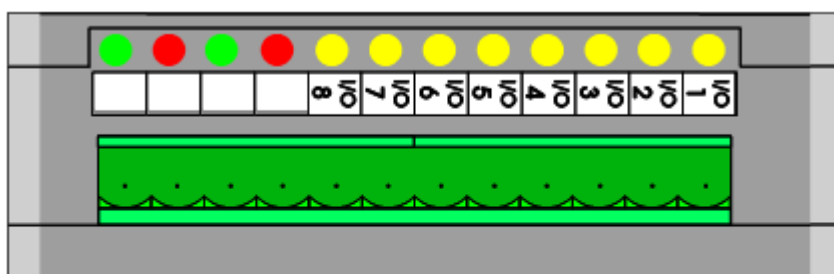


Software description  
I/O module type „EA7 DIO8“  
alias „EA-ARM7 8IO“  
with ARM7-CPU and 8 digital I/O-Ports  
Version 1.0



EA7 DIO8

**PRELIMINARY**

Document Nr: 85066

Original: [C:\cproj\ea\\_arm7\DOKU\art85066\\_SW\\_EA7\\_DIO8\\_V10\\_EN.odt](C:\cproj\ea_arm7\DOKU\art85066_SW_EA7_DIO8_V10_EN.odt)

The content of this documentation has been written and double checked with great care. But **MKT Systemtechnik** -named "MKT" hereafter- cannot be held liable for any damage resulting from the use of this document. The descriptions and technical aspects do not represent any legally-binding guarantee or properties of the product.

Information in this document or in the product may be subject to change without notice. The terms and conditions (in german "AGB") on the MKT website apply.

The copyright for this documentation is at **MKT**. Distribution and duplication in any form requires the written consent by **MKT** .

**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](mailto:info@mkt-sys.de)

Internet: [www.mkt-sys.de](http://www.mkt-sys.de)

# Contents

- [1. REVISION HISTORY OF THIS DOCUMENT.....4](#)
- [2. PREFACE.....4](#)
- [3. CONNECTORS AND LED INDICATORS.....6](#)
- [4. CONFIGURATION VIA DIP-SWITCHES.....7](#)
- [5. CANOPEN.....9](#)
- [6. DIGITAL IN- AND OUTPUTS.....14](#)
- [7. TEST FUNCTIONS.....15](#)

## 1. Revision history of this document

Versionsnummer	Datum (ISO8601)	Autor	Hinweise, Änderungen
V1.0	2006-03-28	W.Büscher	Erstellung für "EA7 DIO8 V1.0" (deutsch)
V1.01	2006-08-30	W.Büscher	CAN-Baudrate 800kBit/sec entfernt
V1.02	2006-11-23	W.Büscher	Abweichungen vom CANopen-Standard erläutert
V1.03	2012-06-08	W.Büscher	Konvertierung aus Word-Format nach OpenOffice. Vereinheitlichung der Produktnamen: unnötige/irreführende Bindestriche entfernt. Beschreibung der Anschlussklemme und LEDs.
	2013-04-02	W.Büscher	First translation into english language

## 2. Preface

<not translated yet...>

Dieses Dokument beschreibt die Softwarefunktionen eines Low-Cost-E/A-Moduls von MKT Systemtechnik mit 8 digitalen Ein- oder Ausgängen mit dem Projektnamen „EA7 DIO8“. Der endgültige Produktname stand bei der Erstellung dieses Dokumentes noch nicht fest. ~~Die Hardwarebeschreibung zu diesem Gerät finden Sie unter der Artikelnummer 85301 .~~

Zur Kommunikation auf der Feldebene dient der CAN-Bus. Das Kommunikationsprotokoll ist kompatibel zu CANopen DS301V4.0. Das Geräteprofil entspricht weitgehend CANopen DS401, mit einigen herstellerspezifischen Erweiterungen (z.B. zur Definition der Klemmen als Ein- oder Ausgänge).

Wesentliche Eigenschaften der Firmware :

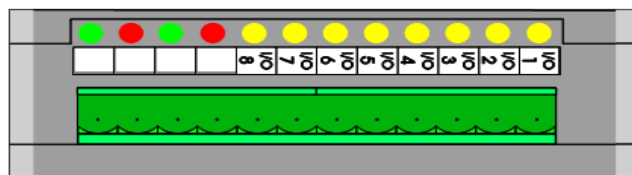
- <sup>35</sup><sub>17</sub> CANopen mit speziellen Erweiterungen (siehe Kapitel 5.2.1)
- <sup>35</sup><sub>17</sub> CANopen-Protokollstack der Firma SYS TEC electronic GmbH
- <sup>35</sup><sub>17</sub> drei Empfangs- und drei Sende-PDOs (Prozeßdatenobjekte)
- <sup>35</sup><sub>17</sub> variables PDO-Mapping, Defaulteinstellung siehe Kapitel 5.2 .
- <sup>35</sup><sub>17</sub> Synchroner oder asynchroner PDO-Übertragung, periodisch oder ereignisgesteuert nach DS301 bzw DS401
- <sup>35</sup><sub>17</sub> ein SDO (Servicedatenobjekt) für die Parametrierung per CAN
- <sup>35</sup><sub>17</sub> Heartbeat, Emergency Messages
- <sup>35</sup><sub>17</sub> per CAN parametrierbare SYNC-Sende-Funktion

