

# **CANopen ChipF40**

## **System-Manual**

**Auflage Juni 2012**

Im Buch verwendete Bezeichnungen für Erzeugnisse, die zugleich ein eingetragenes Warenzeichen darstellen, wurden nicht besonders gekennzeichnet. Das Fehlen der © Markierung ist demzufolge nicht gleichbedeutend mit der Tatsache, daß die Bezeichnung als freier Warenname gilt. Ebenso wenig kann anhand der verwendeten Bezeichnung auf eventuell vorliegende Patente oder einen Gebrauchsmusterschutz geschlossen werden.

Die Informationen in diesem Handbuch wurden sorgfältig überprüft und können als zutreffend angenommen werden. Dennoch sei ausdrücklich darauf verwiesen, daß die Firma SYS TEC electronic GmbH weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf den Gebrauch oder den Inhalt dieses Handbuches zurückzuführen sind. Die in diesem Handbuch enthaltenen Angaben können ohne vorherige Ankündigung geändert werden. Die Firma SYS TEC electronic GmbH geht damit keinerlei Verpflichtungen ein.

Ferner sei ausdrücklich darauf verwiesen, daß SYS TEC electronic GmbH weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf falschen Gebrauch oder falschen Einsatz der Hard- bzw. Software zurückzuführen sind. Ebenso können ohne vorherige Ankündigung Layout oder Design der Hardware geändert werden. SYS TEC electronic GmbH geht damit keinerlei Verpflichtungen ein.

© Copyright 2012 SYS TEC electronic GmbH. Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form ohne schriftliche Genehmigung der Firma SYS TEC electronic GmbH unter Einsatz entsprechender Systeme reproduziert, verarbeitet, vervielfältigt oder verbreitet werden.

<b>Kontakt</b>	<b>Direkt</b>	<b>Ihr Lokaler Distributor</b>
Adresse:	SYS TEC electronic GmbH August-Bebel-Str. 29 D-07973 Greiz GERMANY	Sie finden eine Liste unserer Distributoren unter <a href="http://www.systec-electronic.com/distributors">http://www.systec-electronic.com/distributors</a>
Angebots-Hotline:	+49 (0) 36 61 / 62 79-0 <a href="mailto:info@systec-electronic.com">info@systec-electronic.com</a>	
Technische Hotline:	+49 (0) 36 61 / 62 79-0 <a href="mailto:support@systec-electronic.com">support@systec-electronic.com</a>	
Fax:	+49 (0) 36 61 / 6 79 99	
Webseite:	<a href="http://www.systec-electronic.com">http://www.systec-electronic.com</a>	

7. Auflage Juni 2012

---

<b>1</b>	<b>Einleitung</b> .....	<b>1</b>
<b>2</b>	<b>Einführung in den CANopen ChipF40</b> .....	<b>3</b>
<b>3</b>	<b>Geräteübersicht</b> .....	<b>5</b>
	3.1 Anschlußbelegung.....	5
	3.2 Belegung der Stiftleiste (Dipmodul-Connector).....	7
	3.3 Konfigurationsmöglichkeiten.....	9
	3.4 CAN-Software.....	9
	3.5 RESET.....	10
	3.6 Einstellung der Kommunikationsparameter.....	11
	3.7 Zuordnung der IO-Pins zur Konfiguration F40.....	14
	3.8 Zuordnung der IO-Pins zur Konfiguration F40 V3.....	16
	3.9 Technische Daten.....	18
<b>4</b>	<b>Inbetriebnahme</b> .....	<b>20</b>
	4.1 Versorgungsspannung.....	20
	4.2 CAN-Schnittstelle.....	20
<b>5</b>	<b>Schnelleinstieg</b> .....	<b>22</b>
	5.1 Start der Baugruppe (Boot-Up).....	22
	5.2 Stop der Baugruppe.....	22
	5.3 CAN-Identifizier / -Nachrichten.....	23
	5.4 Mapping der I/O's.....	23
	5.4.1 Default-Mapping CANopen ChipF40.....	24
	5.4.2 Default-Mapping CANopen ChipF40 V3.....	24
	5.5 Baugruppenreset.....	25
	5.6 Node Guarding.....	25
<b>6</b>	<b>Controller Area Network CAN</b> .....	<b>28</b>
	6.1 Kommunikation mit CANopen.....	28
	6.2 CANopen - offene, industrielle Kommunikation.....	29
	6.3 Aufbau und Verdrahtung.....	31
<b>7</b>	<b>CANopen-Kommunikation</b> .....	<b>32</b>
	7.1 Grundlagen.....	32
	7.2 Geräteprofile.....	32
	7.3 Kommunikationsprofil.....	33
	7.4 Servicedatenobjekte.....	33
	7.5 Prozeßdatenobjekte.....	34
	7.6 PDO-Mapping.....	36
	7.7 Fehlerverarbeitung.....	37
	7.8 Netzwerkdienste.....	37
	7.8.1 Life Guarding.....	38
	7.8.2 Heartbeat.....	38
	7.8.2.1 Heartbeat Producer.....	38
	7.8.2.2 Heartbeat Consumer.....	39
	7.9 Netzwerk Book-Up.....	40

