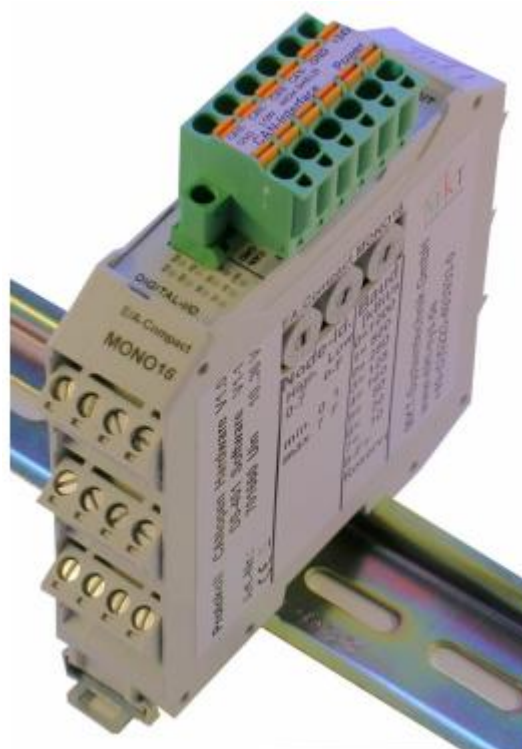


Softwarebeschreibung E/A-Modul-Familie
EA7-COMPACT-xxx
EA7-COMPACT MONOxx

Version 1.0



VORLÄUFIG

(preliminary - sorry, there is no English translation of this document yet)

Dokument-Nr: 85068 (de)

Original: C:\proj\ea_arm7\DOKU\art85068_SW_EA7_MONOxx_V10_DE.odt

Inhaltsverzeichnis

1. VERSIONSTAND DIESES DOKUMENTS.....	4
2. VORWORT.....	4
3. TECHNISCHE DATEN.....	5
4. KONFIGURATION PER CODIERSCHALTER.....	5
5. ANZEIGEELEMENTE.....	8
6. CANOPEN.....	9
7. DIGITALE EIN/AUSGÄNGE.....	14
8. TESTFUNKTIONEN.....	16

Der Inhalt dieser Dokumentation wurde mit größter Sorgfalt erarbeitet und geprüft. **MKT Systeme GmbH** -im Folgenden "MKT" genannt- übernimmt jedoch keine Verantwortung für Schäden, die aus Fehlern in der Dokumentation resultieren könnten. Insbesondere Beschreibungen und technische Daten sind keinerlei zugesicherten Eigenschaften im rechtlichen Sinne.

MKT 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.

Die Rechte an der Dokumentation liegen bei **MKT**. Weitergabe an Dritte und Vervielfältigung jeder Art sind nur mit schriftlicher Genehmigung durch **MKT** gestattet.

MKT Modulare Komponenten Technik für Systeme GmbH

Haßkampstraße 75-77

32257 Bünde

Tel.: 05223 / 49 39 33 - 0

FAX : 05223 / 49 39 33 - 20

E-Mail: info@mkt-sys.de

Internet: www.mkt-sys.de

1. Versionstand dieses Dokuments

Versionsnummer	Datum (ISO8601)	Autor	Hinweise, Änderungen
V1.0	2006-08-28	W.Büscher	Erstellung für "EA7-COMPACT-MONO16 V1.0"
V1.01	2006-08-30	W.Büscher	CAN-Baudrate 800kBit/sec entfernt
V1.02	2006-11-23	W.Büscher	Abweichungen vom CANopen-Standard erläutert

2. Vorwort

Dieses Dokument beschreibt die Softwarefunktionalität der E/A-Modul-Familie "E/A-Compact", z.B. dem „EA7-COMPACT-MONO16“ (mit insgesamt 16 Ein- oder Ausgängen). Der endgültige Produktname stand bei der Erstellung dieses Dokumentes noch nicht fest.

Die *Hardwarebeschreibung* zu diesen Geräten finden Sie unter der Artikelnummer 853xx .

Zur Kommunikation auf der Feldebene dient der CAN-Bus.

Das Kommunikationsprotokoll ist kompatibel mit CANopen DS301 V4.0 .

Das Geräteprofil entspricht weitgehend CANopen DS401, mit einigen herstellerspezifischen Erweiterungen (z.B. zur Definition der Klemmen als Ein- oder Ausgänge).

