

CAN-CBM-DIO8 E/A Modul

**zum Einsatz in
CANopen - Netzwerken**

Handbuch

Dokument-Datei:	I:\texte\Doku\MANUALS\CAN\CBM\DIO8\Deutsch\DIO8-083.ma9
Datum des Ausdrucks:	15.03.04

Änderungen in den Kapiteln

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

Kapitel	Änderungen gegenüber Vorversion
3.6	Bestellhinweise geändert
4.2.5	Prinzipschaltbild der Eingangs- und Ausgangsschaltungen eingefügt

Weitere technische Änderungen vorbehalten.

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.

esd electronic system design gmbh
Vahrenwalder Str. 207
30165 Hannover

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

Inhalt	Seite
1. Vorwort und Allgemeines	3
1.1 Bestimmungsgemäße Verwendung	3
2. Sicherheitshinweise	5
3. Kurzbeschreibung	5
3.1 Komponenten	5
3.2 Mechanische Daten	6
3.3 Elektrische Daten	6
3.3.1 Digitale Eingänge	6
3.3.2 Digitale Ausgänge	6
3.3.3 Anschluss des CAN-BUS	7
3.3.4 Anschluss Versorgungsspannung und I/O-Anschlüsse	7
3.4 Gehäusedesign	8
3.5 Software	8
3.6 Bestellhinweise	8
4. Digitale Ein/Ausgaben	9
4.1 Meldung der digitalen Eingänge	9
4.2 Digitale Ausgänge	9
4.2.1 Einschränkungen der Definition von E/A-Richtungen	10
4.2.2 Rücklesen der digitalen Ausgänge	10
4.2.3 Empfangsüberwachung der digitalen Ausgänge (für ‘Servo’-Modus)	11
4.2.4 Ausgangsstörungen	11
4.2.5 Eingangs- und Ausgangsschaltungen	12
5. Installation	13
5.1 Schutz des E/A-Moduls	13
5.2 CAN-Bus-Verdrahtung	13
6. Anzeigeelemente	14
7. CAN-Bus	17
7.1 CAN-Bus-Timing	17
7.2 CAN-Bus-Kanäle	18
7.3 Prozessdatenkanäle	19
7.3.1 Parameterkanal	20
7.3.2 Interne Master-Slave-Kommunikation	20
8. Parametrierung	22
8.1 Parametrierung mit dem DIP-Schalter	22
8.1.1 Belegung des DIP-Schalters (im Normalzustand):	22
8.1.2 Einstellung der CAN-Baudrate	23
8.1.3 Testmodus (nur für den Hersteller)	23
8.1.4 Programmiermodus	25
8.1.5 Belegung des DIP-Schalters im Programmiermodus	26
8.1.5.1 Programmierung der Ein-/ Ausgabefunktionen	26

8.1.5.2	Programmierung der CAN-Identifizierung und des Software-Modells . . .	26
8.1.5.3	Unterdrücken des E/A-Master/Slave-Verbundes	27
8.1.5.4	Beispiele für Programmierung mit dem DIP-Schalter	27
8.2	Parametrierung mit dem Parameterkanal (SDO)	28
8.2.1	Zugriff auf E/A-Codestellen über den Parameterkanal	28
8.2.1.1	Zeitliches Verhalten bei SDO-Schreibzugriffen	28
8.2.1.2	Interner Ablauf bei SDO-Schreibzugriffen	29
8.2.2	Hinweise zur Verwendung des Parameterkanals	30
9.	CANopen	31
9.1	Das CANopen-Objektverzeichnis	31
9.1.1	Kommunikationsprofil	32
9.1.2	Geräteprofil	33
9.1.2.1	Digitale Ein- und Ausgänge	33
9.1.2.2	Analoge Ein- und Ausgänge	34
9.1.3	Hersteller-spezifisches Profil	35
9.2	Service-Daten-Objekte (SDO's)	35
9.2.1	Aufbau der SDO-Telegramme	35
9.2.2	Nichtflüchtiges Speichern von Parametern	36
9.3	NMT-Boot-up	37
9.3.1	Einfaches Boot-Up	37
9.3.2	Extended Boot-Up	37
9.4	Prozessdatenobjekte (PDO's)	39
9.4.1	PDO-Mapping	40
9.4.2	PDO-Übertragungsarten	41
9.4.2.1	Synchrone Übertragungsarten	41
9.4.2.2	Ereignisgesteuerte Übertragung	41
9.4.3	Zyklische asynchrone Übertragung	42
9.4.4	Node Guarding / Life Guarding	42
9.4.5	Das Emergency-Telegramm	43
10.	Schnellstart	44
10.1	Konfiguration für den Einsatz im CANopen-Netzwerk	44
10.2	Konfiguration für den Einsatz im CANopen-Netzwerk	45
10.3	Tabelle der wichtigsten Identifizierung und Nachrichten für CANopen	46
11.	Anhang	47
11.1	Verwendete CAN-Identifizierung	47
11.2	Code-Tabellen	50

1. Vorwort und Allgemeines

Dieses Systemhandbuch ist als umfassende technische Information zu den E/A-Modulen der CANbloc-Mini-Serie (CBM) 'CAN-CBM-DIO8' vorgesehen.

Ein Großteil der Beschreibungen ist auch für andere CBM-Module gültig. Ergänzende Informationen für einzelne Module finden Sie in den speziellen Handbüchern.

Eine kurze Beschreibung zur Inbetriebnahme des CAN-CBM-DIO8-Moduls befindet sich im Kapitel 'Schnellstart' auf Seite 44.

1.1 Bestimmungsgemäße Verwendung

Falls das in dieser Dokumentation beschriebene Modul für den Einsatz mit einem Antriebsregler vorgesehen ist, sind grundsätzlich auch die Begriffsdefinitionen, Bestimmungen und Sicherheitshinweise aus den Handbüchern der verwendeten 'Servo'-Umrichter zu beachten.

Diese Seite ist bewusst unbedruckt.

2. Sicherheitshinweise

Der Betreiber bzw. sein Sicherheitsbeauftragter ist verpflichtet

- das Einhalten aller relevanten Vorschriften, Hinweise und Gesetze zu kontrollieren,
- zu gewährleisten, dass nur qualifiziertes Personal an und mit den Anlagenkomponenten arbeitet,
- zu gewährleisten, dass das Personal die Betriebsanleitung bei allen entsprechenden Arbeiten verfügbar hat und
- nichtqualifiziertem Personal das Arbeiten an und mit der Anlage zu untersagen.

3. Kurzbeschreibung

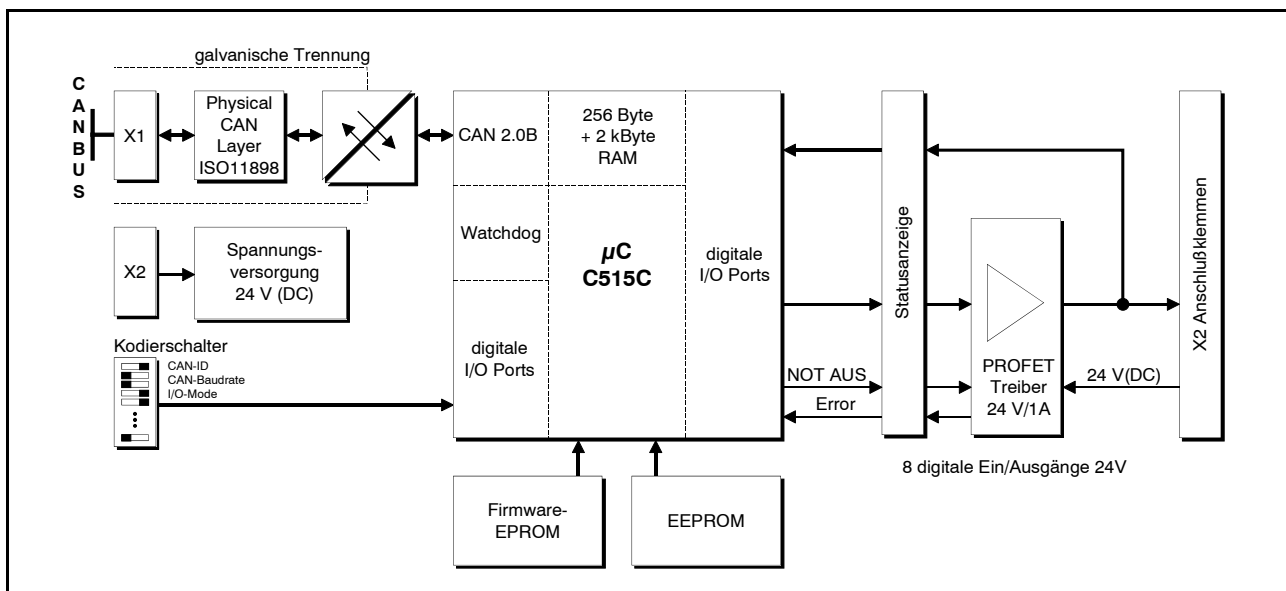


Abb.3: Blockschaltbild des CAN-CBM-DIO8-Moduls

3.1 Komponenten

- dezentrale E/A-Baugruppe für CAN-Bus mit acht frei als Ein- oder Ausgang verwendbaren Anschlussklemmen
- Hutschienen-Montage
- Anschlussstechnik: Schraub-/Steck-Klemmen
- zwei grüne Zustandsanzeige- LED's ('Betrieb'[unten] und 'Mode')
- zwei rote Fehleranzeige-LED's (Modul- und Ausgangsfehler)
- gelbe LED's zur Anzeige der E/A-Signal-Anzeige
- Prozessor-Typ: SAB 80C515C-L
- CAN-Bus-Transceiver: PCA82C250T (SMD) mit galvanischer Trennung

3.2 Mechanische Daten

- Gehäusotyp: UEGM-MSTB der Firma Phoenix Contact
Maße: Höhe: 79 mm
Tiefe: 90,5 mm (+10 mm für E/A-Anschlussklemme)
Breite: 25 mm
- Umgebungstemperatur: 0...50 °C oder erweiterter Temperaturbereich: -20...+70 °C
- Position des DIP-Schalters an der oberen Gehäuseseite (im aufgeschnappten Zustand)
- Position des CAN-Bus-Steckverbinders an der unteren Gehäuseseite
- Position der grünen Betriebs-LED unten

3.3 Elektrische Daten

- Spannungsversorgung: 18 bis 30 Volt DC, typisch 24 V
- Stromaufnahme: typisch 30 mA, max. 80 mA (bei ausgeschalteten Ausgängen)

3.3.1 Digitale Eingänge

- keine galvanische Trennung
- Eingangswiderstand: $3 \text{ k}\Omega \leq R_i \leq 4 \text{ k}\Omega$
- Eingangsspannungsbereich 'low': 0...5 V
- Eingangsspannungsbereich 'high': 13...30 V

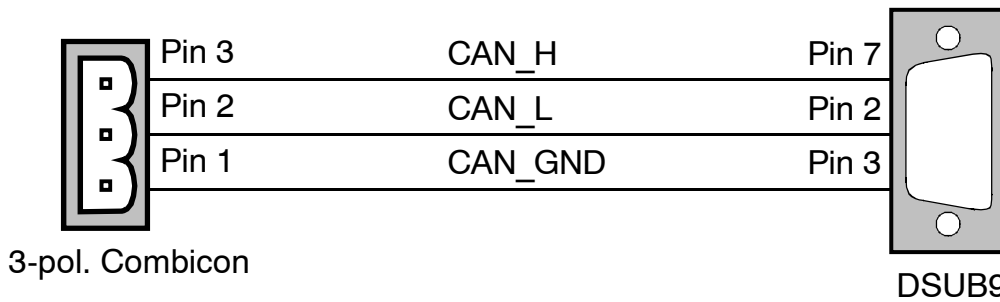
3.3.2 Digitale Ausgänge

- keine galvanische Trennung
- Typ: 'PNP' (High-Side-Treiber: Plus-schaltend)
- kurzschlussfest
- maximaler Ausgangsstrom: ca. 1 A (pro Kanal, Treiber-Spezifikation!)
- Erkennung von Kurzschluss/Überlast ab ca. 4 A (pro Kanal)
- Ausgangsspannung 'low': über Eingangswiderstände 0 V (bei offenem Ausgang)
- Ausgangsspannung 'high': $> U_b - 1,8 \text{ V}$ (bei maximaler Last, Versorgung mit 24 V)
- der aktuelle Zustand der Ausgänge kann zurückgelesen werden
- gemeinsame Kurzschluss-Statusanzeige mit einer roten LED
- Anzeige durch rote LED, wenn Summenstrom überschritten