---

7.10	Objetverzeichniseinträge.....	42
7.11	Beschreibung des PDO-Mapping am Beispiel .....	42
7.12	Abbildung der Ein-/Ausgänge auf Objektverzeichniseinträge ..	44
<b>8</b>	<b>Funktionalität des CANopen ChipF40 .....</b>	<b>46</b>
8.1	CANopen Statusübergänge.....	46
8.2	Power On.....	47
8.3	PRE-OPERATIONAL.....	47
8.4	OPERATIONAL.....	47
8.5	STOPPED .....	47
8.6	Wiederanlauf nach Reset/ Power-On.....	48
8.7	NMT-Boot-Konfiguration.....	49
8.8	Funktionalität der analogen Eingänge .....	50
8.8.1	Ablage der analogen Werte.....	50
8.8.2	Berechnungsformel für den analogen Eingangswert....	51
8.8.3	Auswahl des Interrupt Triggers .....	52
8.8.4	Interrupt Quelle.....	52
8.8.5	Interrupt Aktivierung .....	53
8.8.6	Interrupt oberer und unterer Grenzwert.....	53
8.8.7	Deltafunktion .....	54
8.8.8	Beispiel zu den Triggerbedingungen.....	54
8.9	Funktionalität der PWM Ausgänge (nicht F40 V3) .....	55
8.10	Emergency Telegramm.....	56
8.10.1	Error Code.....	57
8.10.2	Error Register.....	57
8.11	Statusanzeige über Run und Error LED .....	58
8.11.1	Run LED .....	58
8.11.2	Error LED .....	59
<b>9</b>	<b>Systemverhalten im Fehlerfall.....</b>	<b>60</b>
9.1	Zustand des CANopen ChipF40 im Fehlerfall .....	60
9.2	Verhalten der Ausgänge im Fehlerfall.....	61
9.2.1	Digitale Ausgänge.....	61
9.2.2	PWM Ausgänge.....	62
9.3	Wechsel von Fehlerfall zum normalen Betrieb .....	62
<b>10</b>	<b>Das CANopen ChipF40 Objektverzeichnis.....</b>	<b>64</b>
<b>11</b>	<b>Das CANopen ChipF40 V3 Objektverzeichnis.....</b>	<b>66</b>
<b>12</b>	<b>Änderungen im Dokument .....</b>	<b>68</b>
<b>Index</b>	<b>.....</b>	<b>69</b>

---

Bild 1:	Position der Anschlüsse .....	6
Bild 2:	Belegung des DIP-Schalters.....	9
Bild 3	Belegung des DIP-Schalters Version 3301002.....	9
Bild 4:	Prinzipieller Aufbau des Reset Eingangs auf dem CANopen ChipF40.....	10
Bild 5:	NMT-Zustandsdiagramm eines CANopen Gerätes .....	41
Bild 6:	Beispiel Triggerbedingungen.....	54
Tabelle 1:	Pinout des DIPmodul-Connectors.....	7
Tabelle 2:	Einstellung der Bitraten .....	13
Tabelle 3:	mögliche Bitraten, die mittels LSS eingestellt werden können.	13
Tabelle 4:	Einstellung der Konfiguration.....	13
Tabelle 5:	Anzahl der Ein-/Ausgänge je nach Konfiguration F40 .....	14
Tabelle 6:	Zuordnung Ein-/Ausgänge zur Konfiguration F40.....	15
Tabelle 7:	Anzahl der Ein-/Ausgänge je nach Konfiguration F40 V3 ....	16
Tabelle 8:	Zuordnung Ein-/Ausgänge zur Konfiguration F40 V3.....	17
Tabelle 9:	Zuordnung der CAN-ID zu den Prozeßdaten .....	23
Tabelle 10:	Mapping der I/O's F40.....	24
Tabelle 11:	Mapping der I/O's F40 V3.....	24
Tabelle 12:	Empfohlene Leitungsparameter .....	31
Tabelle 13:	Aufbau der COB-Identifizier .....	35
Tabelle 14:	Aufbau der Emergency-Nachricht .....	37
Tabelle 15 :	Aufbau der Heartbeat-Nachricht .....	38
Tabelle 16 :	Aufbau des Consumer Heartbeat Time – Eintrages .....	39
Tabelle 17:	Ermittlung der COB-Identifizier aus der Knotenadresse.....	40
Tabelle 18:	Basisidentifizier .....	40
Tabelle 19 :	Legende zum Zustandsdiagramm.....	41
Tabelle 20:	Objektverzeichnis der Geräteparameter F40 .....	44
Tabelle 21:	Objektverzeichnis der Geräteparameter F40 V3.....	45
Tabelle 22:	NMT-Master Telegramme zur Zustandssteuerung.....	46
Tabelle 23 :	SDOs zum Setzen der NMT-Boot-Konfiguration.....	50

