-Wert	0

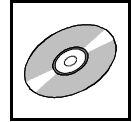
Folgende Bits des Error-Registers werden zur Zeit unterstützt:

Bit	Bedeutung
0	<i>generic</i>
1	<i>current</i>
2	<i>voltage</i>
3	-
4	<i>communication error (overrun, error state)</i>
5	-
6	-
7	-

Die nicht unterstützten Bits (-) werden immer als '0' zurückgegeben. Liegt ein Fehler vor, so ist das entsprechende Fehler-Bit auf '1' gesetzt.

Folgende Fehlermeldungen werden zur Zeit unterstützt:

- 00_h keine Fehler
- 01_h generic error
- 02_h current error
- 04_h voltage error
- 10_h communication error



8.9.4 Pre-defined Error Field (1003_h)

INDEX	1003_h
Name	<i>pre-defined error field</i>
Datentyp	unsigned 32
Zugriff	ro
Default-Wert	No

Im *pre-defined error field* wird eine Liste der zuletzt aufgetretenen Fehler gespeichert. Der Subindex 0 enthält die aktuelle Anzahl der in der Liste gespeicherten Fehler. Unter Subindex 1 wird der zuletzt aufgetretene Fehler abgelegt. Tritt ein neuer Fehler auf, so wird der vorhergehende Fehler auf Subindex 2 gespeichert und der neue Fehler unter Subindex 1 usw.. So entsteht eine Liste mit der Fehler-Historie. Der Fehlerspeicher ist wie ein Ringspeicher aufgebaut. Ist er gefüllt, wird der älteste Eintrag gelöscht um Platz für den aktuellen Eintrag zu schaffen.

Dieses Modul unterstützt maximal 10 Fehlereinträge. Beim Eintreten des 11. Fehlers wird der älteste Fehlereintrag gelöscht. Zum Löschen der gesamten Fehler-Liste ist der Subindex '0' auf '0' zu setzen. Dies ist der einzig zulässige Schreibzugriff auf das Objekt.

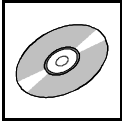
Mit jedem neuen Eintrag in die Liste sendet das Modul ein **Emergency-Frame**, um den Fehler mitzuteilen.

Index	Subindex	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1003_h	0	<i>no_of_errors_in_list</i>	0, 1...10	-	unsigned 8	rw
	1	<i>error-code n</i>	0...FFFFFFFF _h	-	unsigned 32	ro
	2	<i>error-code (n-1)</i>	0...FFFFFFFF _h	-	unsigned 32	ro
	:	:	:	:	:	ro
	10	<i>error-code (n-9)</i>	0...FFFFFFFF _h	-	unsigned 32	ro

Die Bedeutung der Variablen:

no_of_errors_in_list - enthält die aktuelle Anzahl der in der Liste eingetragenen Fehler-Codes
n = Nummer des zuletzt aufgetretenen Fehlers
 - zum Löschen der Fehlerliste ist diese Variable auf '0' zu setzen
 - ist *no_of_errors_in_list* ≠ 0, so wird das Error-Register (Object 1001_h) gesetzt

error-code x Der 32-Bit lange Fehler-Code setzt sich aus dem in CiA 301 [1] aufgeführten CANopen-Emergency-Error-Code und den von esd definierten Fehler-Codes (Manufacturer-Specific Error Field) zusammen.



Implementierte CANopen Objekte

Bit:	31 16	15 0
Inhalt:	<i>manufacturer-specific error field</i>		<i>emergency-error-code</i>	

manufacturer-specific error field: immer '00', es sei denn
emergency-error-code = 2300_h oder 2320_h (siehe unten)

emergency-error-code: Es werden die folgenden Fehler-Codes unterstützt:

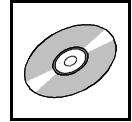
- 8110_h - CAN Overrun Error
 - Sample-Rate ist so hoch eingestellt, dass die Firmware nicht in der Lage ist, alle Daten auf dem CAN-Bus zu senden
- 8120_h - CAN in Error Passive Mode
- 8130_h - Lifeguard Error / Heartbeat Error
- 8140_h - Recovered from "Bus Off"
- 8240_h - Unexpected SYNC data length
- 6000_h - Software-Fehler:
 - EEPROM Checksum Fehler (keine Übertragung dieser Message als Emergency)
- 6110_h - Internal Software Error
 - z.B.:
 - gespeicherte Daten hatten ungültige Checksumme und Default-Werte wurden geladen
 - interner Watchdog hat ausgelöst
- FF10_h - Data lost (A/D-Daten Overflow)
- 5000_h - Hardware Fehler (z.B. A/D-Wandler defekt)
- 3110_h - VIO zu hoch
 - (über Objekt 2300_h konfigurierbar)
- 3120_h - VIO zu niedrig
 - (über Objekt 2300_h konfigurierbar)
- 2300_h - Fehler am Ausgang: Open Load
 - (keine Last an mindestens einem Ausgang erkannt)
- 2320_h - Fehler am Ausgang: Short circuit at outputs
 - (Kurzschluss mindestens eines Ausganges gegen GND)

Emergency Message

Die Daten des von der CAN-CBX-DIO8 gesendeten Emergency-Frames sind wie folgt aufgebaut:

Byte:	0	1	2	3	4	5	6	7
Inhalt:	<i>emergency-error-code</i> (siehe oben)		<i>error-register</i> 1001 _h	<i>no_of_errors_in_list</i> 1003,00 _h	<i>manufacturer-specific error field</i>		0	0

Eine Emergency Message wird beim ersten Auftreten eines Fehlers der digitalen Ausgangstreiber gesendet, wenn das entsprechende Bit in Objekt 2220_h auf '1' gesetzt ist. Tritt bei einem weiteren Ausgang eine Fehlermeldung auf, wird keine weitere Emergency Message gesendet. Sobald die letzte Fehlermeldung zurückgenommen wird, wird wiederum eine Emergency Message gesendet.



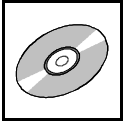
8.9.5 COB-ID of SYNC-Message (1005_h)

INDEX	1005_h
Name	<i>COB-ID SYNC message</i>
Datentyp	unsigned 32
Zugriff	rw
Default-Wert	80 _h

Struktur des Parameters:

Bit-No.	Wert	Bedeutung
31 (MSB)	-	do not care
30	0/1	0: keine SYNC Erzeugung 1: Modul erzeugt SYNC
29	0	immer 0, da 11-Bit-ID
28...11	0	immer 0, da keine 29-Bit-ID unterstützt werden
10...0 (LSB)	x	Bit 0...10 der SYNC-COB-ID

Der Identifier kann Werte zwischen 0...7FF_h annehmen.



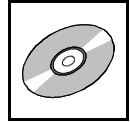
Implementierte CANopen Objekte

8.9.6 Communication Cycle Period (1006_h)

INDEX	1006_h
Name	<i>Communication Cycle Period</i>
Datentyp	unsigned 32
Zugriffsart	rw
Default-Wert	0 μ s

Wertebereich des Parameters:

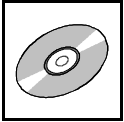
Wert	Bedeutung
0	Übertragung von SYNC-Nachrichten ausgesetzt
1...FFFFFFF _h	Zyklus-Zeit in Mikrosekunden



8.9.7 Manufacturer's Device Name (1008_h)

INDEX	1008_h
Name	<i>manufacturer's device name</i>
Datentyp	visible string
Default-Wert	siehe Kapitel 8.9.1 ab Seite 55

Eine ausführliche Beschreibung des SDO Uploads ist [1] zu entnehmen.



Implementierte CANopen Objekte

8.9.8 Manufacturer's Hardware Version (1009_h)

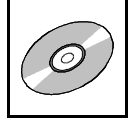
INDEX	1009_h
Name	<i>manufacturer's hardware version</i>
Datentyp	visible string
Default-Wert	string: z.B. '1.0' (Versionsabhängig)

Das Lesen der Hardware-Version erfolgt ähnlich wie das Lesen des Manufacturer Device Names über das SDO Upload Protokoll. Eine ausführliche Beschreibung des Uploads ist [1] zu entnehmen.

8.9.9 Manufacturer's Software Version 100A_h

INDEX	100A_h
Name	<i>manufacturer's software version</i>
Datentyp	visible string
Default-Wert	string: z.B.: '1.2' ' (Versionsabhängig)

Das Lesen der Hardware-Version erfolgt ähnlich wie das Lesen des Manufacturer Device Names über das SDO Upload Protokoll. Eine ausführliche Beschreibung des Uploads ist [1] zu entnehmen.



8.9.10 Guard Time (100C_h) und Life Time Factor (100D_h)

Das CAN-CBX-Modul unterstützt das Node-Guarding oder alternativ die Heartbeat-Funktion (siehe Seite 75).



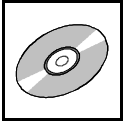
HINWEIS

Nach den CiA-Empfehlungen sollte bevorzugt die Heartbeat-Funktion verwendet werden. Das Node-Guarding sollte nur noch für bestehende Systeme eingesetzt werden!

Guard Time und Life Time Factor werden zusammen ausgewertet. Die Multiplikation beider Werte ergibt die Life Time. Die Guard Time wird in Millisekunden angegeben.

INDEX	100C _h
Name	<i>guard time</i>
Datentyp	unsigned 16
Zugriffsart	rw
Default-Wert	0 [ms]
Minimum Value	0
Maximum Value	FFFF _h (65,535 s)

INDEX	100D _h
Name	<i>life time factor</i>
Datentyp	unsigned 8
Zugriffsart	rw
Default-Wert	0
Minimum Value	0
Maximum Value	FF _h



Implementierte CANopen Objekte

8.9.11 Node Guarding Identifier (100E_h)

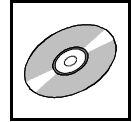
Das Modul unterstützt nur 11-Bit-Identifizier.

INDEX	100E_h
Name	<i>node guarding identifier</i>
Datentyp	unsigned 32
Zugriffsart	rw
Default-Wert	700 _h + Node-ID

Struktur des Parameters *node guarding identifier* :

Bit-No.	Bedeutung
31 (MSB) 30	reserved
29...11	immer 0, da keine 29-Bit-ID unterstützt werden
10...0 (LSB)	Bit 0...10 des Node Guarding Identifiers

Der Identifier kann Werte zwischen 1...7FF_h annehmen.



8.9.12 Store Parameters (1010_h)

INDEX	1010_h
Name	<i>store parameters</i>
Datentyp	unsigned 32

Mit diesem Kommando werden die Parameter im “nicht-flüchtigen” Speicher, hier im EEPROM, gespeichert. Dazu werden die unten aufgeführten Parameter Gruppen unterschieden. Die eingestellten Parameter sind sofort nach der Übergabe aktiv. Das ‘nichtflüchtige’ Speichern der Parameter erfolgt jedoch nicht automatisch. Es muss über einen Schreibzugriff auf das Objekt 1010_h veranlasst werden und sollte nur erfolgen, wenn sich das Modul im Zustand *pre-operational* befindet. Beim Schreiben des Index muss die unten angegebene Byte-Folge gesendet werden.

Das Lesen des Index gibt Informationen über die implementierten Speicher-Funktionen zurück (siehe [1]).