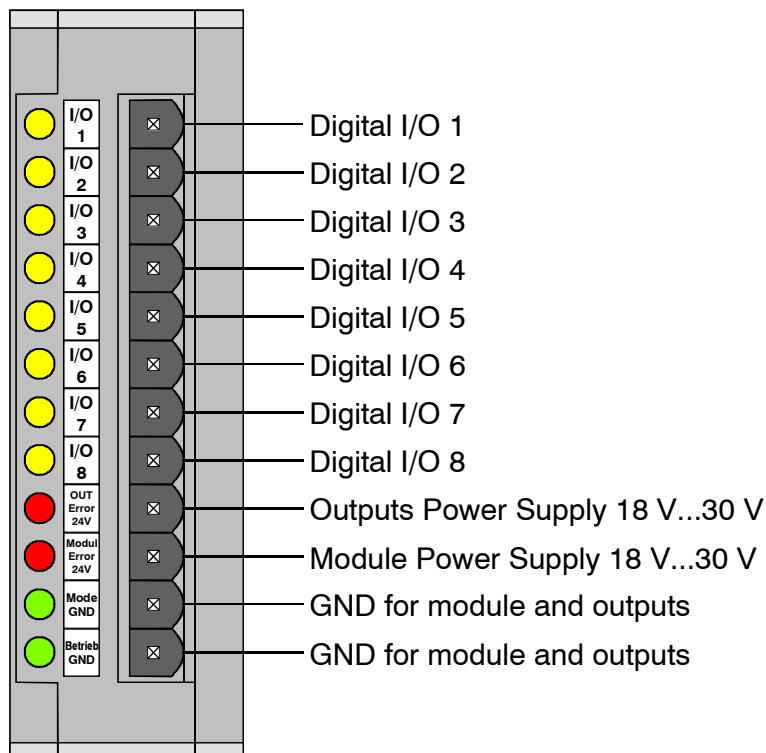
3.3.3 Anschluss des CAN-BUS

3-poliges COMBICON-Stecksystem (Stifte) der Fa. Phoenix ohne Kragen, Rastermaß 5,00 mm
Bezeichnung **MSTBA 2,5/3-G-5,00**

Belegung des Adapterkabels 3-pol. Combicon ↔ DSUB9:



3.3.4 Anschluss Versorgungsspannung und I/O-Anschlüsse



Kurzbeschreibung

3.4 Gehäusedesign

- Gehäusotyp: UEGM-MSTB der Fa. Phoenix Contact

3.5 Software

- CANopen mit speziellen Erweiterungen
- ein Empfangs- und ein Sende-PDO
- ein SDO (expedited protocol, max. 4 Nutzdatenbytes)
- Node- und Life Guarding, Emergency Messages
- Kopplung von bis zu vier Modulen zu einem 'logischen' Modul mit 32 E/A-Leitungen
- SYNC-Frame-Auswertung (mit Einschränkungen bei Modulkopplung)
- Betriebsart 'DS401' oder 'Servo' durch DIP-Schalter wählbar

3.6 Bestellhinweise

Typ	Eigenschaften	Bestell-Nr.
CAN-CBM-DIO8	CANbloc-Mini-Modul mit 8 digitalen Ein-/ Ausgänge, 24 V / 1 A, 24 VDC, CANopen, 0...50 °C Umgebungstemperatur	C.2830.02
CAN-CBM-DIO8-12V	wie C.2830.02, aber 12 VDC Spannungsversorgung	C.2830.04
CAN-CBM-DIO8-T	wie C.2830.02, aber -20...+70 °C Umgebungstemperatur	C.2830.05
CAN-CBM-DIO8-MD	Anwenderhandbuch in deutsch ^{1*)} (dieses Handbuch)	C.2830.20
CAN-CBM-DIO8-ENG	Engineering Manual in englisch ^{2*)} Inhalt: Schaltpläne, Bauteilpositionen, Datenblätter wichtiger Bauteile	C.2830.25

1*) Wird das Handbuch gemeinsam mit dem Produkt bestellt, so wird es kostenlos mitgeliefert.

2*) Für dieses Handbuch wird eine Schutzgebühr erhoben.

4. Digitale Ein/Ausgaben

Jeder digitale E/A-Anschluss wird durch den DIP-Schalter oder die Konfigurations-Codestelle entweder als Ein- oder als Ausgang definiert.

Der Zustand der E/A-Anschlüsse wird jeweils mit einer gelben LED angezeigt.

4.1 Meldung der digitalen Eingänge

Die Übertragungsarten für die digitalen Eingänge können unterschieden werden in:

- *zyklisch, aber nicht synchron*: Das Sende-Zeit-Intervall wird nur durch das E/A-Modul vorgegeben (im CANopen-Standard nicht vorgesehen, für einfache Anwendungen aber sinnvoll!)
- *azyklisch, synchron*: Die Sendung erfolgt nach dem Empfang einer SYNC-Message (PDO - Übertragungstyp 0)
- *zyklisch, synchron*: Die Sendung erfolgt, nachdem jeweils eine bestimmte Anzahl SYNC-Message empfangen wurden (PDO-Übertragungstyp 1...240)
- *ereignisgesteuert, asynchron*: Die Sendung erfolgt, wenn sich der Zustand bestimmter digitaler Eingänge ändert (PDO-Übertragungstyp 255)

Wenn bis zu vier E/A-Module zu einem 'logischen' Modul zusammengefasst sind, senden die E/A-Slave-Module ihre digitalen Eingangs-Informationen zunächst an den E/A-Master. Dieser fasst sie z.B. in einer 4-Byte-Message (PDO) für den Antriebsregler (bzw. für die übergeordnete Steuerung) zusammen und sendet sie anschließend über den Prozessdatenkanal.

4.2 Digitale Ausgänge

Jeder Ausgangstreiber kann maximal 1 A liefern (Nennstrom). Beim Überschreiten von ca. 4 A pro Ausgang wird der entsprechende Ausgang sofort abgeschaltet und ein Fehler angezeigt. Nach Werkseinstellung sind alle E/A-Anschlüsse nur als Eingang definiert.

Für das E/A-Modul Typ 'DIO8' wird empfohlen, die Definition von Ein- und Ausgängen immer in 'Zweiergruppen' zu definieren, wie auf der folgenden Seite beschrieben.

Um sie als Ausgang ansteuern zu können, muß erst durch den DIP-Schalter oder Parameterkanal definiert werden, welcher Anschluss auch ein Ausgang sein soll (siehe Kapitel Parametrierung, ab Seite 26).

4.2.1 Einschränkungen der Definition von E/A-Richtungen

Der Ausgangstreiber für die 8 digitalen Ausgänge verfügt nur über 4 Statusleitungen, d.h. der Fehlerstatus von je zwei Ausgangsleitungen kann nicht unterschieden werden. Aus technischen Gründen muß der Fehlerstatus von als INPUT definierten Anschlüssen ignoriert werden (open-load detection des Treibers).

Wenn z.B. I/O-Anschluss 1 als digitaler Eingang und I/O-Anschluss 2 als digitaler Ausgang verwendet werden, dann können eventuell auftretende Störungen (z.B. Überlastung) von I/O-Anschluss 2 nicht erkannt werden. Daher wird empfohlen, die E/A-Leitungen 1 und 2, 3 und 4, 5 und 6 sowie 7 und 8 immer als 'Zweiergruppen' mit gleicher Richtung zu betreiben.

Beispiel für eine **empfohlene** Kombination:

1+2 = Eingänge, 3+4=Ausgänge, 5+6=Eingänge, 7+8=Ausgänge

Beispiel für eine **nicht** empfehlenswerte Kombination:

1=Ausgang, 2=Eingang, 3+4=Ausgänge, 5+6=Eingänge, 7+8=Ausgänge

In diesem Beispiel wird z.B. eine Überlastung von I/O-Anschluss 1 nicht erkannt, weil I/O-Anschluss 2 als Eingang definiert ist und die interne Statusleitung von I/O 1+2 daher ignoriert werden muß. Fehler an den Ausgängen 3, 4, 7 und 8 werden allerdings weiterhin erkannt und angezeigt.

4.2.2 Rücklesen der digitalen Ausgänge

Jede als Ausgang konfigurierte E/A-Leitung behält grundsätzlich auch ihre Funktion als digitaler Eingang, d.h. wird ein Ausgang geschaltet, so wird der aktuelle Zustand wie in 4.1 beschrieben übertragen.

4.2.3 Empfangsüberwachung der digitalen Ausgänge (für ‘Servo’-Modus)

Die hier beschriebene Überwachung ist im Normalfall nur für den Einsatz im ‘Servo’-Modus sinnvoll (oder wenn aus anderen Gründen kein Node-Guarding nach CANopen möglich ist):

Bei periodischer Übertragung der Telegramme für die digitalen Ausgänge (‘RX-PDO’) kann das E/A-Modul den Empfangszyklus überwachen.

Bleiben diese Telegramme für eine bestimmte Zeit aus (z.B. für 500ms), so schalten die betroffenen Module alle Ausgänge auf den vordefinierten Fehlerzustand (z.B. alle Ausgänge aus) und melden einen Fehler mit der roten LED. Das Modul bleibt aber im normalen Betriebszustand. Sobald wieder ein Telegramm für die digitalen Ausgänge empfangen wird, verschwindet die Fehleranzeige wieder.

Die Default-Einstellung des entsprechenden Parameters (Codestelle 30, siehe Anhang) ist 0, d.h. bei Verwendung der Werkseinstellung findet keine Empfangsüberwachung statt.

4.2.4 Ausgangsstörungen

Bei Kurzschluss oder Überschreitung des maximal zulässigen Ausgangsstroms eines Kanals wird treiberintern nur der betroffene Ausgang abgeschaltet, die anderen Ausgänge werden davon zunächst nicht beeinflusst.

Der verwendete Ausgangstreiber signalisiert auch Fehler, wenn an seine Ausgänge statt einer nach Masse führenden Last eine externe Spannung angelegt wird. Da es sich hierbei um einen schwerwiegenden Fehler handelt, werden die Ausgänge des kompletten Moduls zurückgesetzt. Aus diesem Grund muß vermieden werden, an eine als Output definierte E/A-Leitung eine externe Spannung anzulegen!

Fehler an den Ausgängen werden mit der roten ‘Out Error’-LED angezeigt.

Außerdem wird ein Emergency-Telegramm nach dem CANopen-Standard gesendet.

4.2.5 Eingangs- und Ausgangsschaltungen

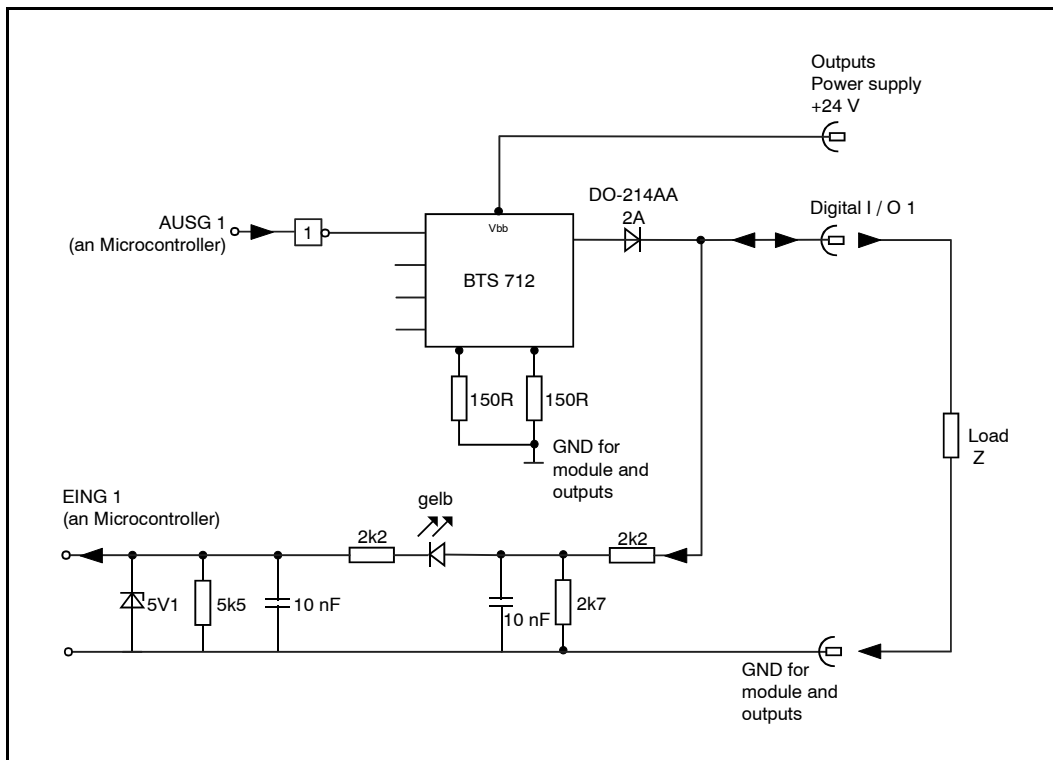


Abb.4.2.1: Prinzipschaltbild der Eingangs- und Ausgangsschaltungen

5. Installation

Generell sind bei der Verdrahtung sämtliche gültigen Richtlinien (DIN, VDE) bzgl. EMV-gerechtem Aufbau, Leitungsführung, Leiterquerschnitte, zu verwendende Materialien, Mindestabstände, Blitzschutz etc. zu beachten.

In diesem Abschnitt wird nur auf einige Besonderheiten genauer eingegangen.

5.1 Schutz des E/A-Moduls

Zu beachten:

Die E/A-Module enthalten elektrostatisch gefährdete Bauelemente!

- Vor Arbeiten im Bereich der Anschlüsse muß sich das personal von elektrostatischen Aufladungen befreien, Entladung z.B. durch Berührung von geerdeten Metallflächen oder PE-Schrauben.
- Beim Anschließen eines E/A-Moduls sollte erst die Masseverbindung hergestellt werden und danach die Verbindung der Signalleitungen, um eventuell vorhandene Potentialdifferenzen gefahrlos abzubauen.