---

Tabelle 24 : Ablage der Analogdaten.....	51
Tabelle 25: Interrupt Trigger Bits.....	52
Tabelle 26: Emergency Telegramm (BYTE-Feld) .....	56
Tabelle 27: Run LED Zustände.....	58
Tabelle 28: Error Led Zustände.....	59
Tabelle 29: Beispiel für Verhalten der Ausgänge im Fehlerfall.....	61
Tabelle 30: Beispiel für Verhalten der Ausgänge im Fehlerfall.....	62
Tabelle 31: Objektverzeichnis (Object Dictionary) des F40 .....	65
Tabelle 32: Objektverzeichnis (Object Dictionary) des F40 V3 .....	67

## 1 Einleitung

Dieses Handbuch beschreibt die Funktionen des CANopen ChipF40.

### **Anmerkungen zum EMV-Gesetz für den CANopen ChipF40**



Der CANopen ChipF40 ist als Zulieferteil für den Einbau in ein Gerät (Weiterverarbeitung durch Industrie) bzw. als Entwicklungsboard für den Laborbetrieb (zur Hardware- und Softwareentwicklung) bestimmt.

Nach dem Einbau in ein Gerät oder bei Änderungen/Erweiterungen an diesem Produkt muß die Konformität nach dem EMV-Gesetz neu festgestellt und bescheinigt werden. Erst danach dürfen solche Geräte in Verkehr gebracht werden.

Die CE-Konformität gilt nur für den hier beschriebenen Anwendungsbereich unter Einhaltung der im folgenden Handbuch gegebenen Hinweise zur Inbetriebnahme! Der CANopen ChipF40 ist ESD empfindlich und darf nur an ESD geschützten Arbeitsplätzen von geschultem Fachpersonal ausgepackt und gehandhabt bzw. betrieben werden.

Der CANopen ChipF40 ist ein Modul für den Bereich Automatisierungstechnik. Durch die Verwendung des CAN-Bus - einer Standard-Netzwerkschnittstelle für verschiedenste Automatisierungslösungen sowie dem standardisierten Netzwerkprotokoll CANopen ergeben sich geringere Entwicklungszeiten bei günstigen Kosten der Hardware. Durch die on-board Realisierung der CANopen-Netzwerkschicht ist die Erstellung einer Firmware durch den Anwender überflüssig geworden.

Mit dem Konzept der Microcontroller-Module von SYS TEC ist es Entwicklungsingenieuren möglich, Entwicklungszeiten zu verkürzen, Entwicklungskosten zu reduzieren und die Durchführung eines Projektes von der Idee bis zur Markteinführung wesentlich zu

beschleunigen. Für weitere Informationen wenden Sie sich bitte an folgende Adressen:

<b>Kontakt</b>	<b>Direkt</b>	<b>Ihr Lokaler Distributor</b>
Adresse:	SYS TEC electronic GmbH August-Bebel-Str. 29 D-07973 Greiz GERMANY	Sie finden eine Liste unserer Distributoren unter <a href="http://www.systec-electronic.com/distributors">http://www.systec-electronic.com/distributors</a>
Angebots-Hotline:	+49 (0) 36 61 / 62 79-0 <a href="mailto:info@systec-electronic.com">info@systec-electronic.com</a>	
Technische Hotline:	+49 (0) 36 61 / 62 79-0 <a href="mailto:support@systec-electronic.com">support@systec-electronic.com</a>	
Fax:	+49 (0) 36 61 / 6 79 99	
Webseite:	<a href="http://www.systec-electronic.com">http://www.systec-electronic.com</a>	