Index	Sub-Index	Beschreibung	Wertebereich	Datentyp	Zugriff
1010_h	0	<i>number_of_entries</i>	4	unsigned 8	ro
	1	<i>save_all_parameters</i> (Objekte 1000 _h ... 9FFF _h)	no default, write: 65 76 61 73 _h (= ASCII: ‘e’ ‘v’ ‘a’ ‘s’)	unsigned 32	rw
	2	<i>save_communication_parameter</i> (Objekte 1000 _h ... 1FFF _h)		unsigned 32	rw
	3	<i>save_application_parameter</i> (Objekte 6000 _h ... 9FFF _h)		unsigned 32	rw
	4	<i>save_manufacturer_parameter</i> (Objekte 2000 _h ... 5FFF _h)		unsigned 32	rw

Belegung der Variablen

save all parameters

speichert die Parameter aller Objekte (sofern vorhanden), deren Zugriffsrecht Read/Write (rw) ist.

save_communication_parameter

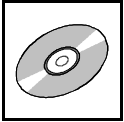
speichert alle Communication Parameter der Objekte (1000_h ... 1FFF_h, sofern vorhanden), deren Zugriffsrecht Read/Write (rw) ist (hier z.B. 1005_h ... 1029_h).

save_application_parameter

speichert alle Application Parameter der Objekte (Objekte 6000_h... 9FFF_h, sofern vorhanden), deren Zugriffsrecht Read/Write (rw) ist (hier z.B. 6xxx_h).

save_manufacturer_parameter

speichert alle Manufacturer Parameter der Objekte (Objekte 2000_h ... 5FFF_h, sofern vorhanden), deren Zugriffsrecht Read/Write (rw) ist (hier z.B. 2xxx_h).



Implementierte CANopen Objekte

Der Speicher-Modus wird über den Inhalt dieses Objektes angezeigt:

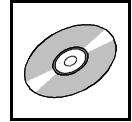
Bit 1 des Objekts 1010_h, Sub-Index 1 ist nicht gesetzt, d.h. das CAN-CBX-Modul speichert die Konfiguration nicht 'automatisch', sondern muss durch Schreiben der Zeichenkette 'save' (73_h 61_h 76_h 65_h, Reihenfolge aus CAN-Telegramm) in das Objekt 1010_h, Sub-Index 1-4 extra dazu veranlasst werden.

Beim Lesen eines Sub-Index liefert das CANopen Device Informationen über die Speicher-Funktionalität im folgenden Format:

Bit:	31	2	1	0	
Inhalt:	reserved			auto	cmd
	0			0	1
	MSB			LSB	

Bit	Wert	Beschreibung
auto	0	das CAN-CBX-Modul speichert die Parameter nicht autonom
	1	das CAN-CBX-Modul speichert die Parameter autonom
cmd	0	das CAN-CBX-Modul speichert die Parameter nicht auf Befehl
	1	das CAN-CBX-Modul speichert die Parameter auf Befehl

Autonomes Speichern bedeutet, dass die Parameter "nicht-flüchtig" ohne weitere Abfrage gespeichert werden.



8.9.13 Restore Default Parameters (1011_h)

INDEX	1011_h
Name	<i>restore default parameters</i>
Datentyp	unsigned 32

Mit diesem Kommando werden die bei der Auslieferung aktiven Default-Parameter wieder geladen. Dazu werden die unten aufgeführten Parameter Gruppen unterschieden.

Alle individuellen Einstellungen, die im EEPROM gespeichert worden sind, gehen verloren, das betrifft auch Kalibrierungswerte. Die Default-Parameter sind erst nach einem Reset aktiv. Das Aktivieren der Parameter muss über einen Schreibzugriff veranlasst werden.

Beim Schreiben des Index muss die unten angegebene Byte-Folge gesendet werden. Das Lesen des Index gibt Informationen über die implementierten Restore-Funktionen zurück (siehe [1])

Index	Sub-Index	Beschreibung	Wertebereich	Datentyp	Zugriff
1011_h	0	<i>number_of_entries</i>	4	unsigned 8	ro
	1	<i>restore_all_default_parameters</i> (Objekte 1000 _h ... 9FFF _h)	no default, write: 64 61 6F 6C _h (= ASCII: 'd' 'a' 'o' '1')	unsigned 32	rw
	2	<i>restore_communication_default_parameter</i> (Objekte 1000 _h ... 1FFF _h)		unsigned 32	rw
	3	<i>restore_application_default_parameter</i> (Objekte 6000 _h ... 9FFF _h)		unsigned 32	rw
	4	<i>restore_manufacturer_default_parameter</i> (Objekte 2000 _h ... 5FFF _h)		unsigned 32	rw

Belegung der Variablen

restore_all_default_parameters

lädt die Default-Werte aller Parameter aller Objekte (sofern vorhanden).

restore_communication_default_parameter

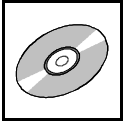
lädt die Default-Werte aller Communication Parameter der Objekte (1000_h ... 1FFF_h, sofern vorhanden, z.B. 1005_h ... 1029_h).

restore_application_default_parameter

lädt die Default-Werte der Application Parameter der Objekte (Objekte 6000_h ... 9FFF_h, sofern vorhanden, z.B. 6xxx_h).

restore_manufacturer_default_parameter

lädt die Default-Werte aller Manufacturer Parameter der Objekte (Objekte 2000_h ... 5FFF_h, sofern vorhanden, hier z.B. 2xxx_h).



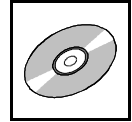
Implementierte CANopen Objekte

Bit 0 des Objekts 1011_h, Sub-Index 1 ist gesetzt, d.h. das CAN-CBX-Modul lädt die Standardwerte durch Schreiben der Zeichenkette 'load' (64_h 61_h 6F_h 6C_h, Reihenfolge aus CAN-Telegramm) in das Objekt 1011_h, Sub-Index 1-4.

Beim Lesen eines Sub-Index liefert das CANopen Device Informationen über die Restore-Funktionalität im folgenden Format:

Bit:	31	1	0
Inhalt:	reserved		cmd
	0		1
	MSB		LSB

Bit	Wert	Beschreibung
cmd	0	die Default-Parameter des CAN-CBX-Moduls werden nicht geladen
	1	die Default-Parameter des CAN-CBX-Moduls werden geladen



8.9.14 COB_ID Emergency Message (1014_h)

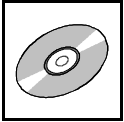
INDEX	1014_h
Name	<i>COB-ID emergency object</i>
Datentyp	unsigned 32
Default-Wert	80 _h + Node-ID

Dieses Objekt bestimmt die COB-ID der Emergency Message (EMCY).

Die Struktur des Objektes ist in der folgenden Tabelle beschrieben:

Bit-No.	Wert	Bedeutung
31 (MSB)	0/1	0: EMCY existiert / ist gültig 1: kein EMCY / EMCY ist nicht gültig
30	0	reserviert (immer 0)
29	0	immer 0, da 11-Bit ID
28...11	0	immer 0, da keine 29-Bit-ID unterstützt werden
10...0 (LSB)	x	Bit 0...10 des COB-ID

Der Identifier kann Werte zwischen 0...7FF_h annehmen.

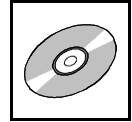


Implementierte CANopen Objekte

8.9.15 Inhibit Time EMCY (1015_h)

INDEX	1015_h
Name	<i>inhibit_time_emergency</i>
Datentyp	unsigned 16
Zugriffsart	rw
Wertebereich	0...FFFF _h
Default-Wert	0

Mit diesem Objekt kann die *Inhibit Time* für die Emergency-Nachricht festgelegt werden. Die Zeit wird als ein Vielfaches von 100 µs angegeben.



8.9.16 Consumer Heartbeat Time (1016_h)

INDEX	1016_h
Name	<i>consumer heartbeat time</i>
Datentyp	unsigned 32
Default-Wert	No

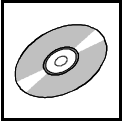
Zur gegenseitigen Überwachung der CANopen-Module (insbesondere zum Erkennen von Verbindungsausfällen) kann die Heartbeat-Funktion genutzt werden. Im Gegensatz zum Node Guarding/Life Guarding kommt die Heartbeat-Funktion ohne RTR-Frames aus.

Funktionsweise:

Ein Modul, der sog. Heartbeat-Producer sendet auf dem Node-Guarding-Identifizier (siehe Object 100E_h) zyklisch eine Heartbeat-Nachricht auf dem CAN-Bus. Ein oder mehrere Heartbeat-Consumer empfangen die Nachricht. Die Nachricht muss innerhalb der auf dem Heartbeat-Consumer gespeicherten Heartbeat-Time empfangen werden, sonst wird auf dem Heartbeat-Consumer-Modul ein Heartbeat-Event ausgelöst. Auf dem CAN-CBX-Modul löst der Heartbeat-Event einen Heartbeat-Error aus.

Jedes Modul kann Heartbeat-Producer und Heartbeat-Consumer sein. Das CAN-CBX-Modul unterstützt in einem CAN-Netz maximal einen Heartbeat-Producer.

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1016_h	0	<i>number_of_entries</i>	1	1	unsigned 8	ro
	1	<i>consumer-heartbeat_time</i>	0...007F FFFF _h	0	unsigned 32	rw



Implementierte CANopen Objekte

Bedeutung der Variablen *consumer-heartbeat_time_x*:

<i>consumer-heartbeat_time_x</i>			
Bit	3124	2316	150
Belegung	reserved (immer '0')	<i>Node-ID</i> (unsigned 8)	<i>heartbeat_time</i> (unsigned 16)

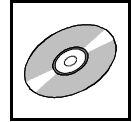
Node-ID Node-Id des zu überwachenden Heartbeat-Producers.

heartbeat_time Zeit in [ms], innerhalb der sich der zu überwachende Heartbeat-Producer auf dem Node-Guarding-ID melden muss, damit kein Heartbeat-Event ausgelöst wird. Diese Consumer-Heartbeat-Time muss immer größer sein als die Producer-Heartbeat-Time.

Beispiel:

consumer-heartbeat_time = 0031 03E8_h

=> *Node-ID* = 31_h = 49_d
=> *heartbeat time* = 3E8_h = 1000_d => 1 s



8.9.17 Producer Heartbeat Time (1017_h)

INDEX	1017_h
Name	<i>producer heartbeat time</i>
Datentyp	unsigned 16
Default-Wert	0 ms

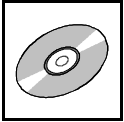
Hier wird die Zeit eingetragen, mit der das CAN-CBX-Modul zyklisch einen Heartbeat-Frame auf dem Node-Guarding-ID sendet.

Wird für die Producer-Heartbeat-Time ein Wert größer Null eingesetzt, so ist sie aktiv und unterbindet das Node-/ Life-Guarding (siehe Seite 65).

Wird die Producer-Heartbeat-Time auf '0' gesetzt, so wird das Senden des Heartbeats durch dieses Modul beendet.

Index [Hex]	Sub-index	Beschreibung	Wertebereich [Hex]	Default	Datentyp	Zugriff
1017	0	<i>producer-heartbeat_time</i>	0...FFFF	0 ms	unsigned 16	rw

producer-heartbeat_time Zykluszeit in [ms] des Heartbeat-Producers zum Senden des Heartbeats auf dem Node-Guarding-ID (siehe Object 100E_h).
 Die Consumer-Heartbeat-Time der überwachenden Module muss immer größer sein als die Producer-Heartbeat-Time dieses Heartbeat-sendenden Moduls.