5.2 CAN-Bus-Verdrahtung

Die CAN-Bus-Verkabelung erfolgt mit Leitungen nach ISO 11898. Die folgenden elektrischen Eigenschaften werden gefordert:

Parameter	Wert			Bemerkung
	min.	nom.	max.	
Impedanz (Z)	108 Ω	120 Ω	132 Ω	Wellenwiderstand (HF), gemessen zwischen zwei Signalleitungen
spez. Widerstand (r)		70 m Ω/m		
spez. Leitungsverzögerung		5 ns/m		muß zu den Verzögerungen der Sende- und Empfangsschaltung addiert werden

Beide Enden des CAN-Busses müssen impedanzrichtig, d.h. mit einem induktivitätsarmen 120 Ω - Widerstand abgeschlossen werden. Die E/A-Module der 'DIO8'er-Serie enthalten keinen Abschlusswiderstand. Die maximal mögliche Buslänge hängt von der Baudrate (Bit-Timing) ab. Dies wird im Kapitel 'CAN-Bus' näher erläutert.

6. Anzeigeelemente

Zur Anzeige werden 8 E/A-Zustandsanzeige-LED's, 2 Betriebsanzeige-LED's und 2 Fehleranzeige-LED's verwendet:

- Betriebs-Anzeige 'Betrieb GND' (grün)
- Mode-Anzeige 'Mode GND' (grün)

- Ausgangs-Fehleranzeige 'OUT Error 24V' (rot)
- Modul-Fehleranzeige 'Modul Error 24V' (rot)

Im Normalbetrieb leuchten nur grüne LED's. Sobald eine rote LED leuchtet, deutet dieses auf einen Fehler oder auf einen besonderen Anlagenzustand (z.B. Programmiermodus) hin.

Beim Auftreten von Fehlern **während der Initialisierungsphase** werden die Betriebs-, Mode- und Modulfehler-LED's 'parallel' zum Blinken gebracht, um diese Fehlerklasse klar von Fehlern die während des Betriebes auftreten zu unterscheiden.

Ursachen für diese Fehler sind z.B.:

- falsche Einstellung des DIP-Schalters, z.B. CAN-Adresse 0
- defektes EEPROM
- sonstige Fehler beim Initialisierungs-Selbsttest

Fehler, die **während des Normalbetriebes** auftreten, werden durch die roten Fehler-LED's angezeigt. Solange die grüne Betriebs-LED auch leuchtet, handelt es sich um einen nicht schwerwiegenden Fehler.

Zu dieser Fehlerklasse gehören z.B.:

- Ausbleiben der periodischen Messages vom Antriebsregler
- Zu hohe Fehlerrate auf dem CAN-Bus, z.B. durch fehlerhaften Busabschluss
- extreme Buslast, so dass Messages nicht abgesetzt werden können
- vorübergehende Überlastung einzelner Ausgänge

Bei schwerwiegenden Fehlern erlischt die Betriebs-LED. Dazu zählen z.B.:

- CAN-Bus-OFF
- Hardwarefehler

In diesen Fällen ist das Modul nicht mehr betriebsbereit, d.h. es akzeptiert keine Befehle mehr über den CAN-Bus. Es kann dann nur durch Aus- und Wiedereinschalten der Modulversorgung neu gestartet werden. Bleibt nach dem Wiedereinschalten die Betriebs-LED erloschen, handelt es sich um einen Schaden an der Hardware.

Die verschiedenen LED-Anzeige Kombinationen werden im folgenden Kapitel aufgelistet!

LED - Anzeige Kombinationen

Betriebs-Anzeige	Mode-Anzeige	Modul-Fehler	Ausgangs-Fehler	Funktion
ON	ON	OFF	OFF	Normalbetrieb
ON	OFF (kurzzeitig)	OFF	OFF	CAN-Message empfangen
BLINKEN	ON	OFF	X	Betriebzustand nicht 'operational' (siehe unten)
OFF	ON	BLINKEN	OFF	Programmier-Modus
X	X	OFF	BLINKEN	Fehler vom Ausgangstreiber
ON	ON	BLINKEN	X	CAN-Bus-Fehler, Protokollfehler etc. (siehe unten) aber Modul noch in Betrieb
OFF	ON	BLINKEN	X	CAN-Bus-OFF, Modul nicht mehr in Betrieb
BLINKEN	BLINKEN	BLINKEN	OFF	Konfigurations- oder Selbsttest-Fehler

Blinken der Betriebsanzeige-LED (grün):

Blink-Code	Bedeutung	Störungsbehebung/Hinweise
kurz-lang-lang-kurz	'preoperational', Modul als E/A-Master konfiguriert.	- Sicherstellen, dass CAN-Anschlüsse korrekt sind - 'Start Node' senden - 'Servo'-Modus: Antriebsregler muß als BUS-Master arbeiten
kurz-kurz-kurz	'preoperational', Modul als E/A-Slave konfiguriert.	siehe oben
gleichmäßiges, langsames Blinken	'preoperational', E/A-Master / Slave- Kaskadierung abgeschaltet.	siehe oben

Anzeigeelemente

Blinken der Modul-Fehler-LED (rot):

Blink-Code	Bedeutung	Störungsbehebung/Hinweise
lang - kurz-kurz	Timeout durch RX-Monitor	<ul style="list-style-type: none"> - CAN-Bus-Verbindung überprüfen - Empfangs-Überwachung des E/A-Moduls abschalten (Codennr. 30) - 'Servo'-Modus: periodisches Senden von CAN-OUT3 des Antriebsreglers aktivieren
lang-kurz	Fehler des analogen Eingangsmoduls (z.B. Übersteuerung)	<ul style="list-style-type: none"> - Klemmenverbindungen prüfen - Kontrollieren ob analoge Eingangssignale im erlaubten Bereich liegen
	Nodeguard - Fehler (Nodeguard-RTR's vom NMT-Master bleiben aus)	<ul style="list-style-type: none"> - CAN-Bus-Verbindung überprüfen - Lifetime-Faktor nach DS203-2 erhöhen (siehe Beschreibung 'Extended NMT Bootup')
lang-kurz-lang-kurz	CAN-Bus-Fehler oder CAN-Bus-OFF	<ul style="list-style-type: none"> - Baudrate bei allen Busteilnehmern korrekt? - CAN-Bus impedanzrichtig abgeschlossen? - CAN-Bus korrekt abgeschirmt?
kurz-kurz-kurz	Slave-Fehler (nur wenn 'interne' Master/Slave-Kaskadierung aktiv)	Das E/A-Master-Modul hat bei einem seiner E/A-Slave-Module einen Fehler festgestellt: CAN-Bus und Versorgungsleitungen überprüfen!
(sonstige Codes)	CAL-Error-Register nicht 0	CAL-Error-Register auslesen und weiter untersuchen (Objekt 1001H, siehe CANopen)!

Gemeinsames Blinken von 'Betrieb'-LED, 'Mode'-LED und 'Error'-LED:

Blink-Code	Bedeutung	Störungsbehebung/Hinweise
3 * Blinken	Selbsttest-Fehler Typ 3	<ul style="list-style-type: none"> - Spannungsversorgung überprüfen - evtl. auch Hardware-Defekt
4 * Blinken	EEPROM-Fehler	<ul style="list-style-type: none"> - siehe Seite 25 - evtl. auch Hardware-Defekt
5 * Blinken	Fehler bei DIP-Konfiguration	<ul style="list-style-type: none"> - siehe Seite 22 (z.B. CAN-Adresse > 0 einstellen)

7. CAN-Bus

Dieses Kapitel beschreibt physikalische Eigenschaften des CAN-Busses und die wichtigsten in den E/A-Modulen implementierten Protokolle, die zum Einsatz des Moduls notwendig sind.

Weitere Informationen über die höheren Protokollschichten sind den CAL/CANopen-Dokumentationen 'CiA Draft Standard 201...207', 'CiA Draft Standard 301' und 'CiA Draft Standard Proposal 401' zu entnehmen. In diesen Unterlagen werden das Kommunikationsprofil (301) und das Geräteprofil (401) beschrieben. Wie weit die dort beschriebenen 'Objekte' in der E/A-Modul-Familie 'DIO8x' implementiert sind, ist den Datenblättern der speziellen Module zu entnehmen. Die grundsätzlich implementierten Objekte sind ab Seite 31 aufgelistet.

Die für den Datenverkehr verwendeten CAN-Identifizierer sind im Anhang aufgeführt.

7.1 CAN-Bus-Timing

Das CAN-Bus-Timing ist nach den CiA-Empfehlungen eingestellt (gewisse Abweichungen sind hardwarebedingt). Der CAN-Controller wird je nach eingestellter CAN-Baudrate (vgl. Seite 23) nach der folgenden Tabelle programmiert, außerdem gelten je nach Baudrate die angegebenen maximalen Buslängen:

Baudrate	Tseg 1	Tseg 2	Tsync	max. Buslänge (Kabel: 5ns/m)
1000 kBit/s	700 ns	200 ns	100 ns	10 m
500 kBit/s	1600 ns	300 ns	100 ns	80 m
250 kBit/s	3400 ns	400 ns	200 ns	250 m
125 kBit/s	6.5 μ s	1 μ s	500 ns	500 m
100 kBit/s	8 μ s	1.5 μ s	500 ns	700 m
50 kBit/s	16 μ s	3 μ s	1 μ s	1000 m
20 kBit/s	40 μ s	7.5 μ s	2.5 μ s	1500 m
10 kBit/s	80 μ s	15 μ s	5.0 μ s	2000 m

Die Größen Tsync, Tseg1 und Tseg2 beziehen sich auf die CAN-Spezifikation des verwendeten CAN-Controllers, sie bezeichnen Zeitabschnitte innerhalb eines Bits aus dem CAN-Telegramm.

Dabei ist:

Tsync: Dauer des Synchronisations-Segments.

Tseg1: Die Summe aus 'propagation time segment' und 'phase buffer segment 1'. Tseg 1 ist die Zeit zwischen dem Sync-Segment und dem Abtastzeitpunkt eines Bits ('sample point').

Tseg2: Die 'restliche' Zeit nach dem Abtastzeitpunkt ('phase buffer segment 2') bis zum Anfang des nächsten Bits.

Achtung: Die maximal erreichbare Buslänge der hier beschriebenen Module CAN-CBM-DIO8, CAN-CBM-AI4, CAN-CBM-AO4, CAN-CBM-OP1 und CAN-CBM-DIO8-Counter ist kürzer als bei anderen esd-CAN-Bus-Baugruppen, da bei diesen beiden Modulen andere Optokoppler zum Einsatz kommen. Dies ist bei der Netzwerk-Planung und Installation zu berücksichtigen!

7.2 CAN-Bus-Kanäle

Für den Datenaustausch zwischen der Steuerung (z.B. Antriebsregler) und E/A-Baugruppen stehen die folgenden Kanäle zur Verfügung:

- ein Prozessdatenkanal (PDO) mit 8 bis 32 Bit, je nach Anzahl E/A-Slave-Modulen,
- ein Parameterkanal (SDO) zum Zugriff auf herstellerspezifische Codestellen.

Je nach Erfordernissen können bis zu vier E/A-Module mit je 8 E/A-Anschlüssen via CAN-Bus zu einem 'großen' Modul mit 32 E/A-Anschlüssen zusammengefasst werden. Derart zusammengefasste Module werden in diesem Handbuch als 'E/A-Master-Modul' und 'E/A-Slaves' bezeichnet. Das E/A-Master-Modul ist nicht mit einem CANopen-Master zu verwechseln!

Für eine derartige Kopplung werden die folgenden Kanäle verwendet:

- ein interner Parameterkanal vom E/A-Master-Modul zu dessen E/A-Slave-Modulen,
- maximal drei interne Parameter- und Prozessdatenkanäle vom E/A-Slave-Modul zum E/A-Master-Modul.

Vor dem Einsatz der internen Modul-Kopplung sollte geprüft werden, ob die dafür benötigten Identifier im CAN-Netzwerk wirklich zur Verfügung stehen (siehe Anhang, ab Seite 47).

7.3 Prozessdatenkanäle

Jedes als Master konfigurierte E/A-Modul ist durch einen Prozessdatenkanal (PDO = **Process Data Object**) mit der Steuerung bzw. dem Antriebsregler verbunden.

Jeder Prozessdatenkanal besteht aus Sicht des E/A-Moduls aus einem Empfangszweig (PDO-RX) und einem Sendezweig (PDO-TX).

Über den Prozessdatenkanal empfängt das E/A-Modul Telegramme zum Schalten seiner digitalen Ausgänge und sendet den Zustand seiner digitalen Eingänge.

Im 'DS401'-Modus kann der Inhalt der Prozessdatentelegramme flexibel eingestellt werden. Näheres dazu befindet sich im Kapitel 'PDO-Mapping' ab Seite 40.

Im 'Servo'-Modus wird das PDO-Mapping bei der Default-Einstellung so konfiguriert, dass die CAN-Telegramme den folgenden Aufbau haben:

Vom E/A-Modul im 'Servo'-Modus gesendetes Prozessdaten-Telegramm (z.B. Adr. 2):

Identifizier	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
768+2 = 770	Eingang 1...8	Eingang 9...16	Eingang 17...24	Eingang 25...32	nicht benutzt (0)	nicht benutzt (0)	nicht benutzt (0)	nicht benutzt (0)

Dabei sind Eingang 1...8 die E/A-Anschlüsse des Master-Moduls, Eingang 9...16 befinden sich an Slave Nr. 1, Eingang 17...24 stammen von Slave Nr. 2 und Eingang 25...32 von Slave Nr. 3.