Eigenschaften der Firmware :

- CANopen mit speziellen Erweiterungen (siehe Kapitel 6.2.1)
- CANopen-Protokollstack der Firma SYS TEC electronic GmbH
- drei Empfangs- und drei Sende-PDOs (Prozeßdatenobjekte)
- variables PDO-Mapping, Defaulteinstellung siehe Kapitel 6.2 .
- Synchroner oder asynchroner PDO-Übertragung, periodisch oder ereignisgesteuert nach DS301 bzw DS401
- ein SDO (Servicedatenobjekt) für die Parametrierung per CAN
- Heartbeat, Emergency Messages
- per CAN parametrierbare SYNC-Sende-Funktion

Weitere Informationen :

- Anschlußbelegung, technische Daten, elektrische Spezifikation entnehmen Sie bitte der [Hardwarebeschreibung](#). Erhältlich bei www.mkt-sys.de .
- Eine von CANopen-Konfigurationstools einlesbare elektronische Gerätebeschreibung (EDS-File) finden Sie ebenfalls auf der Webseite von MKT Systemtechnik.
- Ein einfaches Testprogramm für dieses E/A-Modul befindet sich im Installationsarchiv von MKT's CAN-Tester für Windows.
- Details zu den CANopen-Übertragungsprotokollen finden Sie bei CiA (CAN in Automation), <http://www.can-cia.org/canopen/>

3. Technische Daten

... sind der Hardware-Dokumentation zu entnehmen (in Vorbereitung :-)

4. Konfiguration per Codierschalter

Zur Einstellung der CAN-Baudrate und des CANopen-Node-Identifiers („Knotennummer“) dienen bei diesem Modul Codierschalter (keine "Mäuseklaviere" !). Jeder Codierschalter hat 16 Stufen, die als 4-Bit-Binärzahl angesehen werden können. In hexadezimaler Schreibweise sind mit jedem Schalter die Zahlen 0 bis F einstellbar (dezimal 0 bis 15).

Die beiden linken Schalter (S301 und S302) bilden zusammen die (CANopen-)Knotennummer, der rechte Schalter (S303) dient zum Einstellen der CAN-Baudrate. Für CANopen-Profis: Die Einstellung dieser Parameter per LSS ist nicht möglich !

4.1 CAN-Baudrate

Die CAN-Baudrate wird mit S303 eingestellt (Codierschalter auf der rechten Seite) :

S303 (hex)	CAN-Baudrate [kbit/s]
0	1 Mbit / s
1	500 kbit / s
2	250 kbit / s
3	125 kbit / s
4	100 kbit / s
5	50 kbit / s
6	20 kbit / s (erfordert eine modifizierte Hardware ¹)
7	10 kbit / s (erfordert eine modifizierte Hardware)
8	reserviert
9	reserviert
A	reserviert
B	reserviert
C	reserviert
D	reserviert
E	reserviert
F	Umschaltung in den Programmiermodus (siehe Kapitel 7.1)

¹ Der standardmäßig verwendete *CAN-Transceiver* verfügt über eine Timeout-Überwachung (250 µs für den dominanten Zustand). Dies soll eine dauerhafte Blockade des CAN-Netzwerks bei Ausfall des *CAN-Controllers* im E/A-Modul verhindern. In einem CAN-Telegramm können 6 dominante Bits aufeinander folgen, darum erlaubt dieser CAN-Transceiver keinen Betrieb unter 50 kbit / s . Auf Wunsch kann das Gerät aber mit einem Transceiver ohne Timeout-Überwachung ausgerüstet werden.

4.2 CANopen-Node-Identifizier

Der CANopen-Node-Identifizier ("Knotennummer") wird mit S301 (höherwertige Ziffer, links) und S302 (niederwertige Ziffer) eingestellt.

Hinweis: Für den "predefined connection set" von CANopen sind nur die Knotennummern 1 bis 127 zulässig (hexadezimal 00 bis 7F) !

Knotennummer Null ist bei CANopen grundsätzlich unzulässig, und wird beim Hersteller möglicherweise zum Umschalten in den Testmodus verwendet, oder zum Aktivieren einer Sonderbetriebsart "ohne CANopen".

4.3 Definition der Ein/Ausgaberichtungen beim Modul „MONO16“

Wenn möglich, sollten Sie diese Einstellungen mit einem geeigneten CANopen-Konfigurationstool vornehmen (CANopen-Objekt 0x5FF5) !

Die in diesem Kapitel beschriebene Einstellmöglichkeit per Codierschalter ist sehr unkomfortabel und nur für den Notfall gedacht, wenn vor Ort etwas „auf die Schnelle“ umkonfiguriert werden muss.