Implementierte CANopen Objekte

8.9.18 Identity Object (1018_h)

INDEX	1018_h
Name	<i>identity object</i>
Datentyp	unsigned 32
Default-Wert	No

Dieses Objekt enthält allgemeine Informationen zum CAN-Modul.

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1018_h	0	<i>no_of_entries</i>	4	4	unsigned 8	ro
	1	<i>vendor_id</i>	0...FFFF FFFF _h	0000 0017 _h	unsigned 32	ro
	2	<i>product_code</i>	0...FFFF FFFF _h	2301 0002 _h	unsigned 32	ro
	3	<i>revision_number</i>	0...FFFF FFFF _h	0	unsigned 32	ro
	4	<i>serial_number</i>	0...FFFF FFFF _h	-	unsigned 32	ro

Bedeutung der Variablen:

vendor_id Diese Variable enthält die esd-Vendor-ID. Hier ist immer der Wert 0000 0017_h eingetragen.

product_code Hier ist die esd-Artikelnummer des Produkts abgelegt.
Die Nibble des Langwortes haben folgende Bedeutung:

$$product_code = abcd\ efgh_h$$

a: 1... Artikelnummer beginnt mit Buchstaben "K"

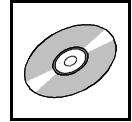
2....Artikelnummer beginnt mit Buchstaben "C"

bcde: 4-stellige Hex-Zahl, die als Dezimalzahl vor dem Dezimalpunkt interpretiert wird.

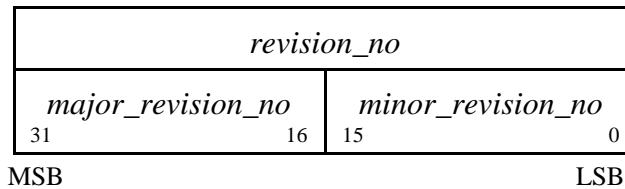
f: wird zur Zeit nicht ausgewertet

gh: 2-stellige Hex-Zahl, die als Dezimalzahl nach dem Dezimalpunkt interpretiert wird.

Beispiel: '2301 0002_h' entspricht der Artikelnummer 'C.3010.02'.



revision_number Hier ist die Software-Version abgelegt. In den oberen zwei Bytes sind gemäß [1] die Revisionsnummern der wesentlichen (major) Änderungen aufgeführt und in den unteren zwei Bytes die Revisionsnummern einfacher Korrekturen oder Änderungen (minor).



serial_number Hier wird die Seriennummer-Kennung der Hardware gelesen. Die ersten beiden Zeichen der Seriennummer sind Buchstaben, die das Fertigungslos kennzeichnen. Die folgenden Zeichen geben die eigentliche Seriennummer wieder.

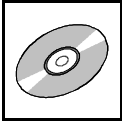
In den beiden höherwertigen Bytes von *serial_no* sind die Buchstaben des Fertigungsloses kodiert. Sie enthalten jeweils den ASCII-Code des Buchstaben mit dem hochwertigsten Bit auf '1' gesetzt, um Buchstaben und Zahlen unterscheiden zu können:

$$(\text{ASCII-Code}) + 80_{\text{h}} = \text{gelesenes_Byte}$$

In den beiden niederwertigen Bytes ist die Nummer des Moduls als BCD-Wert enthalten.

Beispiel:

Wird der Wert 'C1C2 0105_h' gelesen, so entspricht dies der Hardware-Seriennummer-Kennung 'AB 0105'. Dieser Wert muss der aufgeklebten Seriennummer des Moduls entsprechen.



Implementierte CANopen Objekte

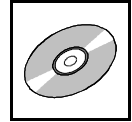
8.9.19 Synchronous Counter Overflow Value (1019_h)

INDEX	1019_h
Name	<i>Synchron_Counter_Overflow</i>
Datentyp	unsigned 8
Default-Wert	0

Dieses Objekt bestimmt den höchsten Wert, den der SYNC-Counter erreichen kann.

Der Wertebereich des Objektes ist in der folgenden Tabelle beschrieben:

Wert	Bedeutung
0	Die SYNC-Nachricht wird als CAN-Nachricht mit der Länge '0' gesendet.
1	reserviert
2...240	Die SYNC-Nachricht wird als CAN-Nachricht mit der Länge '1' gesendet. Das erste Byte der SYNC-Nachricht enthält den aktuellen Wert des SYNC-Counters.
241...255	reserviert



8.9.20 Verify Configuration (1020_h)

INDEX	1020_h
Name	<i>verify configuration</i>
Datentyp	unsigned 32
Default-Wert	No

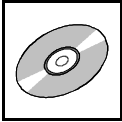
In diesem Objekt können Datum und Tageszeit der letzten Konfiguration abgelegt werden, um zu einem späteren Zeitpunkt prüfen zu können, ob die gespeicherte Konfiguration der erwarteten entspricht.

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1020_h	0	<i>no_of_entries</i>	2	2	unsigned 8	ro
	1	<i>configuration_date</i>	0...FFFF FFFF _h	0	unsigned 32	rw
	2	<i>configuration_time</i>	0...FFFF FFFF _h	0	unsigned 32	rw

Bedeutung der Variablen:

configuration_date Datum der letzten Konfiguration des Moduls, angegeben in Tagen seit dem 01.01.1984.

configuration_time Zeit in ms seit Mitternacht am Tag der letzten Konfiguration.



Implementierte CANopen Objekte

8.9.21 Error Behaviour Object (1029_h)

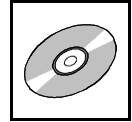
INDEX	1029_h
Name	<i>error behaviour object</i>
Datentyp	unsigned 8
Default-Wert	No

Tritt ein Error-Event ein (z.B. Heartbeat-Error), so wechselt das Modul in den Zustand, der in der Variablen *communication_error* oder *output_error* definiert ist.

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1029_h	0	<i>no_of_error_classes</i>	6	6	unsigned 8	ro
	1	<i>communication_error</i>	0...2	0	unsigned 8	rw
	2	<i>output_error</i>	0...2	0	unsigned 8	rw
	3	<i>input_error</i>	0...2	0	unsigned 8	ro
	4	<i>VIO_extreme_low</i>	0...2	1	unsigned 8	rw
	5	<i>VIO_low_warning</i>	0...2	1	unsigned 8	rw
	6	<i>VIO_high_warning</i>	0...2	1	unsigned 8	rw

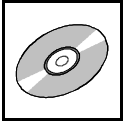
Bedeutung der Variablen:

Variable	Bedeutung
<i>no_of_error_classes</i>	Anzahl der Fehler-Klassen (hier immer '6')
<i>communication_error</i>	Heartbeat/Lifeguard-Fehler und <i>Bus off</i>
<i>output_error</i>	Fehler am Ausgang
<i>input_error</i>	Fehler am Eingang (wird z. Zt. nicht unterstützt)
<i>VIO_extreme_low</i>	Fehler der Spannungsversorgung der Ausgänge: VIO < 5 V
<i>VIO_low_warning</i>	Fehler der Spannungsversorgung der Ausgänge: VIO ist kleiner als der Wert, der in Objekt 2300 _h , Subindex 2 definiert ist
<i>VIO_high_warning</i>	Fehler der Spannungsversorgung der Ausgänge: VIO ist größer als der Wert, der in Objekt 2300 _h , Subindex 3 definiert ist



Das Modul wechselt beim Auftreten eines Fehlers in den jeweils angegebenen Modus.

Wert der Variablen	Modus, in den das Modul im Fehlerfall wechselt
0	pre-operational (nur wenn der aktuelle Zustand = operational)
1	no state change
2	stopped



Implementierte CANopen Objekte

8.9.22 NMT Startup (1F80_h)

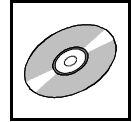
INDEX	1F80_h
Name	<i>NMT startup</i>
Datentyp	unsigned 32
Default-Wert	0002 _h

Um in Umgebungen, in denen kein NMT-Master verfügbar ist, CANopen-Nodes starten zu können, ist der NMT Startup implementiert worden.

Über den NMT-Startup kann der Autostart eines CANopen-Nodes enabled oder disabled werden. Weitere Funktionen des Parameters *NMT startup* werden zur Zeit nicht unterstützt.

Der Wertebereich des Objektes ist in der folgenden Tabelle beschrieben:

Wert	Bedeutung
0002 _h	Autostart disabled (default)
0008 _h	Autostart enabled
alle anderen Werte	reserviert



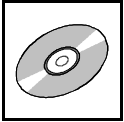
8.9.23 Self Starting Nodes Timing Parameters (1F91_h)

INDEX	1F91_h
Name	<i>Self starting nodes timing parameters</i>
Datentyp	unsigned 16

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1F91_h	0	<i>number_of_entries</i>	3	1	unsigned 8	ro
	1	<i>NMT master detection timeout</i>	0000 _h ...FFFF _h	64 _h	unsigned 16	rw

Subindex 1 dieses Objektes enthält die Verzögerungszeit in [ms] zwischen dem Übergang von “preoperational” > “operational”. In der Default-Einstellung sind dies 100 ms.

Die Sub-Indizes 2 und 3 dieses Objektes werden nicht unterstützt.



Implementierte CANopen Objekte

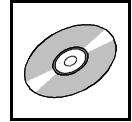
8.9.24 Receive PDO Communication Parameter 1400_h

Mit dem Objekt 'Receive PDO Communication Parameter 1400_h' werden die Eigenschaften eines Empfangs-PDOs (Rx-PDO) definiert.

INDEX	1400_h
Name	<i>receive PDO parameter</i>
Datentyp	PDOCommPar

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1400_h	0	<i>no_of_entries</i>	2	2	unsigned 8	ro
	1	<i>COB_ID used by PDO1</i>	1... 800007FF _h	200 _h + Node-ID	unsigned 32	rw
	2	<i>transmission type</i>	0...FF _h	255 _d	unsigned 8	rw

Es werden alle *transmission types* unterstützt.



8.9.25 Receive PDO Mapping Parameter 1600_h

Mit dem Objekt 'Receive PDO Mapping Parameter 1600_h' wird die Zuordnung der Empfangsdaten zu den Rx-PDOs definiert.

INDEX	1600_h
Name	<i>receive PDO mapping</i>
Datentyp	PDO Mapping

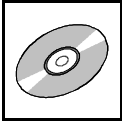
Die folgende Tabelle zeigt die Belegung der Receive PDO Mapping Parameter für die Default-Konfiguration:

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1600	0	<i>no_of_mapped_application_objects_in_PDO</i>	1...8	1	unsigned 8	rw
	1	<i>1st_application_object</i>	00050008 _h , 6200 0108 _h	6200 0108 _h	unsigned 32	rw
	2	<i>2nd_application_object</i>		0005 0008 _h	unsigned 32	rw
	3	<i>3rd_application_object</i>		0005 0008 _h	unsigned 32	rw
	4	<i>4th_application_object</i>		0005 0008 _h	unsigned 32	rw
	5	<i>5th_application_object</i>		0005 0008 _h	unsigned 32	rw
	6	<i>6th_application_object</i>		0005 0008 _h	unsigned 32	rw
	7	<i>7th_application_object</i>		0005 0008 _h	unsigned 32	rw
	8	<i>8th_application_object</i>		0005 0008 _h	unsigned 32	rw

Parallelschaltung von bis zu 8 CAN-CBX-DIO8-Modulen:

Generell besteht die Möglichkeit mehrere CAN-CBX-DIO8-Module gleichzeitig anzusprechen. Die CAN-CBX-DIO8-Module können so konfiguriert werden, dass mit einem einzigen CAN-Frame bis zu 8 CAN-CBX-DIO8-Module gleichzeitig angesprochen werden, so dass in diesem Falle sämtliche 64 digitalen Ausgänge gleichzeitig gesetzt werden können. Die Receive COB-IDs der gewünschten Module müssen über Objekt 1400_h auf den selben Wert gesetzt werden. Siehe auch CiA 301 [1], Objekt 1600_h.