Eingang 1 wird in das niederwertigste Bit von Daten 1 eingeblendet, Eingang 8 in das höchstwertigste Bit usw.

Sind einige Slaves nicht vorhanden, liefern die entsprechenden Eingänge den Wert 0.

Die Initiative zum Senden der digitalen Eingangsinformationen geht im 'Servo'-Modus vom E/A-Modul aus, es sind ereignisgesteuerte und/oder zeitgesteuerte Sendungen möglich (siehe Kapitel 'Parametrierung', ab Seite 26).

Vom E/A-Modul im 'Servo'-Modus empfangenes Prozessdaten-Telegramm (z.B. Adr. 2):

Identifizier	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
767+2 = 769	Ausgang 1...8	Ausgang 9...16	Ausgang 17...24	Ausgang 25...32	nicht benutzt (0)	nicht benutzt (0)	nicht benutzt (0)	nicht benutzt (0)

Der Aufbau dieses Telegramms entspricht dem Aufbau des gesendeten Telegramms.

Ein gesetztes Bit bewirkt, dass der entsprechende Ausgang eingeschaltet wird (wenn der Ausgang definiert ist).

Versuche, nicht als Ausgang definierte Anschlüsse einzuschalten werden von den E/A-Modulen ignoriert.

7.3.1 Parameterkanal

Jedes E/A-Master-Modul kann über einen Parameterkanal konfiguriert werden. Wichtige Parameter werden vom E/A-Master-Modul auch an die E/A-Slaves weitergegeben. Eine genaue Beschreibung des Parameterkanals folgt ab Seite 28.

Für Standard-Anwendungen ist die Verwendung des Parameterkanals nicht erforderlich, da die wichtigsten Parameter auch über DIP-Schalter eingestellt werden können.

7.3.2 Interne Master-Slave-Kommunikation

Für die Kommunikation zwischen E/A-Master-Modul und dessen E/A-Slave-Modulen werden weitere CAN-Identifizierer belegt, die dem Anhang zu entnehmen sind.

Für die CAN-Übertragung von den E/A-Slaves zum E/A-Master werden jeweils eigene Identifizierer für jeden Slave verwendet, weil auf diesem Weg auch 'schnelle' Prozessdaten übertragen werden müssen, die nicht kollidieren dürfen.

Der Aufbau der Telegramme für die Übertragung von den Slave-Modulen zum Master-Modul ist wie folgt:

Identifizierer	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
896 ... + x	Kennung (s. u.)	Index (low)	Index (high)	Sub- Index	Nutz- daten	Nutz- daten	Nutz- daten	Nutz- daten

Als Kennung werden die folgenden Codes verwendet (in Anlehnung an den SDO-Kanal, der allerdings nichts mit dem hier beschriebenen 'internen' Parameterkanal zu tun hat):

Kennung	Bedeutung
0x00	Telegramm mit Prozessdaten und Status
0x40	Lesebefehl
0x2B	Schreibbefehl mit Daten (maximal 32 Bit)
0x4A	Antwort auf Lesebefehl mit Nutzdaten (maximal 32 Bit)
0x60	Quittierung für Schreibbefehl
0x80	Quittierung: 'Fehler'

Für die CAN-Übertragung vom E/A-Master zu dessen E/A-Slaves wird nur ein Identifizierer verwendet. Die Slave-Nummer steht in diesem Fall in den CAN-Nutzdaten, um Identifizierer zu 'sparen'. Auf diesem Weg werden Parameter und Prozessdaten übertragen, die das E/A-Master-Modul nach der Analyse der PDO-Telegramme an seine E/A-Slaves weiterleitet.

Sollte die 'interne' Kommunikation von E/A-Master-Modul und E/A-Slave-Modul stören bzw. nicht benötigt werden, so kann sie auch durch eine spezielle Programmieroption mit dem DIP-Schalter abgeschaltet werden (siehe Seite 27). Dieses kann notwendig sein, wenn im CAN-Netzwerk keine Identifier für die interne Master/Slave-Kommunikation zur Verfügung stehen.

8. Parametrierung

Die Parametrierung des Moduls ist ohne externe Hilfsmittel möglich (Baudrate, CAN-Adressen). Lediglich für die Programmierung der Ein-/Ausgänge und weiterer interner Parameter ist der Anschluss der Stromversorgung notwendig.

8.1 Parametrierung mit dem DIP-Schalter

8.1.1 Belegung des DIP-Schalters (im Normalzustand):

Schalternummer	Funktion	Bemerkungen
1	CAN - Adresse (Bit 0)	OFF = 0, ON = 1
2	CAN - Adresse (Bit 1)	OFF = 0, ON = +2
3	CAN - Adresse (Bit 2)	OFF = 0, ON = +4
4	CAN - Adresse (Bit 3)	OFF = 0, ON = +8
5	CAN - Adresse (Bit 4)	OFF = 0, ON = +16
6	CAN - Adresse (Bit 5)	OFF = 0, ON = +32
7 (*)	Master/Slave-Nummer (L) ODER CAN - Adresse (Bit 6)	OFF = 0, ON = 1 OFF = 0, ON = +64
8 (**)	Master/Slave-Nummer (H) ODER Reserve	OFF = 0, ON = +2
9	CAN-Baudrate (Bit 0)	
10	CAN-Baudrate (Bit 1)	
11	CAN-Baudrate (Bit 2)	
12	Programmierschalter	Normalzustand 'OFF'

Die CAN-Adresse ergibt sich aus der Addition der Zweierpotenzen (siehe Spalte 'Bemerkungen').

(*) ... Die Belegung von Schalter 7 hängt davon ab, ob die E/A-Module mit 'interner Master/Slave-Kaskadierung' (siehe ab Seite 26) arbeiten sollen oder nicht. Die Einstellung ob mit oder ohne Kaskadierung gearbeitet wird, kann u.a. im Programmiermodus (siehe Seite 25) vorgenommen werden.
Bei 'interner Kaskadierung' sind daher nur 63 verschiedene Node-Identifizierer möglich, ohne 'interne Kaskadierung' dagegen 127.

(**) ... Wird ohne 'interne Master/Slave-Kaskadierung' gearbeitet, so hat Schalter 8 zur Zeit noch keine Funktion. Er sollte in diesem Fall auf 'OFF' gesetzt werden, um nicht mit eventuellen zukünftigen Erweiterungen zu kollidieren.

8.1.2 Einstellung der CAN-Baudrate

Schalter 9	Schalter 10	Schalter 11	CAN-Baudrate
OFF	OFF	OFF	1000 kBit/s
OFF	OFF	ON	500 kBit/s
OFF	ON	OFF	250 kBit/s
OFF	ON	ON	125 kBit/s
ON	OFF	OFF	100 kBit/s
ON	OFF	ON	50 kBit/s
ON	ON	OFF	20 kBit/s
ON	ON	ON	10 kBit/s

8.1.3 Testmodus (nur für den Hersteller)

Der Testmodus dient zum schnellen Test der E/A-Anschlüsse und zum Setzen der Werkseinstellung.

Um den Testmodus zu aktivieren, müssen vor Einschalten der Versorgungsspannung *alle* DIP-Schalter auf 'ON' geschaltet werden und dann (max. 2 Sekunden nach Power-On) einer der Schalter 8...12 wieder auf 'OFF' gestellt werden. Das E/A-Modul schaltet daraufhin in den Test-Modus um, der nur durch Abschalten der Versorgungsspannung wieder verlassen werden kann. Außerdem werden **alle EEPROM-Zellen auf die Default-Werte gesetzt**.

Im Testmodus wird mit den letzten beiden DIP-Schaltern definiert, welcher Test durchgeführt werden soll:

S11 ON, S12 OFF: 'Lauflicht'-Test aller LED's und Zurücklesen der digitalen Eingänge (direkt nach Test-Aktivierung). Bei Rücklese-Fehlern wird die rote Modul-Fehler-LED zusätzlich angesteuert.

S11 OFF, S12 OFF: reserviert

S11 ON, S12 ON: reserviert

S11 OFF, S12 ON: 'DIP'-Test: Schalter 1...10 werden über Software auf die Ausgänge 1...8 und die beiden Fehler-LED's gekoppelt.

Bisher implementierte Hersteller-Default-Einstellungen:

(während Power-ON alle DIP-Schalter ON, dann nach ca 2 Sekunden...)

- Schalter 12 OFF: 'Servo'-Modus, kein A/D-Wandler installiert
- Schalter 11 OFF: 'DS401', kein A/D-Wandler installiert
- Schalter 10 OFF: 'DS401', 1 Kanal A/D + D/A, Spannungskopplung (± 10 V)
- Schalter 9 OFF: 'DS401', 1 Kanal A/D + D/A, Stromkopplung (± 20 mA)
- Schalter 8 OFF: 'DS401', Eingang 1 und 2 als Counter-Inputs (C0256)

8.1.4 Programmiermodus

Das Modul befindet sich im ‘Normalbetrieb’, solange der Programmierschalter (S12) in der Stellung ‘OFF’ steht. Durch Umschalten dieses Schalters nach ‘ON’ wird das Modul unabhängig vom aktuellen Betriebsmodus in den Modus ‘Programmieren’ gebracht.

Ist der Programmierschalter schon während Power-ON auf ‘ON’ gestellt, so wird eine Werkseinstellung durchgeführt mit der alle eventuell gespeicherten ‘Sonderkonfigurationen’ überschrieben werden. Wird der Programmierschalter dagegen erst nach Power-ON auf ‘ON’ geschaltet, so bleiben die nur via SDO einstellbaren Parameter erhalten.

Der Programmiermodus wird folgendermaßen signalisiert:

- Erlöschen der grünen Betriebs-LED
- grüne Mode-LED permanent an
- rote Modul-Fehler-LED blinkt langsam, solange der Programmierschalter auf ‘ON’ gestellt ist

Im Programmiermodus sind alle Ausgänge abgeschaltet.

In dieser Phase können die Schalter S1 ... S9 entsprechend der folgenden Kapitel eingestellt werden. Jede Schalterbetätigung wird durch ein kurzes Aufblitzen der grünen Betriebs-LED bestätigt.

Erst beim Umschalten des Programmierschalters von ‘ON’ auf ‘OFF’ wird die eingestellte Konfiguration (Definition von Ein- und Ausgängen) in einen nichtflüchtigen Speicher übernommen.

Nach erfolgreicher Programmierung wird der Blink-Takt der roten Modul-Fehler-LED kürzer. Der Programmiermodus kann aus Sicherheitsgründen nur durch Abschalten der Versorgungsspannung verlassen werden.

Vor dem Wiedereinschalten der Versorgungsspannung müssen die DIP-Schalter für die Adresseinstellung und CAN-Baudrate (S1...S11) wieder auf die aktuellen Werte gesetzt werden!

Alle im Programmiermodus mit dem DIP-Schalter einstellbaren Parameter sind auch via SDO einstellbar (siehe Seite 28). Dazu muß allerdings ein geeignetes Konfigurationstool zur Verfügung stehen.

8.1.5 Belegung des DIP-Schalters im Programmiermodus

Hinweis: Die DIP-Schalterbelegung gilt *nicht* für das CBM-AO4-Modul und das CBM-AI4-Modul (siehe 'CBM-AI4 Handbuch-Ergänzung')!

Schalternummer	Funktion	Bemerkungen
1	Portrichtung I/O 1	OFF = Eingang ON = Ausgang
2	Portrichtung I/O 2	''
3	Portrichtung I/O 3	''
4	Portrichtung I/O 4	''
5	Portrichtung I/O 5	''
6	Portrichtung I/O 6	''
7	Portrichtung I/O 7	''
8	Portrichtung I/O 8	''
9	CAN-ID-Basis , Software-Modell	OFF = 'Servo'-Modus ON = 'DS401'-Modus
10	interne E/A-Kaskadierung (E/A-M/S-Verbund)	OFF = Kaskadierung ON = keine Kaskad.
11	'Reserve'	Sollte 'OFF' bleiben!
12	Programmierschalter	Normalzustand 'OFF'

8.1.5.1 Programmierung der Ein-/ Ausgabefunktionen

Im Programmiermodus haben die Schalter 1 bis 8 des DIP-Schalters die folgenden Sonderfunktionen: Der Schalterzustand 'ON' definiert die entsprechende E/A-Leitung als Ausgang, der Zustand 'OFF' definiert einen Eingang.

Beim E/A-Modul CAN-CBM-DIO8 sind die Hinweise in Kapitel 'Digitale Ausgänge', ab Seite 10 zu beachten!

8.1.5.2 Programmierung der CAN-Identifizierung und des Software-Modells

Im Programmiermodus hat Schalter 9 die folgende Bedeutung:

Schalter 9 'OFF': Es werden die im Anhang ab Seite 47 aufgeführten CAN-Identifizierung verwendet.

Schalter 9 'ON': Es werden *zunächst* CAN-Identifizierung aus dem 'Predefined Connection Set' verwendet ('DS 401'-Modus, siehe Anhang).