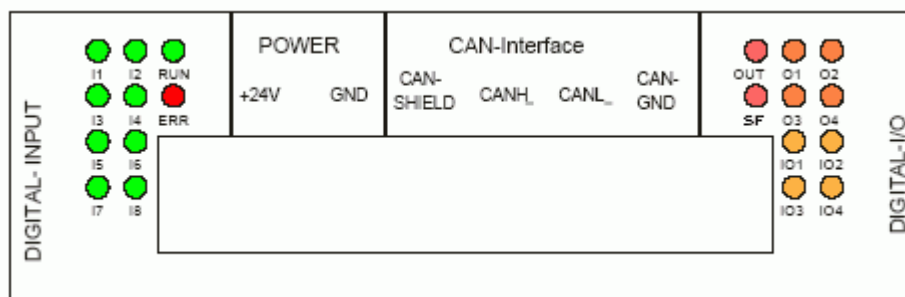
Um die Portrichtungen (digitaler Ein- oder Ausgang) per Codierschalter einzustellen, verfahren Sie folgendermaßen:

- 1) Schalten Sie die Spannungsversorgung des Moduls **aus**
- 2) Stellen Sie den *Codierschalter für die CAN-Baudrate* auf "F" (dezimal 15). Dadurch wird nach dem Einschalten des Moduls der ‚Programmiermodus‘ aktiviert.
- 3) Schalten Sie die Spannungsversorgung **ein**, und warten Sie bis beide CANopen-LEDs im gleichen Takt blinken
- 4) Stellen Sie mit den *Codierschaltern für die Knotennummer* die Portrichtungen ein. Die beiden Codierschalter für die Knotennummer bilden zusammen eine 8-Bit-Zahl (0x00 bis 0xFF). Jedes gesetzte Bit in dieser Zahl definiert die entsprechende Klemme als Ausgang, jedes gelöschte Bit entspricht einem Eingang. Der Wert 0x0F würde z.B. die Klemmen "IO1" bis "IO4" als Ausgang definieren.
Bei manchen Modulen (z.B. "EA7-Compact Mono16") blinken zur Kontrolle die gelben LEDs aller als Ausgang definierten I/O-Klemmen. Selbstverständlich werden *die Ausgänge selbst* in diesem Zustand nicht angesteuert.
- 5) Drehen Sie den *Codierschalter für die Baudrate* von "F" weiter auf "0" ¹. Dadurch wird der im vorhergehenden Schritt eingestellte 8-Bit-Wert in das Port-Richtungs-Register für die ersten 8 umschaltbaren digitalen I/O-Klemmen übernommen und dauerhaft im EEPROM abgespeichert. Hinweis: Beim „EA7-Compact Mono16“ sind nur 4 I/O-Anschlüsse frei definierbar; bei anderen (zukünftigen) E/A-Modulen entnehmen Sie die Angabe, welche Klemmen als Ein/Ausgang konfigurierbar sind, der Hardwareokumentation.
- 6) Anschliessend die Spannungsversorgung **wieder ausschalten**, und die CAN-Baudrate und den CANopen-Node-ID gemäß Kapitel 4.1 und 4.2 einstellen. Beim nächsten Einschalten (im "Normalbetrieb") sind die neuen Portrichtungen dann aktiv.

¹ Weichen Sie bitte nicht von dieser Sequenz ab. Wird z.B. (statt "F"->"0") von "F" auf "E" geschaltet, startet der LED-Test, statt die Portrichtungen umzuprogrammieren.

5. Anzeigeelemente

Die meisten E/A-Module dieser Familie verfügen über mehrere Anzeige-LEDs, die sich links und rechts neben dem Busanschluss befinden:



Im Normalbetrieb (d.h. nicht im Programmiermodus, s. Kap. 4.3) haben die LEDs folgende Bedeutung:

I1 - I8 (grün) : Digitale Eingänge

Diese LEDs zeigen den Zustand der digitalen Eingänge an den gleichnamigen Klemmen an. Hinweis: Diese LEDs werden von der CPU angesteuert ! Im Gegensatz zu älteren Modulen bleiben sie daher dunkel, wenn eine Versorgungsspannung fehlt oder die CPU nicht läuft.

RUN (grün) und ERR (rot) : CANopen-Indikatoren "Betrieb" und "Fehler"

Diese LEDs signalisieren den CANopen-Modulzustand, wie im CiA Draft Recommendation Proposal DRP303-3 beschrieben:

RUN: Blinken = Pre-Operational, Dauernd an = Operational

ERR: Aus = kein Fehler, Blinken = Fehlercode nach DRP303 .

OUT (rot): Ausgangsfehler

signalisiert Fehler vom High-Side-Treiber für die digitalen Ausgänge:

Aus = kein Fehler, Dauernd an = aktueller Fehler, kurzes Aufblitzen = alter Fehler (seit Power-On)

SF (rot): Sonderfunktionsanzeige

z.B. Übersteuerung der analogen Eingänge (falls vorhanden). "To Be Defined".