Weitere Informationen :

- <sup>35</sup><sub>17</sub> ~~Anschlußbelegung, technische Daten, elektrische Spezifikation entnehmen Sie bitte der Hardwarebeschreibung des Gerätes im Download-Bereich bei [www.mkt-sys.de](http://www.mkt-sys.de) .~~
- <sup>35</sup><sub>17</sub> Eine von CANopen-Konfigurationstools einlesbare elektronische Gerätebeschreibung (EDS-File) finden Sie auf der Webseite von MKT Systemtechnik, oder auf der ['MKT-CD'](#) .
- <sup>35</sup><sub>17</sub> Ein einfaches Testprogramm für dieses E/A-Modul befindet sich im Installationsarchiv von MKT's CAN-Tester für Windows (Datei „EaDIO8Test“).

<sup>35</sup><sub>17</sub> Details zu den CANopen-Übertragungsprotokollen finden Sie bei CiA (CAN in Automation), <http://www.can-cia.org/canopen/>

### 3. Connectors and LED indicators



EA7 DIO8

<not translated yet...>

Anschlüsse auf der Vorderseite, von *rechts nach links* in der obigen Abbildung:

Nr.	Beschriftung	Funktion
1	I/O 1	digitaler Ein- oder Ausgang
2	I/O 2	digitaler Ein- oder Ausgang
3	I/O 3	digitaler Ein- oder Ausgang
4	I/O 4	digitaler Ein- oder Ausgang
5	I/O 5	digitaler Ein- oder Ausgang
6	I/O 6	digitaler Ein- oder Ausgang
7	I/O 7	digitaler Ein- oder Ausgang
8	I/O 8	digitaler Ein- oder Ausgang
9	24 V	Versorgungsspannung
10	24 V	Versorgungsspannung
11	GND	Masse
12	GND	Masse

Jeder der acht digitalen I/O-Klemmen ist eine *gelbe* LED direkt neben dem Anschluss zugeordnet. Bedeutung der *roten* und *grünen* LEDs (in der obigen Abbildung *links*) :

LED	Bedeutung
"RUN" (grün)	CANopen-"RUN"-LED. Details nur CiA DR 303 "Indicator Specification". Dauerlicht im Zustand 'operational' . Blinkend im Zustand 'pre-operational'.
"ERR" (rot)	CANopen-"Error"-LED. Details nur in CiA DR 303 .
"RES" (grün)	Reserviert für zukünftige Erweiterungen
"ERROR OUT" (rot)	Fehler vom Ausgangstreiber für die digitalen Ausgänge. Normalerweise aus (= 'kein Fehler') .

Die Belegung des CAN-Steckers, und die elektrischen Daten der Anschlussklemmen entnehmen Sie bitte ggf. dem Datenblatt. Die elektrischen Eigenschaften sind kein Bestandteil der Software-Beschreibung.

## 4. Configuration via DIP-switches

<not translated yet...>

Der 12-polige DIP-Schalter an der Seite des Moduls wird zur Einstellung der Baudrate sowie des CANopen-Node-Identifiers („Knotennummer“) verwendet. Diese Parameter werden dabei als Binärzahl mit 4 beziehungsweise 8 Bit eingegeben.

OFF = Logisch ‚0‘

ON = Logisch ‚1‘

DIP-Schalter S1 bis S8 dienen zum Einstellen der Knotennummer, S9 bis S12 für die CAN-Baudrate. Für CANopen-Profis: Die Einstellung dieser Parameter per LSS ist nicht möglich !

### 4.1 CAN-Baudrate

Hinweis: Die Codierung der CAN-Baudrate ist nicht kompatibel mit MKT's "EA 515" !