8.1.5.3 Unterdrücken des E/A-Master/Slave-Verbundes

Ist der Verbund von E/A-Master mit drei E/A-Slaves nicht erwünscht, so kann diese Funktionalität auch unterbunden werden. Dazu dient Schalter 10 im Programmiermodus:

Schalter 10 'OFF': Verbund von E/A-Master-Modul mit maximal drei Slave-Modulen möglich (dieses ist der Normalfall).

Schalter 10 'ON': Kein E/A-Master-Slave-Verbund möglich!
Sonderfall nur notwendig wenn der ab Seite 26 beschriebene 'interne' Verkehr zwischen E/A-Master und E/A-Slaves stört.

8.1.5.4 Beispiele für Programmierung mit dem DIP-Schalter

'Servo'-Modus mit allen I/O-Pins als Ausgängen :

- Spannungsversorgung aus
- DIP-Schalter 1 bis 8 'ON'
- DIP-Schalter 9 bis 11 'OFF'
- DIP-Schalter 12 'ON'
- Spannungsversorgung an: rote LED blinkt langsam
- DIP-Schalter 12 'OFF': rote LED 'blitzt' langsam (Programmierung fertig)
- Spannungsversorgung wieder aus
- Vor dem nächsten Einschalten:
Gerätenummer und Adressen wieder einstellen (siehe Seite 22, 23)!

'DS401'-Modus ohne interne E/A-Kaskadierung, alle I/O-Pins als Eingänge :

- Spannungsversorgung aus
- DIP-Schalter 1 bis 8 'OFF'
- DIP-Schalter 9 'ON'
- DIP-Schalter 10 'ON'
- DIP-Schalter 11 'OFF'
- DIP-Schalter 12 'ON'
- Spannungsversorgung an: rote LED blinkt langsam
- DIP-Schalter 12 'OFF': rote LED 'blitzt' langsam (Programmierung fertig)
- Spannungsversorgung wieder aus
- Vor dem nächsten Einschalten:
Gerätenummer und Adressen wieder einstellen (siehe Seite 22, 23)!

8.2 Parametrierung mit dem Parameterkanal (SDO)

Mit Hilfe eines PC's mit CAN-Interface und Konfigurationstool oder eines anderen Leitsystems können die wichtigsten internen Parameter des E/A-Moduls konfiguriert werden. Dazu steht in jedem als Master konfigurierten E/A-Modul ein SDO-Kanal zur Verfügung, der im Kapitel 'CANopen' genauer beschrieben wird.

8.2.1 Zugriff auf E/A-Codestellen über den Parameterkanal

Das E/A-Modul verwendet eigene 'Codestellen' zum Zugriff auf interne Parameter, die der Auflistung im Anhang zu entnehmen sind. Der Subindex wird innerhalb dieses 'herstellerspezifischen Profils' (CANopen-Bezeichnung) nicht verwendet, das entsprechende Telegrammfeld ist auf '0' zu setzen.

8.2.1.1 Zeitliches Verhalten bei SDO-Schreibzugriffen

Schreibzugriffe auf E/A-Module ohne 'interne Master/Slave-Kopplung' werden innerhalb von 30 ms bestätigt, wenn *mit* 'automatischer Speicherung' gearbeitet wird.

Wesentlich schneller erfolgt die Bestätigung eines 'normalen' Schreibzugriffs, wenn *ohne* automatische Speicherung gearbeitet wird. In diesem Fall erfolgt die Bestätigung eines Schreibzugriffes innerhalb von 10 ms. Sollen die Parameter auch nach Abschalten der Versorgungsspannung erhalten werden, müssen die Parameter durch einen speziellen 'save'-Befehl (an CANopen-Objekt 1 abgespeichert werden. Die Ausführungszeit des 'save'-Befehls ist dann von der Anzahl geänderter Speicherzellen im EEPROM abhängig. Im ungünstigsten Fall benötigt das verwendete EEPROM bis zu 20 ms pro Schreibzugriff, derart lange Wartezeiten konnten im Test allerdings nie beobachtet werden.

Besonders lange (bis zu 100 ms) kann ein SDO-Schreibzugriff dauern, wenn mit 'interner Master/Slave-Kopplung' gearbeitet wird und Codestellen beschrieben werden, die das E/A-Master-Modul erst an seine E/A-Slaves weitergeben muß. Die dabei intern ablaufende Prozedur wird im nächsten Kapitel beschrieben.

8.2.1.2 Interner Ablauf bei SDO-Schreibzugriffen

Das folgende Beispiel soll den ‘kompliziertesten’ internen Ablauf bei SDO-Schreibzugriffen mit ‘interner Master/Slave-Kopplung’ verdeutlichen:

Beispiel 1: Definition von 32 Ein/-Ausgängen über den Parameterkanal

Alle E/A-Leitungen des Master-Moduls und der drei daran angehängten Slave-Module sollen als Ausgänge definiert werden. Das Master-Modul hat die Geräteadresse 2.

Im Anhang befindet sich die dafür zuständige Codestelle 10_{dez} (Portrichtung). Daraus ergibt sich das folgende vom Leitsystem zu sendende CAN- (SDO-)Telegramm:

Identifizier	Befehls-code	Index (low)	Index (high)	Sub-index	Data 1	Data 2	Data 3	Data 4
1536 + 2 = 1538	(write) 0x23	0xFF - 10_{dez} = 0xF5	0x5F-0 = 0x5F	0x00	(Master) 0xFF	(Slave1) 0xFF	(Slave2) 0xFF	(Slave3) 0xFF

Dieses Telegramm wird nur vom E/A-Master-Modul mit der Geräteadresse 2 empfangen. Weil dieses Telegramm aber auch Daten für die an den Master angeschlossenen Slaves enthält, sendet der Master daraufhin an ‘seine’ Slaves entsprechende Telegramme über den internen Master=>Slave-Kanal (Identifizier 1664+2) und wartet die Antwort der Slaves ab.

Hinweis: Für den Fall, dass nicht alle drei Slaves vorhanden sind, bricht der Master das Warten nach einer bestimmten Zeit (50ms) ab, meldet aber keinen Fehler!

Die Slaves speichern ihre individuellen Port-Richtungsdefinitionen im nichtflüchtigen Speicher ab und bestätigen dieses nach erfolgtem Speichern über die Slave->Master-Kanäle. Anschließend speichert auch der Master seine eigene ‘neue’ Portrichtungsdefinition im nichtflüchtigen Speicher. Auch dieser Vorgang braucht eine gewisse Zeit (max. 20 ms). Erst nach Abschluss der Programmierung, meldet das Master-E/A-Modul die Bestätigung an das Leitsystem:

Identifizier	Befehls-code	Index (low)	Index (high)	Sub-index	Data 1	Data 2	Data 3	Data 4
1408 + 2 = 1410	(w.resp) 0x23	0xFF - 10_{dez} = 0xF5	0x5F-0 = 0x5F	0x00	0x00	0x00	0x00	0x00

Parametrierung

Die tatsächliche Reaktionsdauer kann nur abgeschätzt werden, weil sie z.B. von folgenden Faktoren abhängt:

- CAN-Bus-Baudrate,
- CAN-Bus-Last und Anteil höher priorisierter CAN-Telegramme und
- Belastung der CPU's in den E/A-Modulen durch andere Aufgaben, z.B. schnelle Zustandswechsel an den digitalen Eingängen.

8.2.2 Hinweise zur Verwendung des Parameterkanals

Generell ist bei der Verwendung der SDO's zu beachten:

- SDO's dienen zur Konfiguration der E/A-Module. Sie sind nicht dafür vorgesehen, im laufenden ('Normal'-)Betrieb ständig interne Parameter zu ändern. Dies würde auch auf Kosten der Lebensdauer der zur Speicherung verwendeten EEPROM's gehen!
- Beim Austausch eines über SDO konfigurierten Moduls bzw. Einsatz des Moduls an einem anderen Ort sollte sicherheitshalber die Konfiguration wieder auf die Werkseinstellung gesetzt werden. Dies gilt insbesondere für die oben demonstrierte Änderung der E/A-Richtungen.
- Bei Parametern, die über DIP-Schalter *und* über SDO konfiguriert werden können, gilt grundsätzlich die zuletzt durchgeführte Einstellung.
- Umstellungen von Identifier-'Basisadressen' durch Schreibzugriffe auf Codestellen ab 50 werden erst nach Neu-Initialisierung der Module z.B. durch Netzwerk-Management oder Abschalten der Versorgungsspannung wirksam. Die Möglichkeit des 'Umdefinierens' der CAN-ID-Basisadressen sollte nur verwendet werden, wenn die Einstellmöglichkeit mit Hilfe des DIP-Schalters nicht ausreicht!
- Der Zugriff auf die Codestellen ist nicht vom Zustand des Moduls (operational, preoperational) abhängig.
- Beim Zusammenschalten von mehreren Modulen zu einem 'logischen' Modul sollten die E/A-Slave-Module auch bei der Parametrierung des E/A-Master-Moduls angeschlossen sein, weil das E/A-Master-Modul einige Parameter bei SDO-Zugriffen über den internen Parameterkanal mit seinen E/A-Slave-Modulen austauscht (z.B. die Port-Richtungsdefinitionen).
- Nur nach dem Auftreten schwerwiegender Fehler ist kein Zugriff über SDO mehr möglich.
- Das automatische Speichern der Parameter im EEPROM bei jedem Schreibzugriff kann über Zugriff auf eine hersteller-spezifische Codestelle abgeschaltet werden. Dies kann in Objekt 1010h im CANopen-Kommunikationsprofil abgefragt werden. Nach der Werkseinstellung via DIP-Schalter ist das 'autonome' Abspeichern der Parameter allerdings aktiv, d.h. es braucht kein 'save'-Befehl gesendet zu werden.

9. CANopen

Dieses Kapitel enthält einige Grundlagen zum Thema ‘CANopen’ und die wichtigsten Informationen über die implementierten Funktionen.

Eine komplette CANopen-Beschreibung liegt außerhalb des Rahmens dieses Handbuchs. Der interessierte Leser sei auf die nicht ganz einfache CAL/CANopen-Dokumentation verwiesen!

9.1 Das CANopen-Objektverzeichnis

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

Der Index kann 16-Bit-Parameter nach der CanOpen-Spezifikation (CiA-Draft ‘DS301’, ‘DS401’) 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 (hex)	Objekt	Beispiel
0001 ... 009F	Definition von Datentypen	
1000 ... 1FFF	Communication Profile Area	1001 = Fehler-Register
2000 ... 5FFF	Manufacturer Specific Profile Area	5FF5 = Portrichtungen
6000 ... 9FFF	Standardised Device Profile Area	(hier : nach ‘DS401’) 6000 = Zustand von 8 Eingängen
A000 ... FFFF	reserviert	

9.1.1 Kommunikationsprofil

(Communication Profile Area)

Innerhalb dieses Abschnittes sind folgende Objekte implementiert:

Index (hex)	Name	Subindizes	Typ	Zugriff
1000	Device Type	-	Unsigned32	ro
1001	Error Register	-	Unsigned8	ro
1003	Predef'd Error... (Error History)	variabel	Unsigned32	r, w (nur Subindex 0)
1004	Number of PDO's	0, 1, 2	Unsigned32	ro
1005	COB-ID of Sync Message	-	Unsigned32	r, w (*)
1008	Manufacturer's Device Name	-	Visible String	ro
1009	Manufacturer's Hardware Version	-	Visible String	ro
100A	Manufacturer's Software Version	-	Visible String	ro
100B	Node-ID	-	Unsigned32	ro (=DIP-switch)
100C	Guard Time	-	Unsigned16	r, w (*)
100D	Life Time Factor	-	Unsigned8	r, w (*)
100	Node Guarding ID	-	Unsigned32	r, w (*)
1010	Store Parameters	0, 1	Unsigned32	r, w
1011 (in Vorb.)	Restore Default Parameters	0, 1	Unsigned32	r, w
1400	Receive PDO Communication Parameter	0, 1, 2	PDOCommPar	r, w
1600	Reveive PDO Mapping Parameter	0, 1...max. 8	PDOMapping	r, w
1800	Transmit PDO Communication Parameter	0, 1, 2	PDOCommPar	r, w
1A00	Transmit PDO Mapping Parameter	0, 1...max. 8	PDOMapping	r, w

(*)... Diese Parameter **sollten nur mit Hilfe des 'Extended NMT Bootup's' gesetzt werden!**

9.1.2 Geräteprofil

Standardised Device Profile Area

Die hier implementierten Objekte basieren auf dem ‘CiA Draft Standard Proposal 401’ Version 1.4. Einige Objekte, z.B. die 16- und 32-Bit-Zugriffs-Objekte sind nur verfügbar wenn mehrere CAN-CBM-DIO8-Module mit je 8 E/A-Anschlüssen zu einem ‘Verbund’ gekoppelt sind.