O1-O4 (gelb): Outputs 1-4

aktueller Zustand der digitalen Ausgänge

IO1-IO4 (gelb): Bidirektionale Inputs/Outputs

Diese zeigen den Zustand der bidirektionalen digitalen Ein/Ausgänge :

LED aus (dunkel) = Low-Pegel am Eingang und Ausgang

schwach leuchtend = High-Pegel am Eingang, Ausgang passiv

hell leuchtend = High-Pegel am Ausgang (Modul steuert die Klemme an)

6. CANopen

6.1 Objektverzeichnis

Das Objektverzeichnis (abgekürzt OD = Object Dictionary) entspricht weitgehend dem CANopen-Geräteprofil für ein Ein/Ausgabemodul nach DS401. Darin sind verschiedene standardisierte Objekte enthalten, mit denen die digitalen und analogen Ein- und Ausgänge abgefragt bzw. angesteuert werden. Die Einträge im Objektverzeichnis werden üblicherweise durch einen hexadezimal notierten Index und Subindex beschrieben, z.B. 0x6000.01 = Objekt 6000h Subindex 01h = digitale Eingänge 1 bis 8.

Der Aufbau des Objektverzeichnisses ist einer EDS-Datei beschrieben, die von einem CANopen-Konfigurationstool (kein MKT-Produkt) eingelesen werden kann.

Zu den im E/A-Modul implementierten Objekten nach CANopen DS 401 zählen:

- 1000-1FFF CANopen-Kommunikationsprofil. Siehe CiA DS 301 .
- 6000 Digitale Eingänge in 8-Bit-Gruppen. Siehe CiA DS401 .
- 6005-6006 Parametrierung des ereignisgesteuerten Sendens der digitalen Eingänge
- 6200 Digitale Ausgänge in 8-Bit-Gruppen
- 2000-5FFF Herstellerspezifischer Teil des Objektverzeichnisses.
Hier existieren weitere Objekte zur Konfiguration des Gerätes, z.B. zur Auswahl welche Klemme als Ein- und welche als Ausgang dienen soll. Näheres dazu im nächsten Kapitel.

6.1.1 Herstellerspezifischer Teil im CANopen-OD

Im herstellerspezifischen Teil finden Sie Funktionen, für die im standardisierten Teil des OD's keine entsprechenden Objekte definiert sind.

Index (hex)	Object type	Data type	Name	Remarks
5C00	Var	Unsigned16	Bitmaske zur Erkennung installierter Erweiterungen: Bit 0 = Frequenzzähler vorhanden, Bit 1 = Frequenzausgang , ...	hier vermutlich: 0 = keine Erweiterungen vorhanden !
5FF5.01	Var	Unsigned8	Input/Output-Richtungen für digitale I/O-Klemmen (für die ersten 8 digitalen Klemmen, falls diese bidirektional sind)	Ein bit pro Port: 0=input, 1=output
5FF5.02	Var	Unsigned8	Input/Output-Richtungen für digitale I/O-Klemmen (für die nächsten 8 digitalen Klemmen, falls diese bidirektional sind)	Ein bit pro Port: 0=input, 1=output

6.2 Prozeßdatenobjekte (PDO)

Die Prozeßdatenobjekte dienen zur schnellen Übertragung von Prozeßdaten per CAN-Bus. Das Gerät verfügt über drei Empfangs- und drei Sende-Prozessdatenobjekte, deren Inhalt und Übertragungsart individuell konfiguriert werden kann. Sie benötigen dazu ein CANopen-Konfigurationstool, welches per Parameterkanal (SDO, d.h. per CAN) auf das E/A-Modul zugreifen kann. Falls kein CANopen-Konfigurationstool zur Verfügung steht, müssen die folgenden Default-Einstellungen verwendet werden, die sich weitgehend am CANopen-Standard DS401 orientieren :

1400 Kommunikationsparameter für 1. Empfangs-PDO :

CAN-Identifizier: 0x200 + Knotennummer¹
Übertragungsart: 0xFE
Inhibit-Time : 0 (keine)

1401 Kommunikationsparameter für 2. Empfangs-PDO :

CAN-Identifizier: 0x300 + Knotennummer
Übertragungsart: 0xFE
Inhibit-Time : 0 (keine)

1402 Kommunikationsparameter für 3. Empfangs-PDO :

CAN-Identifizier: 0x400 + Knotennummer
Übertragungsart: 0xFE
Inhibit-Time : 0 (keine)

1600 Mapping-Parameter („Aufbau des Datenfeldes“) für 1. Empfangs-PDO :

Erstes Datenbyte : Digitale Ausgänge 1..8 (Objekt 6200.01, 8 bit)
Zweites Datenbyte : Digitale Ausgänge 9..16 sofern vorhanden

1601 Mapping-Parameter für 2. Empfangs-PDO :

1. und 2. Datenbyte : Analog-Ausgang 1, falls vorhanden (Objekt 6411.01, 16 bit)
3. und 4. Datenbyte : Analog-Ausgang 2, falls vorhanden (Objekt 6411.02, 16 bit)

1602 Mapping-Parameter für 3. Empfangs-PDO :

Per Defaulteinstellung nicht belegt (keine Objekte im PDO gemappt)