## 2 Einführung in den CANopen ChipF40

Der CANopen ChipF40 ist in verschiedenen Firmware-Versionen erhältlich:

MM-217-Y	CANopen ChipF40 Standardversion
MM-217-V3Y	CANopen ChipF40 V3, Kompatibel zum CANopen Chip164.
3301002	CANopen ChipF40, 63 Knotenadressen einstellbar über DIP-Switch, Default- Konfiguration 1, änderbar nur über CANopen SDO

Auf Unterschiede und Besonderheiten wird im weiteren Verlauf gesondert eingegangen.

Mit dem CANopen ChipF40 wurde eine Baugruppe entwickelt, die mit geringem Platzbedarf und unter Verwendung des standardisierten CANopen Protokolls, digitale Ein- und Ausgänge, analoge Eingänge sowie PWM-Ausgänge zur Verfügung stellt. Dieser Chip ist als Herzstück eines Anwenderproduktes gedacht, und trägt diesem Einsatzgebiet durch seine vielfältigen on-board Konfigurationsmöglichkeiten Rechnung. Die on-board Firmware realisiert hierbei das Netzwerkprotokoll.

Der CANopen ChipF40 hat die Maße eines Standard DIL-40 Präzisionssockels. Somit ist die Voraussetzung für eine kostengünstige Adaptierung des Chip's auf die Anwenderhardware gegeben. Der Chip wird mit einer Spannung von +5V, entsprechend der Spezifikation für Standard TTL-Bausteine, versorgt.

Die Firmware des CANopen ChipF40 hat die volle Funktion eines CANopen-Slaves und wird vom CiA (CAN in Automation e.V.) zertifiziert. In der vorliegenden Version der Firmware werden 11-Bit Identifier (Full - CAN 2.0B passive) unterstützt. Es werden das standardisierte Geräteprofil nach **CiA DSP401** und das Kommunikationsprofil nach **CiA DS301 V4.01** unterstützt. Ein weiteres entscheidendes Merkmal des CANopen ChipF40 sind die oben bereits genannten on-board Konfigurationsmöglichkeiten für das CANopen - Netzwerk und die Konfiguration der PDO's (Aufteilung

der Ein- und Ausgänge hinsichtlich Anzahl sowie Digital- und Analog-I/O's). Zur Speicherung aller Konfigurationsdaten des CANopen-Slaves steht ein serielles E<sup>2</sup>PROM zur Verfügung.

### 3 Geräteübersicht

- Microcontroller Fujitsu MB90F352 mit integriertem FULL-CAN-Controller
- E<sup>2</sup>PROM zur Speicherung der Konfigurationsdaten
- CAN-Bustreiber für die Unterstützung von bis zu 100 CAN-Knoten an einem Bus
- DIP-Schalter für die Einstellung der NODE-ID, Baudrate, I/O-Konfiguration
- Signal für die externe galvanische Trennung des CAN-Busses auf der Stiftleiste vorhanden
- Platinengröße entspricht einem IC im DIL-40-Gehäuse

#### 3.1 Anschlußbelegung

Es sei ausdrücklich darauf hingewiesen, daß bei allen Modulanschlüssen unbedingt die Maximalspannungen und -ströme nicht überschritten werden dürfen. Die Grenzwerte hierfür können Sie dem jeweiligen Controller-Handbuch bzw. dem Kapitel „Technische Daten“ entnehmen. Da eventuell auftretende Störungen stark vom Einsatzgebiet bzw. Anwendungsfall abhängen, obliegt es der Verantwortung des Anwenders, in entsprechend kritischer Umgebung geeignete Schutzmaßnahmen zu treffen.

Alle Anschlüsse des Controllers sind direkt und ungeschützt nach außen geführt. ESD-Richtlinien sind unbedingt einzuhalten.

Wie in *Bild 1* dargestellt, werden alle relevanten Signale an die Stiftleiste X1 (im folgenden DIPmodul-Connector genannt) geführt.

*Tabelle 1* gibt eine Übersicht über die Belegung des DIPmodul-Connectors.