Subindex 0_h enthält die Anzahl der gültigen Einträge im Mapping-Datensatz (siehe folgende Tabelle). Die Anzahl der gültigen Einträge sollte bei allen Modulen gleich sein.



Implementierte CANopen Objekte

Wert	Beschreibung
00 _h	Mapping disabled
01 _h	Subindex 1 gültig
02 _h	Subindex 1 und 2 gültig
03 _h	Subindex 1 - 3 gültig
:	:
08 _h	Subindex 1 - 8 gültig

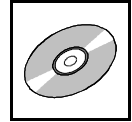
Subindex 01_h bis 08_h enthalten die Objekte, die angesprochen werden sollen mit Index, Subindex und Länge. Für das CAN-CBX-DIO8-Modul darf der Wert 6200 0108_h (Index:6200_h, Subindex: 01_h, und Länge: 08_h) nur einmal eingetragen sein. In den übrigen Subindices ist als so genanntes Dummy Mapping der Wert 0005 0008_h eingetragen.

Beispiel :

Es sollen drei CAN-CBX-DIO8-Module gleichzeitig über RPDO-Mapping angesprochen werden. Dazu müssen die RPDO COB-IDs der Module auf den gleichen Wert konfiguriert werden. Siehe dazu auch CiA 301 [1].

Das Objekt 1600_h muss nun bei den drei Modulen unterschiedlich konfiguriert werden, so dass für das erste Modul das *application_object* unter Subindex 1_h eingetragen wird, für das zweite unter Subindex 2_h und für das dritte unter Subindex 3_h (siehe Abb.24). Bei einer größeren Anzahl von Modulen (hier bis zu 8 Modulen) müssen die Einträge entsprechend weitergeführt werden.

In die voranstehenden, bzw. nachstehenden jeweils nicht benötigten Subindices (hier Subindex 1_h und 2_h) muss als so genanntes Dummy-Mapping der Wert 0005 0008_h eingetragen sein. Subindex 0_h enthält die Anzahl der Subindices, entsprechend der Anzahl der gültigen Objekte.



Eintrag unter Objekt 1600_h für Modul 3:

Index	Subindex	Beschreibung	Wert
1600_h	0	<i>no_of_mapped_application_objects_in_PDO</i>	03
	1	<i>1st_application_object</i>	0005 0008 _h
	2	<i>2nd_application_object</i>	0005 0008 _h
	3	<i>3rd_application_object</i>	6200 0108_h

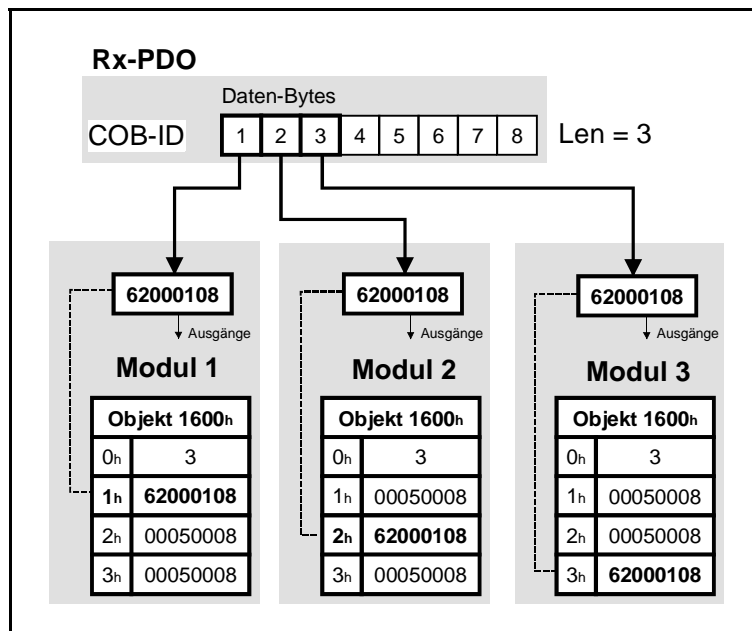
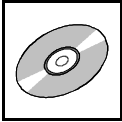


Abb. 21: Beispiel für das Rx-PDO-Mapping mit drei Modulen



Implementierte CANopen Objekte

8.9.26 Objekt Transmit PDO1 Communication Parameter 1800_h

Mit diesem Objekt werden die Eigenschaften eines Sende-PDO1 definiert.

INDEX	1800_h
Name	<i>transmit PDO1 parameter</i>
Datentyp	PDOCommPar

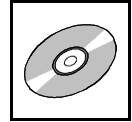
Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1800_h	0	<i>number_of_entries</i>	0..FF _h	5	unsigned 8	ro
	1	<i>COB-ID_used_by_PDO1</i>	1...800007FF _h	0000 0000 _h +180 _h + Node-ID	unsigned 32	rw
	2	<i>transmission type</i>	0..FF _h	255 _d	unsigned 8	rw
	3	<i>inhibit time</i>	0...FFFF _h	0	unsigned 16	rw
	4	<i>reserved</i>	0..FF _h	0	unsigned 8	const
	5	<i>event timer</i>	0...FFFF _h	0	unsigned 16	rw

Es werden die *transmission types* 0, 1-240, 252, 253 und 255 unterstützt.

Defaultmäßig ist für den Parameter *COB-ID used by_PDO1* das oberste Bit **nicht gesetzt**:

0000 0000_h +180_h +Node-ID.

Damit ist das Objekt per default **gültig**.



8.9.27 Objekt Transmit PDO2 Communication Parameter 1801_h

Mit diesem Objekt werden die Eigenschaften eines Sende-PDO2 definiert.

INDEX	1801_h
Name	<i>transmit PDO2 parameter</i>
Datentyp	PDOCommPar

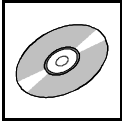
Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1801_h	0	<i>number_of_entries</i>	0...FF _h	5	unsigned 8	ro
	1	<i>COB-ID used by PDO2</i>	1...800007FF _h	8000 0000_h +280 _h +Node-ID	unsigned 32	rw
	2	<i>transmission type</i>	0...FF _h	255 _d	unsigned 8	rw
	3	<i>inhibit time</i>	0...FFFF _h	0	unsigned 16	rw
	4	<i>reserved</i>	0..FF _h	0	unsigned 8	const
	5	<i>event timer</i>	0...FFFF _h	0	unsigned 16	rw

Es werden die *transmission types* 0, 1-240, 252, 253 und 255 unterstützt.

Defaultmäßig ist für den Parameter *COB-ID used by_PDO2* das oberste Bit **gesetzt**:

8000 0000_h +280_h +Node-ID.

Damit ist das Objekt per default **ungültig**.



Implementierte CANopen Objekte

8.9.28 Transmit PDO1 Mapping Parameter 1A00_h

Mit dem Objekt 'Transmit PDO1 Mapping Parameter 1A00_h' wird die Zuordnung der Sendedaten zu den Tx-PDO1s definiert.

INDEX	1A00_h
Name	<i>transmit PDO1 mapping</i>
Datentyp	PDO Mapping

Die folgende Tabelle zeigt die Belegung der Transmit PDO1 Mapping Parameter:

Index	Subindex	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1A00_h	0	<i>number of entries</i>	0...FF _h	1 _h	unsigned 8	ro
	1	<i>object_to_be_mapped</i>	0...FFFFFFFF _h	60000108 _h	unsigned 32	rw

Das Tx-PDO-Mapping ist frei konfigurierbar.

In der Default-Einstellung werden die Digitalen Eingänge DI1-DI8 (Objekt 6000_h, Subindex 1) gemapped.

Objekte, die gemapped werden können (mit Index, Sub-Index und Länge in Bit):

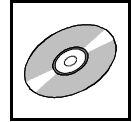
6000 01 08 _h	Digitale Eingänge, <i>Read_Input_DI8-DI1</i> (Objekt 6000 _h , Subindex 1)
2402 01 10 _h ... 2402 08 10 _h	<i>Counter_Value_1 - Counter_Value_8</i> , 16-Bit (Obj. 2402 _h , Subindex 1-8)
2403 01 20 _h ... 2403 08 20 _h	<i>Counter_Value_1 - Counter_Value_8</i> , 32-Bit (Obj. 2403 _h , Subindex 1-8)
0005 00 08 _h	"Dummy Mapping"



HINWEIS

Das PDO1-Mapping ist bei Auslieferung gültig.

Die Transmit PDO1-Mapping-Parameter können dann bei gültigem PDO1 nicht geändert werden. Um den Wert des Parameters *object_to_be_mapped* ändern zu können muss daher zunächst in Objekt 1800_h (Seite 88) der Wert des Parameters *COB-ID_used_by_PDO1* auf *not valid* gesetzt werden.



8.9.29 Transmit PDO2 Mapping Parameter 1A01_h

Mit dem Objekt ‘Transmit PDO2 Mapping Parameter 1A01_h’ wird die Zuordnung der Sendedaten zu den Tx-PDO2s definiert.

INDEX	1A01_h
Name	<i>transmit PDO2 mapping</i>
Datentyp	PDO Mapping

Die folgende Tabelle zeigt die Belegung der Transmit PDO2 Mapping Parameter:


Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
1A01_h	0	<i>number of entries</i>	0...FF _h	4 _h	unsigned 8	rw
	1	<i>object_to_be_mapped_1</i>	0...FFFFFFFF _h	24020110 _h	unsigned 32	rw
	2	<i>object_to_be_mapped_2</i>	0...FFFFFFFF _h	24020210 _h	unsigned 32	rw
	3	<i>object_to_be_mapped_3</i>	0...FFFFFFFF _h	24020310 _h	unsigned 32	rw
	4	<i>object_to_be_mapped_4</i>	0...FFFFFFFF _h	24020410 _h	unsigned 32	rw

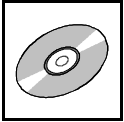
Das Tx-PDO2-Mapping ist frei konfigurierbar.

Die Anzahl der Objekte, die gemapped werden können, ist abhängig von deren Größe. Es können maximal 8 Byte übertragen werden. Das bedeutet, es können maximal vier 16-Bit Werte (siehe Objekt 2402_h) oder zwei 32-Bit-Werte (siehe Objekt 2403_h) in dem PDO übertragen werden. Kombinationen sind jedoch möglich.