1600 Mapping-Parameter („Aufbau des Datenfeldes“) für 1. Empfangs-PDO :

Erstes Datenbyte : Digitale Ausgänge 1..8 (Objekt 6200.01)
Zweites Datenbyte : Digitale Ausgänge 9..16 sofern vorhanden

1800 Kommunikationsparameter für 1. Sende-PDO :

CAN-Identifizier: 0x180 + Knotennummer
Übertragungsart: 0xFF, d.h. asynchron wie im Geräteprofil (DS401) beschrieben
Inhibit-Time : T.B.D. ! (es gilt der Wert im EDS-File)
Event-Timer : 0 (aus, d.h. kein überlagerter periodischer Sendezyklus)

1801 Kommunikationsparameter für 2. Sende-PDO :

CAN-Identifizier: 0x280 + Knotennummer
Übertragungsart: 0xFF, d.h. asynchron wie im Geräteprofil (DS401) beschrieben

¹ Siehe Hinweis zur automatischen, Knotennummer-abhängigen Umschaltung der PDO-CAN-Identifizier in Kapitel 6.2.1 .

Inhibit-Time : T.B.D. ! (es gilt der Wert im EDS-File)
Event-Timer : 0 (aus, d.h. kein überlagerter periodischer Sendezyklus)

1802 Kommunikationsparameter für 3. Sende-PDO :

CAN-Identifizier: 0x380 + Knotennummer
Übertragungsart: 0xFF, d.h. asynchron wie im Geräteprofil (DS401) beschrieben
Inhibit-Time : T.B.D. ! (es gilt der Wert im EDS-File)
Event-Timer : 0 (aus, d.h. kein überlagerter periodischer Sendezyklus)

1A00 Mapping-Parameter für 1. Sende-PDO : Digitale Eingänge

Erstes Datenbyte : Digitale Eingänge 1..8 (Objekt 6000.01, 8 bit)
Zweites Datenbyte : Digitale Eingänge 9..16 sofern vorhanden

1A01 Mapping-Parameter für 2. Sende-PDO : Analoge Eingänge (falls vorhanden)

1. und 2. Datenbyte : Analoger Eingang 1 (Objekt 6401.01, 16 bit)
3. und 4. Datenbyte : Analoger Eingang 2 (Objekt 6401.02, 16 bit)

6.2.1 Knotennummer-abhängige, automatische Umschaltung der PDO-CAN-Identifizier

Um Anwendern, die nicht über ein CANopen-Konfigurationstool verfügen, die Inbetriebnahme mehrerer Module in einem CANopen-Netzwerk zu erleichtern, wurde die folgende, nicht ganz CANopen-konforme Erweiterung implementiert. Aus weiter unten erläuterten Gründen dürfte diese Erweiterung keine Probleme mit CANopen-Konfigurationstools verursachen.

Beim vorliegenden Gerät werden alle PDO-Kommunikations- und -Mapping-Parameter nichtflüchtig in einem EEPROM abgespeichert. Dazu zählen auch die CAN-Identifizier („COB-Identifizier“) der Prozessdatenkanäle. In manchen Anwendungen stört die Tatsache, daß die einmal per SDO eingestellten PDO-CAN-Identifizier nicht mehr „automatisch“ mit dem Umschalten der Knotennummer per DIP-Schalter geändert werden (dies würde bedeuten, daß der Inhalt des Objektverzeichnisses beim Ändern der Knotennummer des Moduls automatisch wieder auf die Defaultwerte gesetzt würde - eine unakzeptable Funktion).

Im CANopen-Standard DS301 sind die folgenden Default-Identifizier für die Prozessdatenobjekte definiert (als Teil des sogenannten „predefined connection sets“):

RPDO1: CAN-ID 200h + < Knotennummer des Moduls>
RPDO2: CAN-ID 300h + < Knotennummer des Moduls>
RPDO3: CAN-ID 400h + < Knotennummer des Moduls>
TPDO1: CAN-ID 180h + < Knotennummer des Moduls>
TPDO2: CAN-ID 280h + < Knotennummer des Moduls>
TPDO3: CAN-ID 380h + < Knotennummer des Moduls>

Modulintern werden die PDO-CAN-ID's als 32-Bit-Werte in Objekt 140X bzw 180X unter Subindex 1 abgespeichert, und zwar komplett (d.h. inklusive der niederwertigsten Bits, die beim „predefined connection set“ der Knotennummer 1..127₁₀ entsprechen). Wenn nach der Parametrierung des Moduls die Knotennummer (per DIP-Schalter) umgeschaltet wird, ändern sich die CAN-Identifizier der PDO-Kanäle daher nicht mehr. Dies ist in einigen Anwendungen erwünscht, in anderen nicht.