9.1.2.1 Digitale Ein- und Ausgänge

Index (hex)	Name	Subindexes	Typ	Zugriff
6000	Read State [of] 8 Input Lines	0, 1...max. 4	Unsigned 8	ro
6006	Input Interrupt Mask [for] 8 Input Lines, any change	0, 1...max. 4	Unsigned 8	r, w
6020	Read State [of] 1 Input Line	0, 1...max. 32	Unsigned 8, Boolean	ro
6100	Read State [of] 16 Input Lines	0, [1, 2]	Unsigned 16	ro
6120	Read State [of] 32 Input Lines	0, max. 1	Unsigned 32	ro
6200	Write State [for] 8 Output Lines	0, 1...max. 4	Unsigned 8	r, w
6206	Fault Mode [for] 8 Output Lines	0, 1...max. 4	Unsigned 8	r, w
6207	Fault State [for] 8 Output Lines	0, 1...max. 4	Unsigned 8	r, w
6220	Write State [for] 1 Output Line	0, 1... max. 32	Unsigned 8, Boolean	r, w
6250	Fault Mode [for] 1 Output Line	0, 1...max. 32	Unsigned 8, Boolean	r, w
6260	Fault State [for] 1 Output Line	0, 1...max. 32	Unsigned 8, Boolean	r, w
6300	Write State [for] 16 Output Lines	0, 1...max. 2	Unsigned 8, Unsigned 16	r, w
6306	Fault Mode [for] 16 Output Lines	0, 1...max. 2	Unsigned 8, Unsigned 16	r, w
6307	Fault State [for] 16 Output Lines	0, 1...max. 4	Unsigned 8, Unsigned 16	r, w
6320	Write State [for] 32 Output Lines	0, max. 1	Unsigned 8, Unsigned 32	r, w
6326	Fault Mode [for] 32 Output Lines	0, max. 1	Unsigned 8, Unsigned 32	r, w
6327	Fault State [for] 32 Output Lines	0, max. 1	Unsigned 8, Unsigned 32	r, w

9.1.2.2 Analoge Ein- und Ausgänge

Nur für CBM-AI4 und CBM-AO4!

Index (hex)	Name	Subindizes	Typ	Zugriff
6401	Reads value of the input channel (not converted)	0 1[...4]	Unsigned 8 Unsigned16	ro
6411	Writes value of the output channel (not converted)	0 1[...4]	Unsigned8 Unsigned 16	r, w
6420	Set Analogue Input Range	0 1[...4]	Unsigned8 Unsigned 16	r, w
6421	Determines which events cause an interrupt for a specific channel	0 1[...4]	Unsigned 8 Unsigned 8	r, w
6423	Globally enable/disable Interrupt	0	Boolean	r, w
6424	When enabled, interrupt triggered when analogue input rises above this value (not converted)	0 1[...4]	Unsigned8 Unsigned 32	r, w
6425	When enabled, interrupt triggered when analogue input falls below this value (not converted)	0 1[...4]	Unsigned8 Unsigned 32	r, w
6426	When enabled, interrupt triggered when analogue changes by more than this value from previous reading (rising or falling) (not converted)	0 1[...4]	Unsigned8, Unsigned 32	r, w
6443	Output Fault Mode	36525	Unsigned8	r, w
6444	Default Output Fault value (unconverted)	1	Unsigned 8 Unsigned 32	r, w

9.1.3 Hersteller-spezifisches Profil

Manufacturer Specific Profile Area

Wichtig ist der 'Manufacturer Specific Profile'-Bereich von 0x2000 bis 0x5FFF, in der spezielle Systemparameter gelesen und gesetzt werden können.

Im E/A-Modul CAN-CBM-DIO8 berechnet sich der Index nach der Formel:

$$\text{Index} = 0x5FFF - \text{Codenummer_des_E/A-Moduls}$$

Die Codenummern des E/A-Moduls sind im Anhang, ab Seite 47 aufgeführt.

9.2 Service-Daten-Objekte (SDO's)

Service-Daten-Objekte 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 beim E/A-Modul CAN-CBM-DIO8 im Zustand 'operational' und 'pre-operational' möglich. Die Hinweise auf Seite 30 sind zu beachten!

9.2.1 Aufbau der SDO-Telegramme

Identifizier	Befehls-code	Index (low)	Index (high)	Sub-index	Data 1	Data 2	Data 3	Data 4
--------------	--------------	-------------	--------------	-----------	--------	--------	--------	--------

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

0x40 = 64 dez: Read Request, d.h. ein Parameter soll gelesen werden,

0x23 = 35 dez: Write Request mit 32 Bit Daten, d.h. ein Parameter soll gesetzt werden.

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'.

Das E/A-Modul antwortet auf **jedes** empfangene Telegramm mit einem Antworttelegramm. Das Antworttelegramm kann folgende Befehlscodes enthalten:

0x42 = 66 dez: Read Response, dieses Telegramm enthält den gewünschten Parameter,

0x60 = 96 dez: Write Response, d.h. ein Parameter wurde erfolgreich gesetzt,

0x80 = 128 dez: Error Response, d.h. das E/A-Modul meldet einen Kommunikationsfehler.

Weitere Informationen sind den CiA-Spezifikationen zu entnehmen.

9.2.2 Nichtflüchtiges Speichern von Parametern

Generell können alle kommunikationsrelevanten Parameter (PDO mapping + Communication) im EEPROM gespeichert werden. Ebenso können die Objekte im Device Profile mit denen das Verhalten des Moduls im Fehlerfall eingestellt wird im EEPROM gespeichert werden.

Der Zustand der digitalen und analogen Ausgänge kann dagegen nicht gespeichert werden.

Durch die Default-Einstellung werden die Parameter bei jedem Schreibvorgang in einem nichtflüchtigen Speicher abgelegt, ohne daß dazu ein spezieller ‘save’-Befehl an das Objekt 1010h gesendet werden muß. Die Antwort auf ein SDO-Telegramm erfolgt dann erst nach erfolgter Aktualisierung des nichtflüchtigen Speichers.

Sollte dieses ‘automatische’ Abspeichern und die damit verbundene zeitliche Verzögerung stören, kann das ‘automatische’ Abspeichern durch Umprogrammieren der herstellereigenen Codestelle 17 auch abgeschaltet werden.

Der Inhalt von Objekt 1010h zeigt den aktuellen Speicher-Modus an:

- Sind Bit 1 und Bit 2 von Objekt 1010h, Subindex 1 gesetzt, so speichert das E/A-Modul die Konfiguration nach jeder Änderung via SDO-Kanal automatisch.
- Ist Bit 1 von Objekt 1010h, Subindex 1 nicht gesetzt, so speichert das E/A-Modul die Konfiguration nicht ‘automatisch’, sondern muß durch Schreiben der Zeichenkette ‘save’ (73h 61h 76h 65h, Reihenfolge aus CAN-Telegramm) in das Objekt 1010h, Subindex 1 extra dazu veranlaßt werden.

Anmerkung: Der CANopen-Standard sieht keine Möglichkeit zum ‘Umschalten’ des Speicher-Modus vor. Die Umschaltmöglichkeit mit Codestelle 17 wurde für eine größtmögliche Flexibilität der E/A-Module vorgesehen.

Wenn bei jedem Hochfahren eines Netzwerkes viele Schreibzugriffe über SDO durchgeführt werden, ist es ratsam, die ‘automatische’ Speicherung durch Null-Setzen von Codestelle 17 abzuschalten und das Aktualisieren des EEPROMs nach der Initialisierung ‘auf einmal’ durchzuführen. Dadurch wird dem E/A-Modul unter anderem das ständige Neu-Berechnen der EEPROM-Prüfsumme erspart, und die vielen SDO-Schreibzugriffe werden erheblich schneller bestätigt (siehe auch Seite 28).

Das verwendete EEPROM hat eine Lebensdauer von 1 Mio. Löschr/Schreib-Zyklen.

9.3 NMT-Boot-up

9.3.1 Einfaches Boot-Up

Die E/A-Module der 'DIO8'-er-Serie können mit dem im CiA-Draft Standard 301 in Kapitel 8.3 beschriebenen 'Minimum Capability Device'-Boot-up initialisiert werden. Im einfachsten Fall reicht nach dem Einschalten ein Telegramm zum Umschalten vom Zustand 'preoperational' in den Zustand 'operational'. Dazu reicht es im einfachsten Fall, an den CAN-Identifizier 0000h das 2-Byte-Telegramm '01 00' zu senden (=Start Remote Node).

9.3.2 Extended Boot-Up

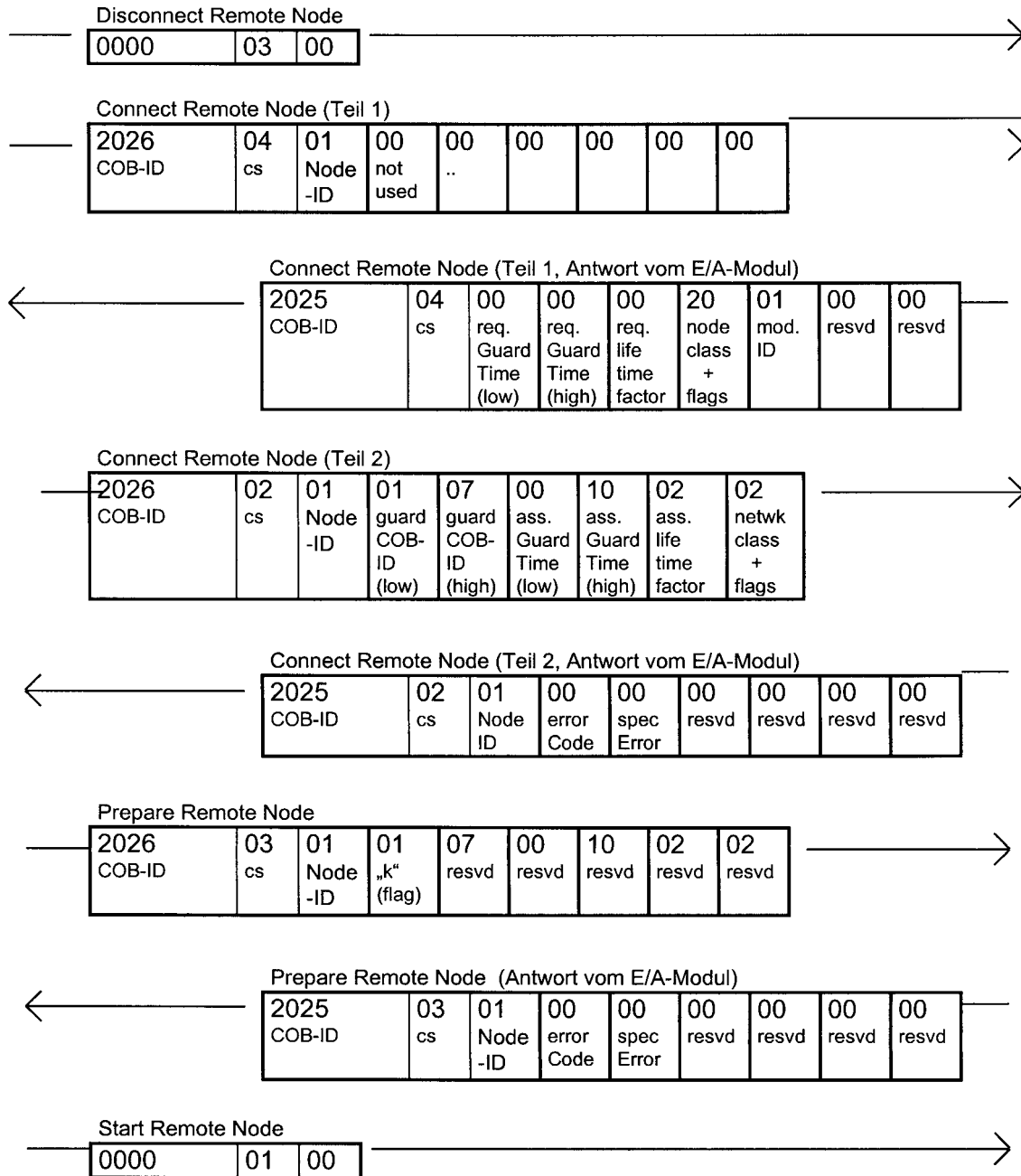
Zusätzlich ist auch das 'Extended Boot-up' mit Nodeguarding-Aktivierung möglich. Der NMT-Master kann dazu die auf der folgenden Seite skizzierte Sequenz an den NMT-Slave senden (hier: Modul-ID = 1, COB-IDs dezimal, CAN-Daten hexadezimal).

Die in den CAN-Telegrammen eingetragenen Parameter haben die folgende Bedeutung:

Node-ID:	'Knoten-Identifizier', beim E/A-Modul CAN-CBM-DIO8 ist das die über die DIP-Schalter eingestellte 'CAN-Adresse' im Bereich 1...63 (ohne 'E/A-Kaskadierung' 1...127).
req. Guard Time:	Vom Slave angeforderte Überwachungszeit in Millisekunden. Das E/A-Modul CAN-CBM-DIO8 fordert keine Überwachungszeit an, der Eintrag ist daher 0.
req. Life Time Factor:	Vom Slave angeforderter 'life time factor'. Wird durch den 'assigned life time factor' überschrieben.
node class+flags:	Gibt die Fähigkeiten des E/A-Moduls als NMT-Slave an. In Bit7 dieses Bytes könnte der NMT-Slave einen 'Konfigurations-Download' anfordern. Diese Möglichkeit besteht beim E/A-Modul CAN-CBM-DIO8 nicht, Bit 7 ist daher immer '0'.
mod. ID:	Beim E/A-Modul CAN-CBM-DIO8 identisch mit dem Node-ID.
guard COB-ID:	Der vom Master an den Slave zugewiesene CAN-Identifizier für das Nodeguarding-Protokoll. Der CAN-Identifizier muß zwischen 1761 und 2015 liegen.