### 3.2 Belegung der Stiftleiste (Dipmodul-Connector)

PIN Nr.	Bezeichnung	E/A	Beschreibung
1, 2	P1.2, P1.3	I/O	Port Pin P1.2, P1.3 des $\mu\text{C}$
3, 4	P4.4, P4.5	I/O	Port Pin P4.4, P4.5 des $\mu\text{C}$
5	/BOOT	I	/BOOT = 0 & RESIN = 1-0-Flanke $\rightarrow$ Boot-Mode aktivieren. Die Eigenschaft des Microcontrollers, mit Hilfe des Boot-Mode Software über die serielle Schnittstelle auf das Modul zu laden, wird für die Flashtools zum Firmwareupdate verwendet.
6	GND	-	Schaltungsmasse 0V
7, 8	P2.0, P2.1	I/O	Port Pin P2.0, P2.1 des $\mu\text{C}$
9	VAREF	-	Anschluss Referenzspannung des $\mu\text{C}$ -internen AD-Wandlers
10	VAGND	-	Bezugspotential Referenzspannung
11, 12, 13, 14, 15, 16, 17, 18	P6.0, P6.1, P6.2, P6.3, P6.4, P6.5, P6.6, P6.7	I	Port Pin P6.0-P6.7 des $\mu\text{C}$
19	RESIN	I	Reset-Eingang des Moduls, 0-1-Flanke löst RESET aus
20	GND	-	Schaltungsmasse 0V
21, 22	P5.0, P5.1	I/O	Port Pin P5.0, P5.1 des $\mu\text{C}$
23, 24, 25, 26, 27, 28, 29, 30	P3.0, P3.1, P3.2, P3.3, P3.4, P3.5, P3.6, P3.7	I/O	Port Pin P3.1-P3.7 des $\mu\text{C}$
31, 32	P5.2, P5.3	I/O	Port Pin P5.2, P5.3 des $\mu\text{C}$
33, 34	P2.2, P2.3	I/O	Port Pin P2.2, P2.3 des $\mu\text{C}$
35	GND	-	Schaltungsmasse 0 V
36	RxDC	I	Empfangsleitung des $\mu\text{C}$ -internen CAN-Controllers
37	CANL	I/O	CANL-Ein/Ausgang des CAN-Treibers
38	CANH	I/O	CANH-Ein/Ausgang des CAN-Treibers
39	TxDC	O	Sendeleitung des $\mu\text{C}$ -internen CAN-Controllers
40	VCC	-	Versorgungsspannung +5 V DC

Tabelle 1: Pinout des DIPmodul-Connectors

Die Zuordnung der I/O's und deren Funktion wird in der *Tabelle 6* bzw. *Tabelle 8* genauer beschrieben.

I – Input

O – Output

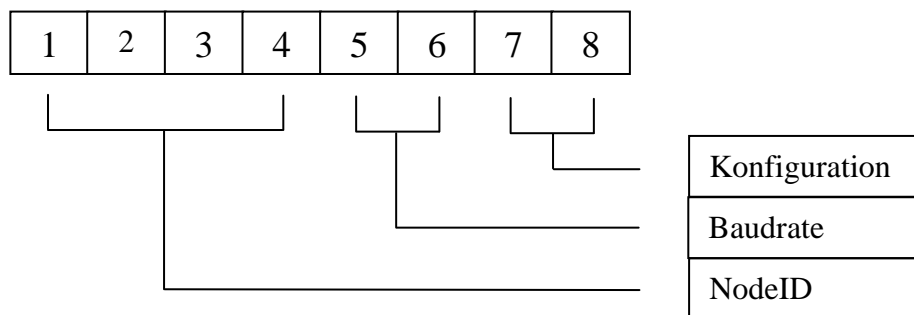
Die Verwendung eines externen CAN-Treibers bzw. eine externe galvanische Entkopplung des CAN-Bus (Verwendung der Signale

RxDC und TxDC statt CANL und CANH) bedingt das Entfernen der Widerstände auf J1 des Moduls. Eine gleichzeitige Verwendung des onboard CAN-Treibers und eines externen CAN-Treibers ist nicht möglich.