S12 / Bit 3	S11 / Bit 2	S10 / Bit 1	S9 / Bit 0	CAN-Baudrate [kbit/s]
0	0	0	0	1 Mbit / s
0	0	0	1	500 kbit / s
0	0	1	0	250 kbit / s
0	0	1	1	125 kbit / s
0	1	0	0	100 kbit / s
0	1	0	1	50 kbit / s
0	1	1	0	20 kbit / s (erfordert eine modifizierte Hardware <sup>1</sup> )
0	1	1	1	10 kbit / s (erfordert eine modifizierte Hardware)
1	0	0	0	reserviert (für 800 kbit/sec, z.Z. nicht realisierbar)
1	0	0	1	reserviert (für 666.7 kbit/sec, z.Z. nicht realisierbar)
1	0	1	0	reserviert
1	0	1	1	reserviert
1	1	0	0	reserviert
1	1	0	1	reserviert
1	1	1	0	reserviert
1	1	1	1	Umschaltung in den Programmiermodus

<sup>1</sup> 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 Identifier

S8 / Bit 7	S7 / Bit 6	S6 / Bit 5	S5 / Bit 4	S4 / Bit 3	S3 / Bit 2	S2 / Bit 1	S1 / Bit 0	NODE-ID
0	0	0	0	0	0	0	0	Testmodus
0	0	0	0	0	0	0	1	1
...	...	...	...	...	...	...	...	...
1	1	1	1	1	1	1	1	255

## 4.3 Changing the input/output port directions for module „DIO8“

<not completely translated yet...>

Note: Wherever possible, you should use a suitable CANopen configuration tool to make these changes. After importing the EDS file, the tool should "know how" (using object 0x5FF5, subindex 1) !

Only if nothing else helps, use the DIP switches to modify the I/O-port directions as described below. The DIP switches are quite fragile, very uncomfortable to use, and must be treated with great care to avoid breaking them !

To change the port directions (for the bidirectional in- and outputs) via DIP switch, proceed as follows:

- 1) Turn the module's supply voltage **off**
- 2) Set dip switches S1 to S8 according to the required "port directions" (seen from the module's point of view) :  
all digital outputs: ‚ON‘  
all digital inputs: ‚OFF‘
- 3) Set switches S9..S12 to ON (which causes the module to enter 'Programming mode' when the power is turned on in the next step)
- 4) Turn the supply voltage **on**, and wait until both CANopen-indicator-LEDs are blinking in the same rhythm
- 5) Flip S12 from ON back to OFF. The module will now copy the state of dip switches S1..S8 (as 8-bit-value) into the port-direction-register for the first 8 configurable I/O ports, and saves this value permanently in an EEPROM .  
Note: In the „EA7 DIO8“ all eight I/O-port-lines are randomly configurable; but this may be different in future devices: Not all of the available I/O lines may be "electrically bi-directional" ! Please consult the hardware manual of those devices for details.
- 6) Turn the supply voltage **off again**, and restore the default settings for S1..S12 for your desired CANopen-node-ID and CAN-baudrate (according to chapter 4.1 and 4.2) setzen.



## 5. CANopen

### 5.1 Object directory

<not translated yet...>

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:

<sup>35</sup><sub>17</sub> 1000-1FFF CANopen-Kommunikationsprofil. Siehe CiA DS 301 .

<sup>35</sup><sub>17</sub> 6000 Digitale Eingänge in 8-Bit-Gruppen. Siehe CiA DS401 ...

<sup>35</sup><sub>17</sub> 6005-6006 Parametrierung des ereignisgesteuerten Sendens der digitalen Eingänge

<sup>35</sup><sub>17</sub> 6200 Digitale Ausgänge in 8-Bit-Gruppen

<sup>35</sup><sub>17</sub> 6208 "Filtermaske" für die digitalen Ausgänge, ebenfalls in 8-Bit-Gruppen

<sup>35</sup><sub>17</sub> 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.

#### 5.1.1 Manufacturer specific part of the CANopen OD

<not translated yet...>

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 !
5FF5.0 1	Var	Unsigned8	Input/Output directions for digital I/O 1..8	one bit per port: 0=input, 1=output

## 5.2 Process Data Objects (PDO)

<not translated yet...>

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 + Knotennummer<sup>1</sup>

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 (Objekt 6411.01, 16 bit)

3. und 4. Datenbyte : Analog-Ausgang 2 (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

<sup>1</sup> Siehe Hinweis zur automatischen, Knotennummer-abhängigen Umschaltung der PDO-CAN-Identifizier in Kapitel 5.2.1 .

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)

### **5.2.1 Node-ID-independent, automatic switching of PDO CAN Identifiers**

<not translated yet...>

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“):

<sup>35</sup>/<sub>17</sub> RPDO1: CAN-ID 200h + < Knotennummer des Moduls>  
<sup>35</sup>/<sub>17</sub> RPDO2: CAN-ID 300h + < Knotennummer des Moduls>  
<sup>35</sup>/<sub>17</sub> RPDO3: CAN-ID 400h + < Knotennummer des Moduls>  
<sup>35</sup>/<sub>17</sub> TPDO1: CAN-ID 180h + < Knotennummer des Moduls>  
<sup>35</sup>/<sub>17</sub> TPDO2: CAN-ID 280h + < Knotennummer des Moduls>  
<sup>35</sup>/<sub>17</sub> 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..127<sub>10</sub> 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:

<sup>35</sup><sub>17</sub> 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.