Um diese Problematik zu umgehen und beiden Anwendungen gerecht zu werden, wurde der folgende Ansatz gewählt:

Normalerweise wird beim Schreibzugriff auf den PDO-Kommunikationsparameter, Subindex 1, der „komplette“ CAN-Identifizier programmiert, so wie er später verwendet werden soll. Dies ist vollkommen CANopen-konform.

Alternativ kann beim Schreibzugriff auf den Kommunikationsparameter, Subindex 1, auch „nur“ Basis-Identifizier (z.B. 200h, 300h) ohne Knotennummer programmiert werden. Beim Lesezugriff addiert die Firmware dann automatisch die Knotennummer, wenn in Subindex 1 nur der Basis-Identifizier (z.B. 200h, 300h..) programmiert wurde. Beim Lesezugriff erscheint daher immer der tatsächlich verwendete CAN-Identifizier.

Bei der Auslieferung des Gerätes sind unter Subindex 1 in allen PDO-Kommunikationsparametern tatsächlich nur die „Basis-Identifizier“ abgelegt, d.h. ohne Knotennummer. Die Knotennummer wird daher beim Einschalten abhängig vom DIP-Schalter addiert. Dies geht auch aus dem EDS-File hervor: Dort steht z.B. unter Objekt 1400, Subindex 1, der Eintrag 0x200 (nicht 0x201).

6.2.2 Besonderheiten bei der PDO-Konfiguration per SDO

Da sich bei der Inbetriebnahme der E/A-Baugruppe mit einem bestimmten CANopen-Master (in einer SPS) verschiedene Probleme zeigten, musste die CANopen-Implementierung im Slave (E/A-Modul) leicht modifiziert werden. Daraus ergaben sich die folgenden *Abweichungen vom "strikten" CANopen-Standard*:

6.2.2.1 Abweichung vom CANopen-Standard im PDO Communication Parameter

Nach CANopen-Spezifikation DS301 V4.02 darf der CAN-Identifizierer eines PDOs nur geändert werden, wenn Bit 31 im entsprechenden Objekt (z.B. 1400.01) gesetzt, und der PDO damit "ungültig" ist. Zitat aus DS301 V4.02 (Beschreibung der Objekte 1600h - 17FFh = Receive PDO Mapping Parameter) :

- >
- > It is not allowed to change bit 0-29 while the PDO exists (Bit 31=0).
- >

Im E/A-Modul kann der CAN-Identifizierer nun (seit 2006-11-22) auch bei gelöschtem Bit 31 überschrieben werden. Strenggenommen ist das E/A-Modul damit nicht mehr CANopen-konform (nach DS 301 V4.02), aber ohne diese Modifikation konnte das Modul nicht mit dem SPS-eigenen CANopen-Master konfiguriert werden.

6.2.2.2 Abweichung vom CANopen-Standard im PDO Mapping Parameter

Nach CANopen-Spezifikation DS301 V4.02 darf auch das 'Mapping' eines PDOs (z.B. Objekt 1600) nur dann geändert werden, wenn Bit 31 im PDO-COB-ID (z.B. Objekt 1400.01) gesetzt, und der PDO damit "ungültig" ist. Zitat aus DS301 V4.02 (Beschreibung der Objekte 1600h - 17FFh = Receive PDO Mapping Parameter) :

- >
- >> For changing the PDO mapping first the PDO has to be deleted,
- >> the sub-index 0 must be set to 0 (mapping is deactivated).
- >
- > As someone at CiA explained, deleted means: Bit31 of COB-ID = 1.
- >

Leider schien sich der besagte CANopen-Master nicht daran zu halten. Stattdessen wurde beim Schreiben der Konfiguration zunächst der PDO-Kommunikations-Parameter fertig konfiguriert (mit gelöschtem Bit 31 im COB-ID, d.h. PDO bereits gültig). Erst danach wird das PDO-Mapping eingestellt - was bei strikter Implementierung von DS 301 V4.02 eigentlich nicht funktionieren dürfte. In der Tat funktionierte aus *genau diesem Grund* die Konfiguration von MKT's E/A-Modul durch einen CANopen-Master in einer SPS *nicht* (da im E/A-Modul bislang eine strikte CANopen-Implementierung verwendet *wurde*).

Um mit besagtem CANopen-Master zu harmonisieren, kann bei MKT's E/A-Modul das PDO-Mapping nun (seit 2006-11-22) auch dann geändert werden, wenn Bit 31 in Subindex 1 des PDO-Kommunikations-Parameters *nicht* gesetzt ist.

Falls die oben beschriebene (strenggenommen nicht CANopen-konforme) Funktionalität in Ihrem CANopen-Netzwerk Probleme verursachen, setzen Sie sich bitte mit dem Hersteller in Verbindung.