In der Default-Einstellung werden die Counter 1-4 (Objekt 2402_h, Subindex 1-4) gemapped. Objekte, die gemapped werden können (mit Index, Sub-Index und Länge in Bit):

6000 01 08_h Digitale Eingänge, *Read_Input_DI8-DI1* (Objekt 6000_h, Subindex 1)
 2402 01 10_h ... 2402 08 10_h *Counter_Value_1 - Counter_Value_8*, 16-Bit (Obj. 2402_h, Subindex 1-8)
 2403 01 20_h ... 2403 08 20_h *Counter_Value_1 - Counter_Value_8*, 32-Bit (Obj. 2403_h, Subindex 1-8)
 0005 00 08_h “Dummy Mapping”

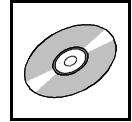
	<p>HINWEIS Das PDO2-Mapping ist bei Auslieferung ungültig. Die Transmit PDO2-Mapping Parameter können bei ungültigem PDO2 direkt geändert werden. Um das PDO2-Mapping zu aktivieren muss in Objekt 1801_h (siehe Seite 89) der Wert des Parameters <i>COB-ID_used_by_PDO2</i> auf <i>valid</i> gesetzt werden.</p>
---	---



Implementierte CANopen Objekte

Die folgende Tabelle zeigt beispielhaft eine beliebige Belegung der Transmit PDO2 Mapping Parameter:

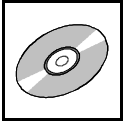
Index	Sub-index	Beschreibung	Wert	Datentyp
1A01_h	0	<i>number of entries in PDO2</i>	4 _h	unsigned 8
	1	<i>object_to_be_mapped_1</i>	00050008 _h	unsigned 8
	2	<i>object_to_be_mapped_2</i>	60000108 _h	unsigned 8
	3	<i>object_to_be_mapped_3</i>	24030520 _h	unsigned 32
	4	<i>object_to_be_mapped_4</i>	24020110 _h	unsigned 16



8.10 Device Profile Area

8.10.1 Übersicht der implementierten Objekte 6000_h ... 6207_h

Index	Name	Datentyp
6000 _h	<i>Read Input 8-bit</i>	unsigned 8
6002 _h	<i>Set Groups of 8 Input Polarities</i>	unsigned 8
6003 _h	<i>Filter Constant Input 8-bit</i>	unsigned 8
6005 _h	<i>Global Interrupt Enable Digital</i>	boolean
6006 _h	<i>Interrupt Mask Any Change 8-bit</i>	unsigned 8
6007 _h	<i>Interrupt Mask Low-to-High 8-bit</i>	unsigned 8
6008 _h	<i>Interrupt Mask High-to-Low 8-bit</i>	unsigned 8
6200 _h	<i>Write Output 8-bit</i>	unsigned 8
6202 _h	<i>Change Polarity Output</i>	unsigned 8
6206 _h	<i>Error Mode Output 8-bit</i>	unsigned 8
6207 _h	<i>Error Value Output 8-bit</i>	unsigned 8



8.10.2 Zusammenhang der implementierten Objekte für die digitalen Eingänge

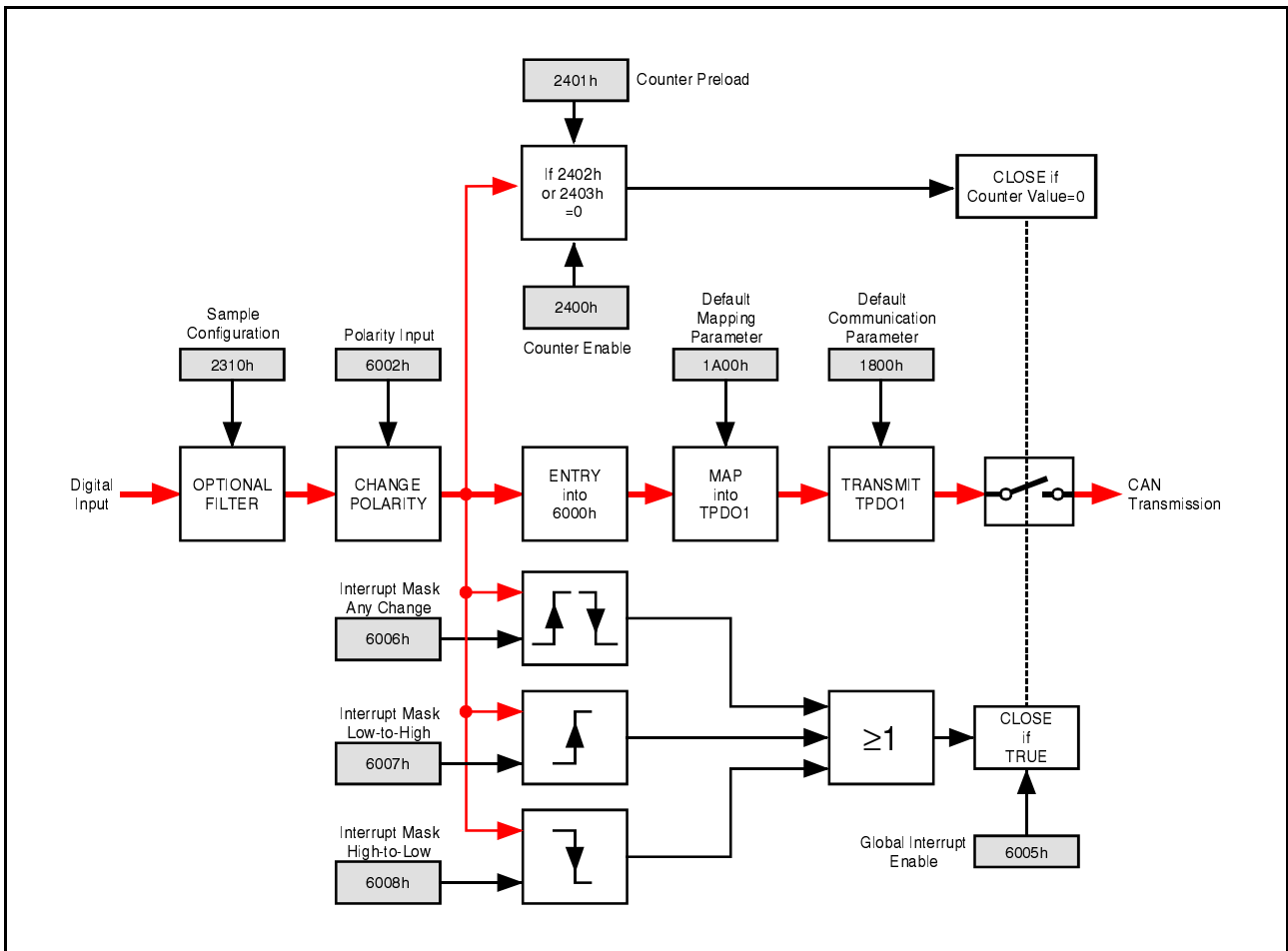
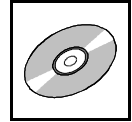


Abb. 22: Übersicht der Objekte für die digitalen Eingänge



8.10.3 Zusammenhang der implementierten Objekte für die digitalen Ausgänge

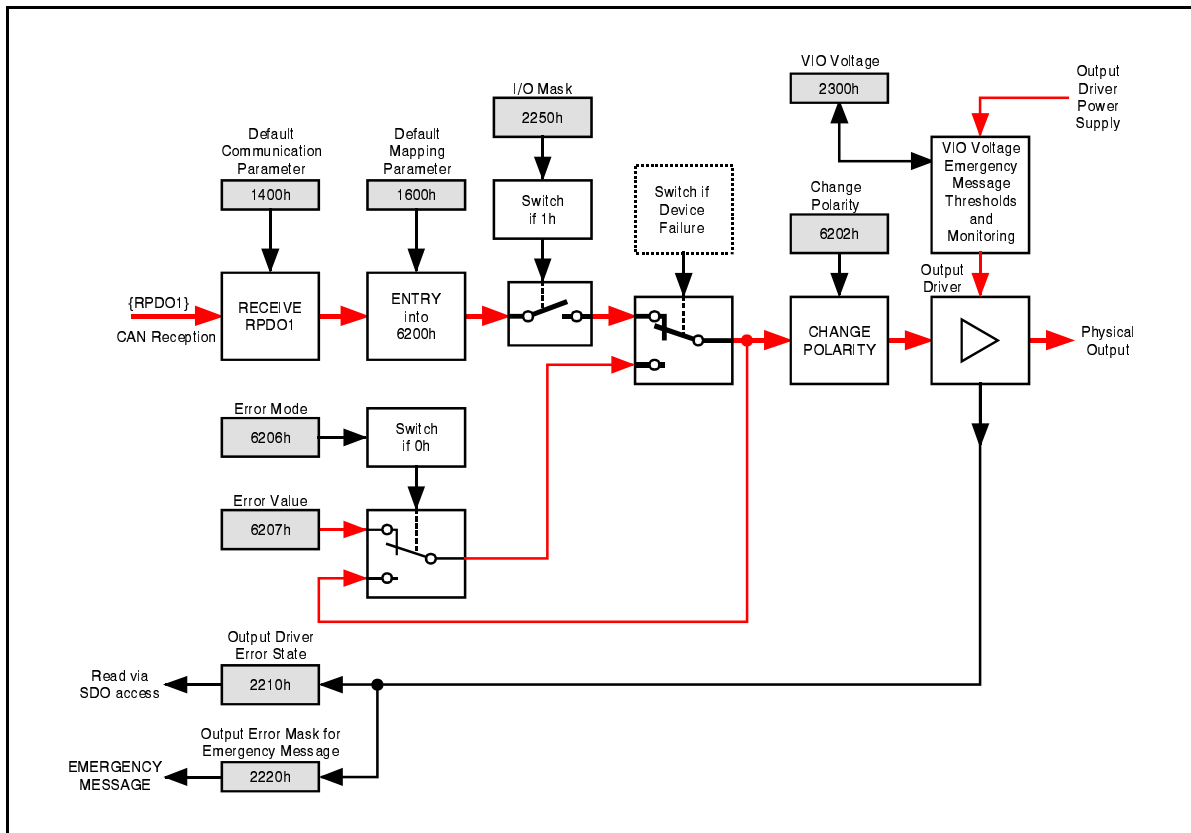
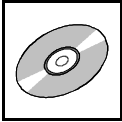


Abb. 23: Übersicht der Objekte für die digitalen Ausgänge



Device Profile Area

8.10.4 Read Input 8-bit (6000_h)

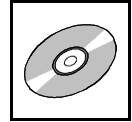
Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
6000 _h	0	Anzahl der Einträge	1	1	unsigned 8	ro
	1	Read_Input_DI8-DI1	0...FF _h	-	unsigned 8	ro

Belegung der Variablen *Read_Input_DI8-DI1*:

Index: 6000_h, Subindex: 1

Bit:	7	6	5	4	3	2	1	0
Eingang:	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

Wird ein Eingangsbit als '1' gelesen, so ist der entsprechende Eingang aktiv, d.h. Spannung liegt an (sofern das entsprechende Bit im Objekt 6002_h, "Polarity Input 8-Bit" auf '0' gesetzt ist).



8.10.5 Polarity Input 8-bit (6002_h)

Mit diesem Objekt kann die Polarität der 8 digitalen Eingänge individuell gesetzt werden.

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
6002 _h	0	Anzahl der Einträge	1	1	unsigned 8	rw
	1	Polarity_Input_DI8-DI1	0...FF _h	0	unsigned 8	rw

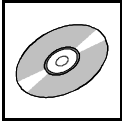
Belegung der Variablen *Polarity_Input_DI8-DI1*:

Index: 6002_h, Subindex: 1

Bit:	7	6	5	4	3	2	1	0
Eingang:	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

Bitwert DI _x	Eingang
1	invertiert
0	nicht invertiert

x ... 1 - 8



Device Profile Area

8.10.6 Filter Constant Input 8-bit (6003_h)

Mit dieser Variablen kann der digitale Filter für jeden Eingang separat aktiviert werden. Über das Objekt 2310_h wird die Sample-Rate der digitalen Eingänge definiert.