NMT-Master

NMT-Slave



Nach dieser Prozedur befindet sich das E/A-Modul im Zustand ‘operational’ und ist bereit zum Senden und Empfangen von Prozessdaten-Objekten. Außerdem ist das Node-Guarding nun aktiv, so das der ‘Node-Guarding-Master’ von nun an periodisch über das RTR-Telegramm den Modulzustand anfordern **muß** (Näheres dazu im Kapitel ‘Node Guarding’).

9.4 Prozessdatenobjekte (PDO's)

Die PDO's dienen zur Übertragung von Prozessdaten.

Im 'Receive'-PDO werden i.A. die Daten für die digitalen Ausgänge vom Steuerungssystem (z.B. Umrichter) an die E/A-Module übertragen.

Im 'Transmit'-PDO sendet das E/A-Master-Modul (im CANopen-Netz immer SLAVE!) den Zustand seiner digitalen Eingänge an das Steuerungssystem (oder - in besonderen Fällen - an andere E/A-Geräte).

Der Aufbau eines PDO's wurde für ein einfaches Beispiel ('Servo'-Antriebsregler) bereits in Kapitel 'Prozessdatenkanäle' vorgestellt. Für den allgemeinen Anwendungsfall kann der Aufbau eines PDO-Telegramms aber (fast) beliebig zusammengestellt werden. Dazu dient das im folgenden Kapitel beschriebene 'PDO-Mapping'.

9.4.1 PDO-Mapping

Das variable PDO-Mapping dient zum ‘Einblenden’ bestimmter Objekte aus dem CANopen-Objekt-Verzeichnis in ein maximal 8 Byte langes Prozessdaten-Telegramm.

Dafür ist für den Receive-PDO (digitale Eingänge) das Objekt 1800h und für den Transmit-PDO (digitale Ausgänge) das Objekt 1A00h zuständig. Beide genannten Objekte haben den gleichen prinzipiellen Aufbau. Diese ‘PDO-Mapping Parameter’-Objekte entsprechen einer Liste, in der alle im PDO abgebildeten Objekte stehen.

Jeder Listeneintrag enthält den Index eines abgebildeten Objektes, den Subindex des abgebildeten Objekts und die Anzahl der von diesem Objekt im PDO eingeblendeten Datenbits.

Beispiel:

Im Transmit-PDO soll nur der Zustand der ersten 8 digitalen Eingänge eines E/A-Moduls abgebildet werden.

Das Objekt, welches die digitalen Eingänge in Gruppen von je 8 Eingängen (also ‘byte-weise’) enthält, ist das Objekt 6000h aus ‘DS 401’ (z.Z. nur ‘DSP 401 V1.4’, Kapitel 8.1.1).

Subindex 1 von Objekt 6000h enthält immer den Zustand der ersten 8 digitalen Eingänge, es könnte notfalls auch periodisch über SDO ausgelesen werden. Um dieses Objekt in den PDO ‘hineinzumappen’, müssen Index, Subindex und Bit-Anzahl wie folgt zu einem 32-Bit-Wert (‘unsigned 32’) zusammengefaßt werden:

MSB, Bits 31...24	Bits 23...16	Bits 15...8	LSB, Bits 7...0
Index (High)	Index (Low)	Subindex	Anzahl Datenbits
60	00	01	08

Bevor dieser 32-Bit-Wert über SDO in das Mapping-Objekt 1A00h geschrieben werden kann, muß allerdings die ‘Listenlänge’ des Mapping-Objektes mindestens auf ‘1’ gesetzt werden. Dies erfolgt durch Schreiben einer ‘1’ in das Mapping-Objekt 1A00h, Subindex 0.

Damit ist in der Liste Platz für genau einen Eintrag, der anschließend an Subindex 1 geschrieben wird. Bei dem Versuch auf ein Listenobjekt zuzugreifen welches nicht in die Liste paßt (wegen des Inhalts von Subindex 0), erfolgt ein ‘Error Response’ beim SDO-Zugriff auf das Mapping-Objekt.

Nach der erfolgreichen Programmierung des Mapping-Objektes hat dieses den folgenden Inhalt:

Objekt 1A00h, Subindex 0: **01h** (d.h. ein Eintrag in der Liste)

Objekt 1A00h, Subindex 1: **60000108h** (d.h. bilde 8 Bits von Objekt 6000, Subindex 1 im PDO ab)

Durch diese Programmierung wird das Prozessdaten-Telegramm nur ein Byte enthalten, in dem der aktuelle Zustand der ersten acht digitalen Eingänge steht.

Sollen in den noch ‘freien’ 7 Bytes des Prozessdaten-Telegramms weitere Informationen stehen, so ist zunächst Objekt 1A00h Subindex 0 zu erhöhen (z.B. auf 4). Anschließend können weitere Listeneinträge in Objekt 1A00 Subindizes 2,3,4 vorgenommen werden.

Bei den E/A-Modulen CAN-CBM-DIO8 können maximal acht verschiedene Objekte in einem PDO abgebildet werden. Die Anzahl der Datenbits muß aus Geschwindigkeitsgründen 8 oder ein ganzzahliges Vielfaches davon sein.

9.4.2 PDO-Übertragungsarten

9.4.2.1 Synchrone Übertragungsarten

Die synchronen Übertragungsarten sind verwendbar, wenn ein Teilnehmer im CANopen-Netzwerk die entsprechenden SYNC-Telegramm erzeugen kann. Das E/A-Modul CAN-CBM-DIO8 kann diese Telegramme nur empfangen, aber nicht selber erzeugen.

Die synchrone Übertragungsart wird durch den 'PDO transmission type' im Kommunikationsparameter des entsprechenden Prozessdatenobjekts definiert.

Ein 'transmission type' von 5 bedeutet z.B., daß nach jeweils fünf empfangenen SYNC-Messages ein Prozessdatenobjekt gesendet wird. Details sind dem CiA Draft Standard 301 zu entnehmen.

9.4.2.2 Ereignisgesteuerte Übertragung

Jeder Änderung eines digitalen Einganges an Master oder Slaves kann eine Meldung der digitalen Eingänge auslösen.

Nach Werkseinstellung erfolgt dies durch alle bestückten Eingänge, durch Ändern einer Ereignis-Maske kann dies geändert werden (siehe Anhang 'Code-Tabelle' oder 'DS401').

Wird diese Ereignis-Maske auf 0xFFFFFFFF gesetzt, lösen alle Eingänge eine Meldung aus. Enthält dies Maske 0x00000000, so findet keine ereignisgesteuerte Übertragung statt.

Um eine zu hohe Buslast durch ereignisgesteuerte Übertragung zu vermeiden, sind zwei Mechanismen implementiert:

- Entprellung der digitalen Eingänge (mit einstellbarer Entprellzeit),
- Definition einer 'Mindest-Pausenzeit' die zwischen dem Senden zweier ereignisgesteuerter(!) Prozessdaten-Telegramme vergehen muß.

Die **Default-Werte** sind:

- Entprellzeit 1 ms,
- keine Mindest-Pause zwischen zwei ereignisgesteuerten Sendungen.

9.4.3 Zyklische asynchrone Übertragung

Die Eingangsinformation kann zyklisch (z.B. alle 100 ms) mit dem Prozessdatenkanal übertragen werden. Die Sendezykluszeit kann über den Parameterkanal geändert oder auch abgeschaltet werden. Sie stellt eine **zusätzliche** Übertragungs-Initiative zu den im 'CANopen'-Standard definierten Möglichkeiten dar, die speziell für den Einsatz am 'Servo'-Antriebsregler vorgesehen ist. In der CANopen-Terminologie könnte man diese Übertragungsart als 'zyklisch aber nicht vom SYNC-Objekt abhängig' bezeichnen.

Default-Wert:

- 'Servo'-Modus: Zyklische Sendung alle 100 ms.
- 'DS401'-Modus: Zyklische Sendung abgeschaltet (0), Übertragung nur vom 'PDO-Transmission Type' abhängig.

Die Aktivierung dieser Übertragungsart ist durch Ändern von Codestelle 14 möglich (im herstellerspezifischen Objektbereich, siehe Anhang).

9.4.4 Node Guarding / Life Guarding

Zur gegenseitigen Überwachung von Steuerung und den angeschlossenen E/A-Modulen (insbesondere zum Erkennen von Verbindungs-Ausfällen) dient das Node/Life-Guarding.

In der CANopen-Terminologie ist das Node-Guarding die Überwachung der E/A-Module durch den NMT-Master, das Life-Guarding läuft dagegen im E/A-Modul ab und überwacht den NMT-Master.

Node-Guarding:

Der NMT-Master ruft mit einer speziellen RTR-Message (CAN-Request) zyklisch ein besonderes Telegramm von seinen NMT-Slaves (hier: E/A-Module) ab. Stimmt die Antwort der Slaves nicht mit der erwarteten Antwort überein, erkennt der NMT-Master einen Fehler und reagiert entsprechend. Im Antworttelegramm ist der Modulstatus und ein Togglebit enthalten. Identifier und Zykluszeit werden dem NMT-Slave beim Extended-NMT-Bootup mitgeteilt.

Life-Guarding:

Die NMT-Slaves überwachen, ob sie im Rahmen des Node-Guardings vom NMT-Master abgefragt werden. Bleiben diese Abfragen für eine gewisse Zeit aus, sendet der betroffene NMT-Slave eine Emergency Message. Beim E/A-Modul CAN-CBM-DIO8 wird das Ausbleiben der Node-Guard-Requests auch mit der roten Modul-Fehler-LED angezeigt (natürlich nur, wenn das Node-Guarding aktiviert wurde).

Die Zeit bis zum Auslösen des Fehlers ist das Produkt aus der Node-Guarding-Zyklus-Zeit und dem sogenannten **Life-Time-Faktor**, der dem NMT-Slave ebenfalls beim Extended-NMT-Bootup mitgeteilt wird.

9.4.5 Das Emergency-Telegramm

Beim Auftreten interner Fehler sendet das E/A-Modul ein 'Emergency Telegramm'. Dieses ist folgendermaßen aufgebaut:

Emergency Telegram

128+ADR COB-ID	00 error code (low)	23 error code (high)	00 error regis- ter	00 manu. spec. field	00 manu. spec. field	00 manu. spec. field	00 manu. spec. field	00 manu. spec. field
-------------------	------------------------------	-------------------------------	------------------------------	-------------------------------	-------------------------------	-------------------------------	-------------------------------	-------------------------------

Das E/A-Modul CAN-CBM-DIO8 unterstützt (mindestens) die folgenden (Emergency-) Error Codes:

Error Code (hex)	Bedeutung
00xx	Fehler verschwunden oder kein Fehler
10xx	allgemeiner Fehler
23xx	zu hoher Ausgangsstrom
50xx	Modul-Hardware - Fehler
60xx	Modul-Software - Fehler
70xx	Fehler in Zusatzmodulen
80xx	Überwachung
81xx	Kommunikationsfehler

Im Feld 'error register' des Emergency-Telegramms wird der aktuelle Inhalt von CANopen-Objekt 1001h eingeblendet.

Im E/A-Modul wird eine Fehler-Historie über die gesendeten Emergency-Telegramme gespeichert. Dazu dient Objekt 1003h im CANopen-Objekt-Verzeichnis.

10. Schnellstart

10.1 Konfiguration für den Einsatz im CANopen-Netzwerk

Für einen schnellen Start mit der einfachsten Konfiguration sind die folgenden Schritte notwendig:

1. Definition der Ausgänge des E/A-Moduls über DIP-Schalter

- Stromversorgung des E/A-Moduls einschalten.
- Schalter 12 ON. Rote Modul-Fehler-LED blinkt langsam (Programmiermodus).
- Schalter 11 OFF (zukünftige Erweiterungen passiv).
- Schalter 10 ON (E/A-Kaskadierung passiv).
- Schalter 9 ON ('DS401'-Modus).
- Schalter 1...8 entsprechend E/A-Richtung: OFF = Eingang, ON = Ausgang.
- Schalter 12 OFF. Die Blink-Impulse der roten Modul-Fehler-LED werden kürzer.
- Modulversorgung abschalten. Damit ist die Definition der Ein-/Ausgänge abgeschlossen.

2. Einstellen von CAN-Adresse (= 'Node ID') und CAN-Baudrate am E/A-Modul

(z.B.: 125 kBit/sec, CAN-Adresse 2)

- Schalter 1 OFF, Schalter 2 ON.
- Schalter 3 bis 6 OFF (CAN-Adresse, vgl. Seite 22, 23).
- Schalter 7 bis 9 OFF.
- Schalter 10 und 11 ON (125 kBit/s, vgl. Seite 23).
- CAN-Verbindung zur Steuerung herstellen
- Modulversorgung einschalten.
Mode-Anzeige (grün) dauernd an, Betriebs-Anzeige (grün) blinkt bis 'Start Node' empfangen wurde.