7. Digitale Ein/Ausgänge

Je nach Hardwarevariante verfügt das E/A-Modul über digitale Eingänge, digitale Ausgänge, und --eventuell- auch bidirektionale digitale "I/O"-Klemmen. Letzere sind per Default als Eingang konfiguriert. Per Objekt 0x5FF5 können diese Anschlüsse individuell auch als Ausgang definiert werden (siehe nächstes Kapitel).

Die Ausgangstreiber verfügen über eine Open-Load- und eine Überstrom-Erkennung (näheres im Datenblatt). Eventuell auftretende Fehler werden per Ausgangs-Fehler-LED (rot) wie folgt signalisiert:

- solange kein Fehler auftrat, ist die LED aus
- sobald ein Fehler auftritt, geht die LED an
- verschwindet der Fehler wieder, geht die LED wieder aus, blitzt aber periodisch wieder auf (bis zum Quittieren des Fehlers, bzw Aus- und Wiedereinschalten)

Die Klemmen "DI1" bis "DI8" sind unidirektional, sie dienen -aus Sicht des Moduls- nur als elektrische Eingänge.

Die Klemmen "DO1" bis "DI8" (bzw "DO4") dienen dagegen nur als elektrische Ausgänge (High-Side-Treiber).

Die Klemmen "IO1" bis "IO4" sind bidirektional.

Die digitalen In- und Outputs sind grundsätzlich in Gruppen zu 8 Bit organisiert. Diese 8-Bit-Gruppe entsprechen den CANopen-Objekten 0x6000 ("digital input module" nach DS401) und 0x6200 ("digital output module" nach DS401). Dabei gelten folgende Regeln für die Zuordnung von Klemmen und CANopen-Objekten:

- Sofern vorhanden, befinden sich die ersten 8 FESTEN (nicht bidirektionalen) Eingänge immer in der erste 8-Bit-Gruppe (Subindex 1) von Objekt 0x6000 .
- Entsprechendes gilt auch für die ersten 8 FESTEN (nicht bidirektionalen) Ausgänge, die immer in der erste 8-Bit-Gruppe (Subindex 1) von Objekt 0x6200 zugeordnet sind.
- Die "bidirektionalen" I/O's folgen erst danach. Sie belegen immer 'eigene' 8-Bit-Gruppen in den Objekten 0x6000 **und** 0x6200. Details dazu in Kapitel 7.1.

7.1 Bidirektionale digitale I/O-Klemmen

Welche Leitung als Eingang, und welche als Ausgang verwendet wird, wird mit Objekt 0x5FF5 (im CANopen-OD) eingestellt. Unter Umständen kann die Konfiguration auch per Terminalprogramm über die serielle Schnittstelle erfolgen. Wir empfehlen aber dringend, dies mit einem geeigneten CANopen-Konfigurationstool bzw CANopen-Master in der SPS zu erledigen !

Wie bereits in der Einleitung dieses Kapitels erwähnt, folgen die "bidirektionalen" I/O's in der Reihenfolge der CANopen-Subindizes erst nach den "festen" Eingängen und den "festen" Ausgängen ¹. Kurz zusammengefasst, die Anschlüsse "DI1"..."DI8" finden Sie *immer* in Objekt 0x6000.01, "DO1"..."DO8" *immer* in Objekt 0x6200.01; die Anschlüsse "IO1"..."IO8" belegen den nächsten freien Subindex (der weder von DIx noch DOx besetzt ist).

Hier einige Beispiele für verschiedene E/A-Baugruppen :

EA7-COMPACT MONO16 V1.0 :

- 8 feste Eingänge "DI1" bis "DI8" in Objekt 0x6200, Subindex 1 .
- 4 feste Ausgänge "DO1" bis "DO4" in Objekt 0x6200, Subindex 1 .
- 4 bidirektionale Ein/Ausgänge "IO1" bis "IO4" in Objekt 0x6000, Subindex 2 (als Eingang); und in Objekt 0x6200, Subindex 2 (als Ausgang).
- Port-Richtungs-Definitionen für "IO1" bis "IO4" daher in Objekt 0x5FF5, Subindex 2 (nicht Subindex 1 !).

Fiktives Modul mit 32 bidirektionalen I/O-Klemmen:

- Bidirektionale Ein/Ausgänge "IO1" bis "IO32" in Objekt 0x6000, Subindex 1 bis 4 (als Eingänge); und in Objekt 0x6200, Subindex 1 bis 4 (als Ausgänge).
- Port-Richtungs-Definitionen für "IO1" bis "IO32" entsprechend in Objekt 0x5FF5, Subindex 1 bis 4.