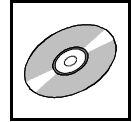
Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
6003 _h	0	Anzahl der Einträge	1	1	unsigned 8	rw
	1	Filter Constant <i>_DI8-DI1</i>	0...FF _h	0	unsigned 8	rw

Belegung der Variablen *Filter Constant _DI8-DI1*:

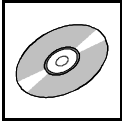
Index: 6003_h, Subindex: 1

Bit:	7	6	5	4	3	2	1	0
Eingang:	<i>DI8</i>	<i>DI7</i>	<i>DI6</i>	<i>DI5</i>	<i>DI4</i>	<i>DI3</i>	<i>DI2</i>	<i>DI1</i>

Bitwert <i>DIx</i>	Bedeutung
1	Filter enabled
0	Filter disabled

**8.10.7 Global Interrupt Enable Digital (6005_h)**

Index	Sub-index	Beschreibung	Wertebereich [Boolean]	Default	Datentyp	Zugriff
6005_h	0	<i>Global Interrupt Enable-Bit</i> = false : interrupts disabled <i>Global Interrupt Enable-Bit</i> = true : interrupts enabled	True, False	1	boolean	rw



Device Profile Area

8.10.8 Interrupt Mask Any Change 8-bit (6006_h)

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
6006 _h	0	Anzahl der Einträge	1	1	unsigned 8	const
	1	IRQ-Mask_Any_8_DI8-DI1	0...FF _h	FF _h	unsigned 8	rw

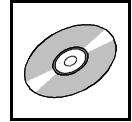
Belegung der Variablen *IRQ-Mask_Any_8_DI8-DI1*:

Index: 6006_h, Subindex: 1

Bit:	7	6	5	4	3	2	1	0
Eingang:	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

Bei aktiviertem IRQ wird bei **fallender oder steigender** Flanke des Eingangssignals ein Interrupt ausgelöst.

Bit-Wert <i>DI_x</i>	Interrupt-Freigabe
0	Interrupt disabled
1	Interrupt enabled



8.10.9 Interrupt Mask Low to High 8-bit (6007_h)

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
6007 _h	0	Anzahl der Einträge	1	1	unsigned 8	ro
	1	IRQ-Mask_LH_8_DI8-DI1	0...FF _h	0	unsigned 8	rw

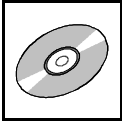
Belegung der Variablen *IRQ-Mask_LH_8_DI8-DI1*:

Index: 6007_h, Subindex: 1

Bit:	7	6	5	4	3	2	1	0
Eingang:	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

Bei aktiviertem IRQ wird bei **steigender** Flanke des Eingangssignals ein Interrupt ausgelöst.

Bit-Wert <i>DI_x</i>	Interrupt-Freigabe
0	Interrupt disabled
1	Interrupt enabled



Device Profile Area

8.10.10 Interrupt Mask High to Low 8-bit (6008_h)

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
6008 _h	0	Anzahl der Einträge	1	1	unsigned 8	ro
	1	IRQ-Mask_HL_8_DI8-DI1	0...FF _h	0	unsigned 8	rw

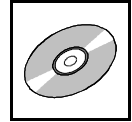
Belegung der Variablen *IRQ-Mask_HL_8_DI8-DI1*:

Index: 6008_h, Subindex: 1

Bit:	7	6	5	4	3	2	1	0
Eingang:	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1

Bei aktiviertem IRQ wird bei **fallender** Flanke des Eingangssignals ein Interrupt ausgelöst.

Bit-Wert DI_x	Interrupt-Freigabe
0	Interrupt disabled
1	Interrupt enabled



8.10.11 Write Output 8-Bit (6200_h)

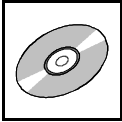
Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
6200 _h	0	Anzahl der Einträge	1	1	unsigned 8	const
	1	Write_output_DO8-DO1	0...FF _h	0	unsigned 8	rw

Belegung der Variablen *Write_output_DO8-DO1*:

Index: 6200_h, Subindex: 1

Bit:	7	6	5	4	3	2	1	0
Ausgang:	DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1

Wird ein Ausgangsbit auf '1' gesetzt, so wird der entsprechende Ausgang aktiviert, d.h. die Ausgangsspannung liegt an.



Device Profile Area

8.10.12 Change Polarity Output 8-bit (6202_h)

Dieses Objekt definiert den Zustand der 8 digitalen Ausgänge. Jeder Ausgang kann einzeln invertiert werden.

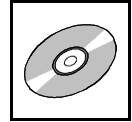
Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
6202 _h	0	<i>number_of_output 8-bit</i>	1	1	unsigned 8	const
	1	<i>change_polarity_output_DO8-DO1</i>	0...FF _h	0	unsigned 8	rw

Belegung der Variablen *change_polarity_output_DO8-DO1*:

Diese Variable bestimmt welcher digitale Ausgang invertiert wird.

Bit:	7	6	5	4	3	2	1	0
Ausgang:	<i>DO8</i>	<i>DO7</i>	<i>DO6</i>	<i>DO5</i>	<i>DO4</i>	<i>DO3</i>	<i>DO2</i>	<i>DO1</i>

Wird ein Ausgangsbit auf '1' gesetzt, so wird der entsprechende Ausgang invertiert.



8.10.13 Error Mode Output 8-bit (6206_h)

Der Error Mode wird ausgewertet, wenn das Modul in den Zustand *Stopped* gebracht wird. Dieses Objekt bestimmt zusammen mit Objekt 1029_h, ob die Ausgänge beim Auftreten eines internen Device-Fehlers auf einen in Objekt 6207_h bestimmten Error-Wert gesetzt werden.

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
6206 _h	0	<i>number_of_output_8 bit</i>	1	1	unsigned 8	const
	1	<i>error_mode_output_DO8-DO1</i>	0...FF _h	FF _h	unsigned 8	rw

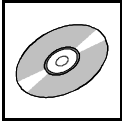
Belegung der Variablen *error_mode_output_DO8-DO1*:

Diese Variable zeigt an, welcher Ausgang auf einen vordefinierten Error-Wert (siehe Objekt 6207_h) gesetzt wird, wenn ein interner Device-Fehler auftritt.

1 = Ausgang soll auf einen in Objekt 6207_h definierten Wert gesetzt werden.

0 = Ausgang soll auch bei aufgetretenem Fehler nicht verändert werden.

Bit:	7	6	5	4	3	2	1	0
Ausgang:	<i>DO8</i>	<i>DO7</i>	<i>DO6</i>	<i>DO5</i>	<i>DO4</i>	<i>DO3</i>	<i>DO2</i>	<i>DO1</i>



8.10.14 Error Value Output 8-bit (6207_h)

Bei internen Gerätefehlern werden die Ausgänge auf den in diesem Objekt definierten Wert gesetzt, wenn der zugehörige Error Mode (Objekt 6206_h) aktiviert ist.

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
6207 _h	0	<i>number_of_output 8 bit</i>	1	1	unsigned 8	const
	1	<i>error_value_output_DO8-DO1</i>	0...FF _h	0	unsigned 8	rw

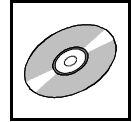
Belegung der Variablen *error_value_output_DO8-DO1*:

Index: 6207_h, Subindex: 1

Bit:	7	6	5	4	3	2	1	0
Ausgang:	<i>DO8</i>	<i>DO7</i>	<i>DO6</i>	<i>DO5</i>	<i>DO4</i>	<i>DO3</i>	<i>DO2</i>	<i>DO1</i>

Diese Variable enthält den Wert auf den die Ausgänge gesetzt werden, wenn ein Fehler aufgetreten ist.

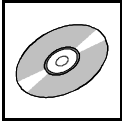
- 0 = Im Fehlerfall wird der Ausgang auf '0' (disabled) gesetzt, wenn Objekt 6206_h aktiviert ist.
- 1 = Im Fehlerfall wird der Ausgang auf '1' (enabled) gesetzt, wenn Objekt 6206_h aktiviert ist.



8.11 Manufacturer Specific Profile Area

8.11.1 Übersicht der Objekte 2210_h ... 2403_h

Index	Name	Datentyp
2210 _h	<i>Output Errors 8-Bit</i>	unsigned 8
2220 _h	<i>Output Error Mask Bit 1 to 8</i>	unsigned 8
2250 _h	<i>I/O Direction Mask 8-Bit</i>	unsigned 8
2300 _h	<i>VIO Voltage 16-Bit</i>	unsigned 16
2310 _h	<i>Sample Configuration</i>	unsigned 16
2400 _h	<i>Counter Enable</i>	unsigned 8
2401 _h	<i>Counter Preload</i>	unsigned 32
2402 _h	<i>Counter Value 16- Bit</i>	unsigned 16
2403 _h	<i>Counter Value 32-Bit</i>	unsigned 32



Manufacturer Specific Profile Area

8.11.2 Output Errors 8-Bit (2210_h)

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
2210 _h	0	Anzahl der Einträge	2	2	unsigned 8	ro
	1	<i>output_error_open_DO8-DO1</i>	0..FF _h	0	unsigned 8	ro
	2	<i>output_error_short_DO8-DO1</i>	0..FF _h	0	unsigned 8	ro

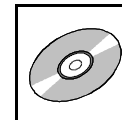
Dieses Objekt gibt eine Bitmaske zurück, aus der hervorgeht bei welchen Ausgängen Fehler (Fehlerzustände: offen, überlast) aufgetreten sind.

Belegung der Variablen *output_errors_open_DO8-DO1*, bzw. *output_errors_short_DO8-DO1*:

Index: 2210_h, Subindex: 1/2

Bit:	7	6	5	4	3	2	1	0
Ausgang:	<i>error_ DIO8</i>	<i>error_ DIO7</i>	<i>error_ DIO6</i>	<i>error_ DIO5</i>	<i>error_ DIO4</i>	<i>error_ DIO3</i>	<i>error_ DIO2</i>	<i>error_ DIO1</i>

Ist an einem Ausgang ein Fehler aufgetreten, werden die entsprechenden Ausgangsbits auf '1' gesetzt.



8.11.3 Output Error Mask Bit 1 to 8 (2220_h)

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
2220 _h	0	Anzahl der Einträge	2	2	unsigned 8	ro
	1	<i>output_error_mask_open_DO8-DO1</i>	0...FF _h	FF _h	unsigned 8	rw
	2	<i>output_error_mask_short_DO8-DO1</i>	0...FF _h	FF _h	unsigned 8	rw

Mit diesem Objekt können Bitmasken gesetzt werden, die bestimmen, ob beim Auftreten eines Fehlers eine Emergency Message (siehe auch Objekt 1003_h und 1014_h) gesendet wird oder nicht.