10.2 Konfiguration für den Einsatz im CANopen-Netzwerk

Für einen schnellen Start mit der einfachsten Konfiguration sind die folgenden Schritte notwendig:

1. Definition der Ausgänge des E/A-Moduls über DIP-Schalter

- Stromversorgung des E/A-Moduls einschalten.
- Schalter 12 ON. Rote Modul-Fehler-LED blinkt langsam (Programmiermodus).
- Schalter 11 OFF (zukünftige Erweiterungen passiv).
- Schalter 10 OFF (E/A-Kaskadierung aktiv).
- Schalter 9 OFF ('Servo'-Modus).
- Schalter 1...8 entsprechend E/A-Richtung: OFF = Eingang, ON = Ausgang.
- Schalter 12 OFF. Die Blink-Impulse der roten Modul-Fehler-LED werden kürzer.
- Modulversorgung abschalten. Damit ist die Definition der Ein-/Ausgänge abgeschlossen.

2. Einstellen von CAN-Adresse, Master/Slave-Nummer und CAN-Baudrate am E/A-Modul

(z.B. 500 kBit/s, Antriebsregler auf CAN-Adresse 1, E/A-Master-Modul auf CAN-Adresse 2, Standard-Einstellungen für 'Servo'-Antriebsregler 93xx)

- Schalter 1 OFF, Schalter 2 ON.
- Schalter 3 bis 6 OFF (CAN-Adresse, vgl. Seite 22, 23).
- Schalter 7 OFF und 8 OFF (Modul wird E/A-Master).
- Schalter 9 und 10 OFF.
- Schalter 11 ON (500 kBit/s, vgl. Seite 23).
- CAN-Verbindung zum Antriebsregler herstellen.
- Modulversorgung einschalten.
Mode-Anzeige (grün) dauernd an, Betriebs-Anzeige (grün) blinkt.

3. Einstellungen am Antriebsregler

(siehe 'Konfiguration', 'Systembus' im System-Handbuch des 'Servo'-Antriebsreglers)

- Kontrollieren, ob die Baudrate des Antriebsreglers mit der Baudrate aller E/A-Module übereinstimmt (C0351).
- Sicherstellen, daß der Antriebsregler mit CAN-IN3 und CAN-OUT3 entsprechend der im Anhang definierten CAN-Identifizier sendet bzw. empfängt.
- Sendezyklus für CAN-OUT3-Telegramm auf 100 ms setzen, damit die Überwachung im E/A-Modul keine Fehler meldet.
- Antriebsregler für CAN-BUS-Masterbetrieb konfigurieren (C0352).

10.3 Tabelle der wichtigsten Identifier und Nachrichten für CANopen

Identifier [HEX]	Bezeichnung	Länge	Daten [HEX]	Bemerkungen
\$0	NMT	2	01 00	Start an alle (preoperational => operational)
\$0	NMT	2	80 00	operational => preoperational
\$80	NMT	0	-	Sync an alle
\$80 + NodeNo	SDO	0...8 Bytes	Nutzdaten	Emergency Message vom CBM
\$600 + NodeNo	SDO	0...8 Bytes	Nutzdaten	zum CAN-CBM-DIO8 (Rx)
\$580 + NodeNo	SDO	0...8 Bytes	Nutzdaten	vom CAN-CBM-DIO8 (Tx)
\$200 + NodeNo	PDO	0...8 Bytes	Nutzdaten	zum CAN-CBM-DIO8 (Rx)
\$180 + NodeNo	PDO	0...8 Bytes	Nutzdaten	vom CAN-CBM-DIO8 (Tx)

Begriffserläuterung:

NodeNo. ...	1...127 (dezimal)	
Emergency-Id...	Emergency Data Object	Notfalldaten
NMT...	Network Management (Master)	Netzwerkmanagement
PDO...	Process Data Objects	Prozessdaten (Ist-/Sollwerte)
Rx...	receive	empfangen
SDO...	Service Data Object	Service-Daten (Parameter)
Sync...	Sync(frame) Telegramm	Synchronisations-Identifier
Tx...	transmit	senden

11. Anhang

11.1 Verwendete CAN-Identifizier

(‘Adr’ = mit dem DIP-Schalter eingestellte Geräteadresse im Bereich 1...63 bzw. 1...127)

‘DS401’-Modus mit interner E/A-Master / Slave - Kaskadierung

(predefined connection set nach DS301 V3.0 Kapitel 8.4.1)

Funktion	Identifizier	Identifizier max	Anzahl
Netzwerkmanagement (einf., NMT-Master → NMT-Slave)	0	0	1
Netzwerkmanagement (erw., NMT-Slave → NMT-Master)	2025	2025	1
Emergency-Message	128 + Adr	191	63
PDO (transmit - z.B. digitale Eingänge 1...32) vom E/A-Master-Modul zur Steuerung	384 + Adr	447	63
PDO (receive - z.B. digitale Ausgänge 1...32) von der Steuerung zu den E/A-Modulen	512 + Adr	575	63
interne Messages von Slave1 zum E/A-Master	896 + Adr	959	63
interne Messages von Slave 2 zum E/A-Master	960 + Adr	1023	63
interne Messages von Slave 3 zum E/A-Master	1024 + Adr	1087	63
interner Parameterkanal vom E/A-Master-Modul zu seinen Slaves 1...3	1664 + Adr	1727	63
SDO (tx) vom E/A-Master-Modul zum Leitsystem	1408 + Adr	1471	63
SDO (rx) vom Leitsystem zum E/A-Master-Modul	1536 + Adr	1599	63
Nodeguarding (Identifizier-Vergabe nur durch NMT-Bootup)	1792 + xxx	2015	255

‘DS401’-Modus ohne interne E/A-Master / Slave - Kaskadierung (Werkseinstellung)
(predefined connection set nach DS301 V3.0 Kapitel 8.4.1)

Funktion	Identifizier	Identifizier max	Anzahl
Netzwerkmanagement (einf., NMT-Master → NMT-Slave)	0	0	1
Netzwerkmanagement (erw., NMT-Slave → NMT-Master)	2025	2025	1
Emergency-Message	128 + Adr	255	127(*)
PDO (tx) (z.B. digitale Eingänge 1...8)	384 + Adr	511	127
PDO (rx) (z.B. digitale Ausgänge 1...8)	512 + Adr	639	127
SDO (tx) vom E/A-Master-Modul zum Leitsystem	1408 + Adr	1535	127
SDO (rx) vom Leitsystem zum E/A-Master-Modul	1536 + Adr	1663	127
Nodeguarding (Identifizier-Vergabe nur durch NMT-Bootup)	1792 + xxx	2015	255

(*)... 127 Node-ID's sind nur verfügbar, wenn auf die 'interne Master/Slave-Kaskadierung' (d.h. das Zusammenschalten von bis zu vier E/A-Modulen zu einem 'logischen' Modul) verzichtet wird.

‘Servo’-Modus

Funktion	Identifizier	Identifizier max	Anzahl
Netzwerkmanagement	0	0	1
E/A-Eingänge (Eingang 1...32) vom E/A-Master-Modul zum Antriebsregler	767 + Adr	831	63
E/A-Ausgänge (Ausgang 1...32) vom Antriebsregler zu den E/A-Modulen	768 + Adr	830	63
interne Messages von Slave1 zum E/A-Master	896 + Adr	959	63
interne Messages von Slave 2 zum E/A-Master	960 + Adr	1023	63
interne Messages von Slave 3 zum E/A-Master	1024 + Adr	1087	63
interner Parameterkanal vom E/A-Master-Modul zu seinen Slaves 1...3	1664 + Adr	1727	63
SDO (tx) vom E/A-Master-Modul zum Leitsystem	1408 + Adr	1471	63
SDO (rx) vom Leitsystem zum E/A-Master-Modul	1536 + Adr	1599	63

11.2 Code-Tabellen

Codestellen für die elementare Modul-Konfiguration

Code-Nr. [DEZ]	CANopen-Index [HEX]	Inhalt	Datentyp, [Einheit]	Zugriff r = read w = write	Einstell-/Anzeigemöglichkeiten	Werkeinstellung	Bemerkungen, Verweise
5	5FFA	Bootup-Zeit	16 Bit, [ms]	r, w	0...65000 ms, 0xFFFF = keine Boot-Message	0	Zeitintervall zwischen Power-ON und Senden der Boot-Message
10	5FF5	Portrichtungs- Definitionen	32 Bit	r, w (*)	Bitvektor: 0 = Eingang, 1 = Ausgang	00000000b	LSB = Master MSB = Slave3
11	5FF4	Entprellzeit für digitale Eingänge	8 Bit, [ms]	r, w	0 = aus ... 255 ms	1 ms	
12	5FF3	Ereignis-Maske für digitale Eingänge	32 Bit	r, w	Bitvektor: 0 = kein IRQ 1 = Interrupt	1111...1111bin	ähnlich CANopen- Objekt 6006h
13	5FF2	Filterkonstanten der analogen Eingänge	16 Bit	r, w	siehe 'Handbuch- Ergänzung' für analoge Eingänge	0 = keine Filterung	
14	5FF1	Sende-Zyklus der digitalen und analogen Eingänge	16 Bit, [ms]	r, w	0 = nur durch Ereignis ... 65000 ms	100 ms	unabhängig von der Sendepause
15	5FF0	Mindest- Sende-Pause bei ereignisgesteu- ertem Senden	8 Bit, [ms]	r, w	0 = keine Pause ... 255 ms	0	
16	5FEF	internen Master- Slave-Verbund ermöglichen / verhindern	8 Bit	r, w	0 = keine Kaskadierung 1 = Kaskadierung möglich	1	
17	5FEE	'autonomes' Konfigurations- Speichern aktivieren	8 Bit	r, w	0 = Speichern nur auf Befehl 1 = Speichern nach jedem Schreiben	1	siehe CANopen- Objekt 1010h
18	5FED	Hardware-Modell	8 Bit	r...			n. f. Hersteller
19	5FEC	Modell-Optionen	8 Bit	r...			n. f. Hersteller

(*)... das Beschreiben dieser Codestelle wirkt sich erst nach Neustart aus.

Codestellen zur Definition von Überwachungs- und Anzeige-Zeiten des Moduls, die auf den Normalbetrieb keinen Einfluß haben

Code-Nr. [DEZ]	CANopen-Index [HEX]	Inhalt	Datentyp [Einheit]	Zugriff r = read w = write	Einstell-/Anzeigemöglichkeiten	Werkeinstellung	Bemerkungen, Verweise
30	5FE1	PDO-RX-Monitor (Überwachungszeit, wenn kein Life-Guarding)	16 Bit, [ms]	r, w	0= keine Überwachung ... 25500 ms	0	Auflösung 100 ms

Codestellen zur Definition von CAN-Identifizier-Basisadressen, zu denen noch die am DIP-Schalter eingestellte Geräteadresse (Bereich 1...63) addiert werden muß

Code-Nr. [DEZ]	CANopen-Index [HEX]	Inhalt	Datentyp [Einheit]	Zugriff r = read w = write	Einstell-/Anzeigemöglichkeiten	Werkeinstellung	Bemerkungen, Verweise
50	5FCD	PDO-RX	16 Bit	r, w (*)	siehe CANopen	je nach Modus ('Servo', 'DS401')	(dig. Outputs)
51	5FCC	PDO-TX	16 Bit	r, w (*)	siehe CANopen	je nach Modus	(dig. Inputs)
52	5FCB	Slave1	16 Bit	r	-	896	
53	5FCA	Slave2	16 Bit	r	-	960	
54	5FC9	Slave3	16 Bit	r	-	1024	
55	5FC8	Master->Slaves	16 Bit	r	-	1664	

Codestellen für Anzeige- und Diagnosezwecke

Code-Nr. [DEZ]	CANopen-Index [HEX]	Inhalt	Datentyp [Einheit]	Zugriff r = read w = write	Einstell-/Anzeigemöglichkeiten	Werkeinstellung	Bemerkungen, Verweise
93	5FA2	Drive Ident	16 Bit-Zahl	r	z.B. CAN-CBM-DIO8		
99	5F9C	S/W Version	String[4]	r	z.B. '0.01'		
101	5F9A	interne Fehler- und Zustandscodes	32 Bit	r	(1) EAM_error_flags (2) EAM_slave_errors (3) IOP_status (4) CAL_comm_pha.	-	interne Fehler und Zustände, nur für interne Verwendung

Anhang

Code-Nr. [DEZ]	CANopen-Index [HEX]	Inhalt	Datentyp [Einheit]	Zugriff r = read w = write	Einstell-/Anzeigemöglichkeiten	Werkeinstellung	Bemerkungen, Verweise
102	5F99	Angeschlossene Slaves	16 Bit	r	Bitvektor: 0 = kein Slave erkannt +1 = Slave 1 ok, +2 = Slave 2 ok, +4 = Slave 3 ok	-	Der Test wird erst beim Lesen dieser Codestelle durchgeführt
103	5F98	Hauptschleifengeschwindigkeit	16 Bit	r	liefert Schleifen/sec z.B. 3000	-	abhängig von CPU-Last
120	5F87	A/D-Wandler-Kanal setzen	8 Bit	w	0 ...7 (CPU-interne Multiplexer-Nummer)		
121	5F86	A/D-Wandler lesen	10 Bit	r	0 = 0 V ... 307 = 1.5 V... 511 = 2.5 V... 1023 = 5.0 V	-	NUR zum Test der internen 'digitalen' Eingangspegel
123	5F84	CAN-Register-Zugriff	8 Bit	r	liest den Inhalt der 256 CAN-Controller-Register		NUR für Diagnosezwecke