<sup>35</sup><sub>17</sub> 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.

<sup>35</sup><sub>17</sub> 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).

## 5.2.2 Specialities of the PDO configuration via SDO

<not translated yet...>

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*:

### 5.2.2.1 Deviation from CANopen for the PDO communication parameters

According to CANopen-Spezifikation DS301 V4.02, the CAN identifier of a PDO may only be modified, if bit 31 in that object (for example, 1400.01) has been set, which makes the PDO "invalid". Quote from DS301 V4.02 (description of objects 1600h - 17FFh = Receive PDO Mapping Parameter) :

>

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

>

Since 2006-11-22, some of the I/O modules by MKT Systemtechnik allow overwriting the CAN-ID even if bit 31 is cleared (not set before trying to write the ID itself). Strictly spoken, this is not conformal with the cluttered CANopen standard (DS 301 V4.02). But without this ugly modification, a "good customer's" PLC (with integrated CANopen master) was unable to modify the module's PDO CAN identifier at all !

### 5.2.2.2 Abweichung vom CANopen-Standard im PDO Mapping Parameter

According to CANopen specification DS301 V4.02 also the 'mapping' of a PDO (for example, via object 1600) can only be modified, if bit 31 in the PDO's "COB ID" (CANopen geek speak for "CAN Identifier") is set, which makes the PDO "invalid". Quoted from DS301 V4.02 (specification of objects 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.

>

Unfortunately, the already mentioned CANopen master in the "good customer's" PLC didn't seem to give a damn. Instead, the CANopen master first finished configuring the PDO communication parameters completely (including bit 31 cleared, i.e. making the PDO "valid"). After that, the PDO mapping was configured, which -if we had implemented CANopen DS 301 V4.02 strictly- should not be possible. In fact, *before MKT didn't deliberately implemented the same mistake as the PLC manufacturer in their CANopen master*, MKT's I/O-module could not be configured by the PLC !

To make the I/O module "compatible" with the customer's poorly designed PLC, the PDO mapping in MKT's I/O module (at least the "E/A7 - DIO8") will happily modify its PDO mapping (since 2006-11-22), even if bit 31 in subindex of the PDO-communication-parameter is *not* set ! (It's a bug, we know, but unavoidable thanks to the cluttered CANopen standard).

If the aforementioned, strictly spoken NON-CANOPEN-CONFORMAL, behaviour causes problems in your CANopen network, please get in contact with the manufacturer. With some luck, the "good customer's" broken CANopen master has been fixed in the meantime, so we can fix the firmware for our CANopen I/O-modules, too.

## 6. Digital In- and Outputs

<not translated yet...>

Beim "EA7 DIO8" sind die 8 I/O-Klemmen per Default als Eingang konfiguriert. Per Objekt 0x5FF5, Subindex 1, können diese Anschlüsse individuell auch als Ausgang definiert werden (notfalls auch per DIP-Schalter, siehe Kapitel 4.3).

Die in diesem Modul verwendeten Ausgangstreiber verfügen über eine Open-Load- und eine Überstrom-Erkennung. Eventuell auftretende Fehler werden per Ausgangs-Fehler-LED (rot) wie folgt signalisiert:

<sup>35</sup>/<sub>17</sub> solange kein Fehler auftrat, ist die LED aus

<sup>35</sup>/<sub>17</sub> sobald ein Fehler auftritt, geht die LED an

<sup>35</sup>/<sub>17</sub> verschwindet der Fehler danach, geht die LED zunächst wieder aus,  
blitzt dann aber periodisch auf (ca 1 mal pro Sekunde),  
zum zu signalisieren dass seit dem Einschalten ein Fehler aufgetreten ist.

## 7. Test Functions

<not translated yet...>

### 7.1 System test via diagnostic interface

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:

<sup>35</sup><sub>17</sub> 9600 bits pro Sekunde  
<sup>35</sup><sub>17</sub> 8 Datenbits  
<sup>35</sup><sub>17</sub> keine Parität  
<sup>35</sup><sub>17</sub> 1 Stopbit  
<sup>35</sup><sub>17</sub> kein Protokoll

Zur Steuerung des E/A-Moduls im Testmodus dienen einzelne ASCII-Buchstaben, die z.B. per Hyperterminal 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.

---

<sup>35</sup><sub>17</sub> digital Outputs (kleines "o")

Erlaubt einen schnellen Test aller digitalen Ausgänge auch ohne CAN-Bus

(sorry, diese *nervtötende Textverarbeitung* ersetzt das kleine "o" durch ein Aufzählungszeichen. Unter "Extras" nicht abzuschalten ?!)

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.