Fiktives Modul mit 8 festen Eingängen, keinen festen Ausgängen, aber 8 bidirektionalen I/O's:

- 8 digitale Eingänge "DI1" bis "DI8" in Objekt 0x6000, Subindex 1
- keine "reinen" Ausgänge vorhanden, trotzdem existiert Objekt 0x6200, Subindex 1, als "Dummy ohne Funktion" damit die bidirektionalen I/O's in den Objekten 0x6000 und 0x6200 die gleichen Subindizes belegen können !
- 8 digitale Ein/Ausgänge "IO1" bis "IO8" in Objekt 0x6000 und 0x6200, jeweils Subindex 2 (!).
- Port-Richtungs-Definitionen für "IO1" bis "IO8" entsprechend in Objekt 0x5FF5, Subindex 2.

¹ Durch dieses Prinzip wird sichergestellt, daß sich die Bedeutung der Subindizes in den Objekten 0x6000 und 0x6200 für die "normalen" Eingänge bzw Ausgänge nicht in Abhängigkeit der "bidirektionalen" I/O's ändern kann. Die "bidirektionalen" I/O's stellen -so gesehen- einen hoffentlich seltenen Sonderfall dar !

8. Testfunktionen

8.1 Per Codierschalter gesteuerte Testmöglichkeiten

Um die weiter folgenden Tests durchzuführen, muss der Baudraten-Schalter vor dem Einschalten auf "F" gestellt werden (womit normalerweise die Port-Richtungen eingestellt werden). Nach dem Einschalten kann dann mit dem Baudraten-Schalter der gewünschte Test gewählt werden:

- "F" Einstellen der Port-Richtungen (siehe Kapitel 4)
- "E" Lampentest: Alle LEDs an.

8.2 Systemtest per Diagnoseschnittstelle

Zum Aktualisieren der Firmware und für Testzwecke verfügt das E/A-Modul über eine asynchrone serielle Schnittstelle („RS232 mit TTL-Pegel“). Zur Verbindung mit dem PC dient ein Testadapter mit RS232-Pegelwandler und DIP-Schalter für die Umschaltung zwischen „Programmiermodus“ und „Testmodus“.

Hinweis: Verwenden Sie diese Schnittstelle -wenn überhaupt- nur mit dem geeigneten Testadapter, und wirklich **nur zu Testzwecken !**

Im Testmodus kann mit einem geeigneten Terminalprogramm (z.B. Hyperterminal[®]) eine Verbindung ohne CAN-Bus zwischen PC und E/A-Modul hergestellt werden. Die notwendigen Einstellungen für die serielle Schnittstelle sind:

9600 bits pro Sekunde
8 Datenbits
keine Parität
1 Stopbit
kein Protokoll

Zur Steuerung des E/A-Moduls im Testmodus dienen einzelne ASCII-Buchstaben, die per Terminalprogramm gesendet werden können. Das E/A-Modul reagiert dann mit entsprechenden Anzeigen als Klartext. Verfügbare Kommandos (als einzelner Kleinbuchstabe):

- e Enter debug/test mode
Aktiviert den Debug- bzw. Testmodus. Dabei wird die Firmware-Variante, die Firmware-Version und das Compilationsdatum angezeigt. Die meisten der folgenden Testkommandos sind erst nach dieser Aktivierung verfügbar.
- h Help
Zeigt eine Übersicht aller verfügbaren Kommandos an
- a Analog input values
Zeigt den aktuellen Zustand der analogen Eingänge an
- d DIP-switch-test / jumpers used for configuration.
Diese Funktion kann verwendet werden, um zu prüfen ob alle Jumper bzw DIP-Schalter passend gesetzt sind (z.B. für die CAN-Baudrate, Knotennummer, Konfiguration der analogen Ein/Ausgänge, Umschaltung Frequenzeingang oder digitaler Ausgang, ...)

-
- f Frequency inputs
Zeigt die wichtigsten Variablen des Frequenzmessers an. Zum Beenden der Endlosschleife eine beliebige Taste drücken.
 - i digital inputs
Zeigt den aktuellen Zustand der digitalen Eingänge an.
 - k Kill object dictionary in non-volatile memory (EEPROM).
Damit kann das Modul „radikal“ in den Ursprungszustand versetzt werden. Alle Einträge im nichtflüchtigen Speicher des Objektverzeichnisses werden gelöscht. Empfehlenswert nach Firmware-Update, falls „neues“ und „altes“ OD nicht zueinander passen.
 - o digital Outputs (kleines "o", falls die Textverarbeitung dies wieder in eine Aufzählung umwandelt :)
Erlaubt einen schnellen Test aller digitalen Ausgänge auch ohne CAN-Bus.
 - r Ramp
Erzeugt eine Rampenfunktion an den analogen Ausgängen, und ein durchlaufendes Testmuster an den digitalen Ausgängen. Verwendbar als Schnelltest aller Ausgänge.
 - q Quit
Beendet den Test/Debug-Modus, die normale CANopen-Kommunikation wird wieder aufgenommen.