Belegung der Variablen *output_error_mask_open_DO8-DO1*, bzw. *output_error_mask_short_DO8-DO1*:

Index: 2220_h, Subindex: 1/2

Bit:	7	6	5	4	3	2	1	0
Ausgang:	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1



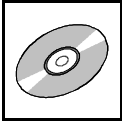
HINWEIS

Eine Emergency Message wird beim ersten Auftreten eines Fehlers der digitalen Ausgangstreiber gesetzt, wenn das entsprechende Bit in *Output Error Mask Bit 1 to 8* auf '1' gesetzt ist. Tritt bei einem weiteren Ausgang eine Fehlermeldung auf, wird keine weitere Emergency Message gesendet. Sobald die letzte Fehlermeldung zurückgenommen wird, wird wiederum eine Emergency Message gesendet.



HINWEIS

Auf die aktuellen Fehlerzustände, die über Objekt *Output Errors 8-Bit* (2210_h) gelesen werden, hat das Objekt *Output Error Mask Bit 1 to 8* keinen Einfluss.



Manufacturer Specific Profile Area

8.11.4 I/O Direction_Mask 8-Bit (2250_h)

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
2250 _h	0	Anzahl der Einträge	1	1	unsigned 8	ro
	1	I/O_direction_mask_DIO8-DIO1	0...FF _h	0	unsigned 8	rw

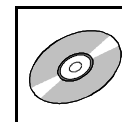
Mit diesem Objekt kann eine Bitmaske gesetzt werden, die bestimmt welcher der 8 I/O-Kanäle als Eingang oder Ausgang verwendet werden soll.

Belegung der Variablen I/O_direction_mask_DIO8-DIO1:

Index: 2250_h, Subindex: 1

Bit:	7	6	5	4	3	2	1	0
Port:	DIO8	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1

Wert	Bedeutung
1	Kanal wird als Ausgang verwendet.
0	Kanal wird als Eingang verwendet.



8.11.5 VIO Voltage 16-Bit (2300_h)

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
2300 _h	0	Anzahl der Einträge	4	4	unsigned 8	ro
	1	VIO_voltage	0..FFFF _h	-	unsigned 16	ro
	2	VIO_low_emergency_voltage	0..FFFF _h	B4 _h	unsigned 16	rw
	3	VIO_high_emergency_voltage	0..FFFF _h	118 _h	unsigned 16	rw

Mit diesem Objekt kann die Betriebsspannung für die Digitalausgänge als Analogwert ausgelesen werden und es können die Schwellen für die Betriebsspannung der Digitalausgänge eingestellt werden, bei deren Über- oder Unterschreitung eine Emergency Message ausgelöst wird.

Berechnung der Spannungswerte für die Subindizes 1, 2, 3:

Wert der Variablen = (Spannung in V) x 10 _d
--

Belegung der Variablen VIO_voltage (Index: 2300_h, Subindex: 1):

Diese Variable gibt den Messwert der Betriebsspannung für die Digitalausgänge zurück.

Beispiel:

Wert der Variablen VIO_Voltage = F0_h = 240_d

Spannung in V = Wert der Variablen / 10_d = 240_d / 10_d = 24,0 V

Belegung der Variablen VIO_low_emergency_voltage (Index: 2300_h, Subindex: 2):

Mit dieser Variablen wird die Spannung eingestellt, die eine Emergency-Message auslöst, sobald sie **unterschritten** wird.

Beispiel:

0 = disable

B4_h = 180_d (Wert der Variablen)

180_d/10_d = **18 V** Spannung, **unterschreiten** dieser Schwelle löst eine Emergency aus.

Belegung der Variablen VIO_high_emergency_voltage (Index: 2300_h, Subindex: 3):

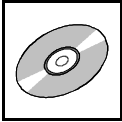
Mit dieser Variablen wird die Spannung eingestellt, die eine Emergency-Message auslöst, sobald sie **überschritten** wird.

Beispiel:

0 = disable

118_h = 280_d (Wert der Variablen)

280_d/10_d = **28 V** Spannung, **überschreiten** dieser Schwelle löst eine Emergency aus.



Manufacturer Specific Profile Area

8.11.6 Sample Configuration (2310_h)

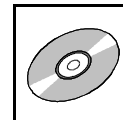
Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
2310 _h	0	<i>Anzahl der Einträge</i>	1	1	unsigned 8	ro
	1	<i>sample_rate</i>	C8 _h ...FFFF _h	C8 _h	unsigned 16	rw

Mit diesem Objekt kann der Filter für die digitalen Eingänge aktiviert und konfiguriert werden.

Belegung der Variablen *Sample Rate*:

Diese Variable definiert die Sample-Rate, mit der die digitalen Eingänge abgefragt werden. Die Sample-Rate wird in Schritten zu 0,5 µs eingestellt. Der Minimalwert der Sample-Rate von 100 µs kann nicht unterschritten werden:

Wert der Variablen <i>sample_rate</i>	Eingestellte Sample-Rate [µs]
≤ 199	keine Auswertung (alter Wert bleibt erhalten + SDO-Fehlermeldung)
200	100 (Default-Wert)
201	100,5
:	:
65535	32767,5



8.11.7 Funktionsweise der Counter

Für jeden Eingang kann ein 32-Bit-Counter aktiviert werden (Objekt 2400_h). Die flankensensitiven Counter zählen in der Default-Einstellung die steigenden Flanken der Eingangssignale. Über Anpassung des Parameters *Polarity_Input* (Objekt 6002_h) kann erreicht werden, dass die fallenden Flanken gezählt werden.

Der Startwert des 32-Bit-Counters *counter_preload_x* wird über das Objekt 2401_h vorgegeben. Der Counter wird mit jedem Schritt um '1' dekrementiert. Der aktuelle Counter-Wert ist im entsprechenden *counter_value_x* (siehe Objekte 2402_h und 2403_h) abgelegt.

Der 32-Bit Counter kann als Event Timer für Sende-PDOs verwendet werden.

Erreicht der 32-Bit-Counter (*counter_value_x_32-bit*) den Wert '0' (d.h. der Event Timer ist abgelaufen), so wird ein Event generiert. Durch dieses Event wird dann die Versendung eines Tx-PDOs ausgelöst, wenn die folgenden Bedingungen dazu erfüllt sind:

- Der Counter muss gemappt sein.
Diese Einstellung ist für den Transmit PDO2-Mapping Parameter als Default-Wert enthalten (siehe, Objekt 1A01_h, *object_to_be_mapped_1-4*).
- Das PDO2-Mapping muss über Objekt 1801_h zugelassen sein.
Dafür darf für den Parameter *COB-ID used by PDO2* das oberste Bit **nicht gesetzt** sein.
Per Default ist dieses Bit gesetzt, siehe Objekt 1801_h, Subindex 1, d.h. in der Default-Einstellung ist der Event-Timer nicht aktiviert.
- Die Übertragungsart *ereignisgesteuert, asynchron* muss für das Tx-PDO2 konfiguriert sein, d.h. der PDO-Übertragungstyp muss 254 oder 255 sein.
Per Default ist die Übertragungsart 255 gesetzt, siehe Objekt 1801_h, Subindex 2.

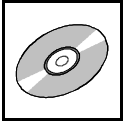
Nach Ablauf des Counters (*counter_value_x_32-bit* = '0') und damit des Event Timers wird der *counter_value_x_32-bit* wieder auf den *counter_preload_x*-Wert gesetzt ($x = 1...8$).

Die Zusammenhänge der Objekte der digitalen Eingänge sind in der Abbildung 22 auf Seite 94 anschaulich dargestellt.



HINWEIS

Wird die Counter-Funktion eines Eingangs benutzt, so ist dringend anzuraten, die lokalen Interrupts über die *Interrupt Mask*-Objekte 6006_h, 6007_h und 6008_h zu unterbinden (entlastet den CAN-Bus).



Manufacturer Specific Profile Area

Die maximal mögliche Zählfrequenz ergibt sich aus der Abtastrate der Eingänge (siehe Objekt *Sample Configuration 2310_h*):

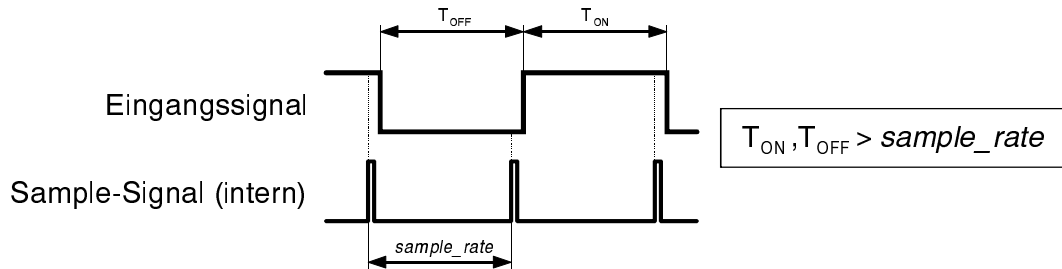
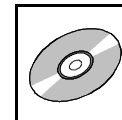


Abb. 24: Zusammenhang zwischen maximaler Zählfrequenz und eingestellter Abtastrate

Damit ergibt sich ein absolutes Maximum der zählbaren Eingangsfrequenz von 5 kHz.



8.11.8 Counter Enable (2400_h)

Index	Subindex	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
2400 _h	0	<i>counter_enable</i>	0...FF _h	0	unsigned 8	rw

Mit diesem Objekt können die Counter 1..8 aktiviert werden. Die Counter zählen die Anzahl der steigenden Flanken der digitalen Eingänge 1..8.

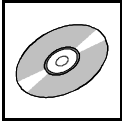
Über Anpassung des Parameters *Polarity_Input* (Objekt 6002_h) kann erreicht werden, dass die fallenden Flanken gezählt werden.

Belegung der Variablen *counter enable*:

Index: 2400_h, Subindex: 0

Bit:	7	6	5	4	3	2	1	0
Port:	<i>counter8</i>	<i>counter7</i>	<i>counter6</i>	<i>counter5</i>	<i>counter4</i>	<i>counter3</i>	<i>counter2</i>	<i>counter1</i>

Wert	Bedeutung
1	Counter aktiviert (enabled)
0	Counter nicht aktiv (disabled)

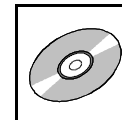


Manufacturer Specific Profile Area

8.11.9 Counter Preload (2401_h)

Index	Subindex	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
2401 _h	0	<i>no_of_objects</i>	1...8	8	unsigned 8	ro
	1	<i>counter_preload_1</i>	0... FFFF FFFF _h	0	unsigned 32	rw
	2	<i>counter_preload_2</i>		0	unsigned 32	rw
	3	<i>counter_preload_3</i>		0	unsigned 32	rw
	4	<i>counter_preload_4</i>		0	unsigned 32	rw
	5	<i>counter_preload_5</i>		0	unsigned 32	rw
	6	<i>counter_preload_6</i>		0	unsigned 32	rw
	7	<i>counter_preload_7</i>		0	unsigned 32	rw
	8	<i>counter_preload_8</i>		0	unsigned 32	rw

Dieses Objekt enthält die Startwerte *counter_preload_1...8*, von denen die Counter anfangen zu zählen. Beginnend vom Startwert *counter_preload_x* wird der Wert bei Auftreten einer Änderung um 1 dekrementiert ($x = 1...8$).



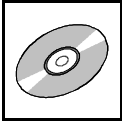
8.11.10 Counter Value 16-Bit (2402_h)

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
2402 _h	0	<i>no_of_objects</i>	1...8	8	unsigned 8	ro
	1	<i>counter_value_1_16-bit</i>	0...FFFF _h	0	unsigned 16	ro
	2	<i>counter_value_2_16-bit</i>		0	unsigned 16	ro
	3	<i>counter_value_3_16-bit</i>		0	unsigned 16	ro
	4	<i>counter_value_4_16-bit</i>		0	unsigned 16	ro
	5	<i>counter_value_5_16-bit</i>		0	unsigned 16	ro
	6	<i>counter_value_6_16-bit</i>		0	unsigned 16	ro
	7	<i>counter_value_7_16-bit</i>		0	unsigned 16	ro
	8	<i>counter_value_8_16-bit</i>		0	unsigned 16	ro

Dieses Objekt enthält die niederwertigen 16 Bit *counter_value_x_16-bit* der aktuellen 32-Bit-Werte der Counter 1...8 (siehe auch Objekt 2400_h) ($x = 1...8$).

Die folgende Tabelle zeigt die Position des 16-Bit-Counter-Wertes *counter_value_x_16-bit* im 32-Bit-Counter-Wert *counter_value_x_32-bit*:

Bit 31...		...Bit 0	
<i>counter_preload_x</i>			
<i>counter_value_x_32-bit</i>			
Bit 31...	...Bit 16	Bit 15...	...Bit 0
-		<i>counter_value_x_16-bit</i>	



Manufacturer Specific Profile Area

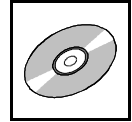
8.11.11 Counter Value 32-Bit (2403_h)

Index	Sub-index	Beschreibung	Wertebereich	Default	Datentyp	Zugriff
2403 _h	0	<i>no_of_objects</i>	1...8	8	unsigned 8	ro
	1	<i>counter_value_1_32-bit</i>	0... FFFF FFFF _h	0	unsigned 32	ro
	2	<i>counter_value_2_32-bit</i>		0	unsigned 32	ro
	3	<i>counter_value_3_32-bit</i>		0	unsigned 32	ro
	4	<i>counter_value_4_32-bit</i>		0	unsigned 32	ro
	5	<i>counter_value_5_32-bit</i>		0	unsigned 32	ro
	6	<i>counter_value_6_32-bit</i>		0	unsigned 32	ro
	7	<i>counter_value_7_32-bit</i>		0	unsigned 32	ro
	8	<i>counter_value_8_32-bit</i>		0	unsigned 32	ro

Dieses Objekt enthält die aktuellen 32-Bit- Werte *counter_value_x_32-bit* der Counter 1..8 (siehe auch Objekt 2400_h).

Die Counter zählen die Anzahl der Änderungen der digitalen Eingänge 1..8. Beginnend vom 32-Bit-Startwert *counter_preload_x* wird der Wert bei Auftreten einer Änderung um 1 dekrementiert ($x = 1..8$).

Es wird ein Event generiert, wenn der **32-Bit**-Counter den Wert '0' erreicht. Wird für *counter_preload_x* der Wert von '0000.0000' eingetragen, so ergibt sich ein Preload von 2^{32} !



8.12 Programm Download über CiA DSP-302-Objekte

Die im folgenden beschriebenen Objekte werden benötigt, wenn die Firmware des CAN-CBX-DIO8 ausgetauscht werden soll.



ACHTUNG

Das Firmware-Update sollte nur durch erfahrene Anwender durchgeführt werden!

Unsachgemäße Durchführung des Updates kann zum ungewollten Löschen des Speichers und zum Verlust der Firmware führen. Das Modul kann dann nicht weiter betrieben werden!



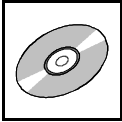
HINWEIS

Für ein Firmware-Update bietet esd das Programm CANfirmdown. Bitte wenden Sie sich dazu an unseren Support.

Im normalen CiA 301-Modus kann auf die Objekte 1F50_h und 1F52_h nicht zugegriffen werden. Das Objekt 1F51_h steht auch im DSP-301-Modus zur Verfügung.

Weitere Informationen zu den Objekten und zum Firmware-Update entnehmen Sie bitte [2].

Index	Sub-index	Beschreibung	Datentyp	Zugriff
1F51 _h	1	Boot-Loader: FLASH-Kommando	unsigned 8	rw
1F52 _h	0,1,2	Boot-Loader: Firmware-Datum	unsigned 32	ro



Programm Download über CiA DSP-302-Objekte

8.12.1 Download-Steuerung über Objekt 1F51_h

INDEX	1F51_h
Name	Program Control

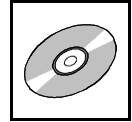
Index	Sub-index	Beschreibung	Wertebereich	Default-Wert	Datentyp	Zugriff
1F51 _h	0	<i>number of programs</i>	1	1	unsigned 8	ro
	1	<i>program_number_1</i>	0...2	1	unsigned 8	rw



HINWEIS

Der Wertebereich dieses Objektes in der Implementierung für die CAN-CBX-DIO8 weicht vom Wertebereich, der in der [2] angegeben ist, ab.

Weitere Informationen zum Objekt 1F51_h und zum Firmware-Update entnehmen Sie bitte dem Standard [2].



8.12.2 Download-Steuerung über Objekt 1F52_h

INDEX	1F52_h
Name	Verify Application Software

In diesem Objekt werden Datum und Tageszeit der Erstellung der Applikationssoftware abgelegt, um zu einem späteren Zeitpunkt prüfen zu können, ob die gespeicherte Software der erwarteten entspricht.

Index	Sub-index	Beschreibung	Wertebereich	Default-Wert	Datentyp	Zugriff
1F52_h	0	<i>number of entries</i>	2	2	unsigned 8	ro
	1	<i>application_software_date</i>	0...FFFF FFFF _h	-	unsigned 32	ro
	2	<i>application_software_time</i>	0...FFFF FFFF _h	-	unsigned 32	ro

Bedeutung der Variablen:

application_software_date Datum der letzten Applikationssoftware des Moduls, angegeben in Tagen seit dem 01.01.1984.
 Beispiel: 0000 196D_h -> 26.10.2001

application_software_time Zeit in ms seit Mitternacht am Tag der Erstellung der letzten Applikationssoftware.
 Beispiel: 02A0AB8E_h -> ca. 12:14 Uhr



9. Referenzen

- [1] CiA 301 Standard
CANopen Application Layer and Communication Profile V4.2 (02.2011)
- [2] CiA DSP 302
CANopen Additional Application Layer Functions V4.1 (02.2009)
- [3] CiA 303
CANopen Additional Specification V1.7 (12.2009)
- [4] Phoenix Contact GmbH & Co. KG, Blomberg.
Technische Daten entnommen aus COMBICON Online-Katalog,
<https://www.phoenixcontact.com/online/portal/de>
Leiterplattensteckverbinder - FKCT-2,5/4-ST KMGY - 1921900, Abgerufen am 09.10.2013
- [5] Phoenix Contact GmbH & Co. KG, Blomberg.,
Technische Daten entnommen aus COMBICON Online-Katalog,
<https://www.phoenixcontact.com/online/portal/de>
Leiterplattensteckverbinder - FK-MCP 1,5/ 5-STF-3,81 - 1851261, Abgerufen am 09.10.2013
- [6] Phoenix Contact GmbH & Co. KG, Blomberg.,
Technische Daten entnommen aus COMBICON Online-Katalog,
<https://www.phoenixcontact.com/online/portal/de>
Leiterplattensteckverbinder -FMC 1,5/10-ST-3,5 - 1952348, Abgerufen am 09.10.2013

10. Konformitätserklärung

EU-KONFORMITÄTSERKLÄRUNG EU DECLARATION OF CONFORMITY



Adresse **esd electronic system design gmbh**
Address **Vahrenwalder Str. 207**
30165 Hannover
Germany

esd erklärt, dass das Produkt
esd declares, that the product

CAN-CBX-DIO8
CAN-CBX-DIO8-J1939

Typ, Modell, Artikel-Nr.
Type, Model, Article No.

C.3010.02
C.3010.05

die Anforderungen der Normen
fulfills the requirements of the standards

EN 61000-6-2:2005,
EN 61000-6-4:2007+A1:2011

gemäß folgendem Prüfbericht erfüllt.
according to test certificate.

H-K01-0247-07

Das Produkt entspricht damit der EU-Richtlinie „EMV“
Therefore the product corresponds to the EU Directive 'EMC'

2014/30/EU

Das Produkt entspricht der EU-Richtlinie „RoHS“
The product corresponds to the EU Directive 'RoHS'

2011/65/EU

Diese Erklärung verliert ihre Gültigkeit, wenn das Produkt nicht den Herstellerunterlagen entsprechend eingesetzt und betrieben wird, oder das Produkt abweichend modifiziert wird.
This declaration loses its validity if the product is not used or run according to the manufacturer's documentation or if non-compliant modifications are made.

Name / Name T. Ramm
Funktion / Title CE-Koordinator / CE Coordinator
Datum / Date Hannover, 2015-01-21

Rechtsgültige Unterschrift / authorized signature

!:\Texte\Doku\MANUALS\CAN\CBX\DIO8\CAN-CBX-DIO8-Konformitaetserklaerung_2015-01-21.odt



11. Bestellhinweise


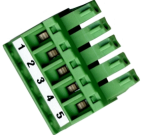

Typ	Eigenschaften	Bestell-Nr.
CAN-CBX-DIO8	-8 digitale I/O-Kanäle, 24 V Eingangsspannung, Ausgangsstrom 1 A, - 2-Leiter Anschluss - jeder getrennt als Eingang oder Ausgang programmierbar, - In-Rail-Bus 1x CAN-CBX-TBUS (C.3000.01)	C.3010.02
Zubehör		
 CAN-CBX-TBUS	Tragschienen-Busverbinder des CBX-InRailBus für CAN-CBX-Module, (ein Busverbinder ist im Lieferumfang des CAN-CBX- Moduls enthalten) Phoenix Contact Bestellangaben.: ME 22,5 TBUS 1,5/5- ST-3,81 KMGY (2713722)	C.3000.01
 CAN-CBX-TBUS- Connector	Anschlussstecker des CBX-InRailBus zum Anschluss der +24V Versorgungsspannung und des CAN-Interface, Buchsenkontakte Phoenix Contact Bestellangaben: MCVR 1,5/ 5-STF-3,81 AU - (1846631)	C.3000.02
 CAN-CBX-TBUS- Connection adapter	Anschlussstecker des CBX-InRailBus zum Anschluss der +24V Versorgungsspannung und des CAN-Interface, Stiftkontakte Phoenix Contact Bestellangaben: IMC 1,5/ 5-ST-3,81 AU (1943276)	C.3000.03

Tabelle 14: Bestellhinweise

PDF-Handbücher

Handbücher sind in Englisch und üblicherweise auch in Deutsch erhältlich. Die Verfügbarkeit der Handbücher entnehmen Sie bitte der unteren Tabelle.

Die Handbücher im PDF-Format können Sie kostenlos von unserer Webseite www.esd.eu herunterladen.



Handbücher		Bestell-Nr.
CAN-CBX-DIO8-MD	Anwenderhandbuch in Deutsch	C.3010.20
CAN-CBX-DIO8-EN	Anwenderhandbuch in Englisch	C.3010.21

Tabelle 15: Verfügbare Handbücher

Gedruckte Handbücher

Benötigen Sie zusätzlich einen Ausdruck des Handbuches, kontaktieren Sie bitte unser Sales-Team sales@esd.eu für ein Angebot. Gedruckte Handbücher können gegen eine Gebühr bestellt werden.