### 3.3 Konfigurationsmöglichkeiten

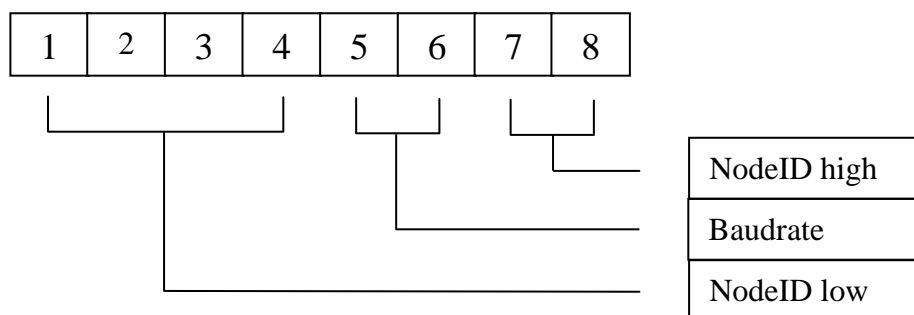
Auf der Baugruppe befindet sich ein 8-facher DIP-Schalter. Über 4 dieser Schalter wird die CANopen Knotenadresse für den CANopen ChipF40 vergeben. Weitere 2 Schalter geben die CAN-Baudrate vom CAN-Bus und die letzten 2 Schalter die Konfiguration der I/O's an. Weitere Konfigurationsmöglichkeiten sind über CANopen möglich. (siehe Kapitel 3.6).

Die Belegung des DIP-Schalters zeigt *Bild 1*. Die Schalterposition "OFF" entspricht dem logischen Wert 0, "ON" entspricht logisch 1.



*Bild 2: Belegung des DIP-Schalters*

Abweichend zur Standardversion des CANopen Chip F40 werden bei der Variante 3301002 die DIP-Schalter 7 und 8 zur Einstellung der Knotenadresse und nicht zur Einstellung der Konfiguration verwendet.



*Bild 3 Belegung des DIP-Schalters Version 3301002*

### 3.4 CAN-Software

Von der Firmwareversion mit CANopen-Protokoll werden 11-Bit Identifier (Full-CAN 2.0B passive) unterstützt. Die Auswahl des CAN-Treibers erfolgt über die Verwendung der jeweiligen Pins des

Moduls. Möglich ist die Verwendung des on-board CAN-Treibers (Pins 37, 38) oder eines externen optisch entkoppelten CAN-Treibers (Pins 36, 39).

### 3.5 RESET

Zum Auslösen eines automatischen power-on Reset befindet sich am Reset Eingang des Microcontrollers ein Kondensator von  $1\mu\text{F}$ . Dieser Kondensator wird bei power-on über einen Widerstand von  $50\text{ k}\Omega$  geladen und hält dadurch den Reset Eingang für ca. 50ms auf Low.

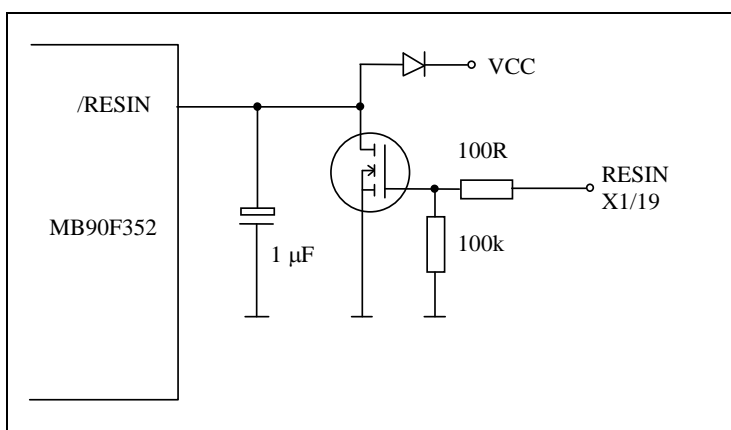


Bild 4: Prinzipieller Aufbau des Reset Eingangs auf dem CANopen ChipF40



### 3.6 Einstellung der Kommunikationsparameter

#### Einstellung Knotenadresse am DIP-Schalter:

Der CANopen ChipF40 besitzt die Basis-Knotenadresse 40H. Mit Hilfe der DIP-Schalter DIP1 bis DIP4 auf dem Modul kann diese Basis-Knotenadresse mit einem Offset im Bereich 0H bis 0EH beaufschlagt werden. Die absolute Knotenadresse ergibt sich wie folgt:

$$\text{NodeID} = 40\text{H} + \text{DIP1} * 2^0 + \text{DIP2} * 2^1 + \text{DIP3} * 2^2 + \text{DIP4} * 2^3$$

mit  $\text{DIP}_x=1$  bei geschlossenem Schalter (Stellung ON) und  $\text{DIP}_x=0$  bei offenem Schalter.

**Beispiel:** Bei geschlossenem Schalter DIP1 und DIP4 ergibt sich folgende Knotenadresse:

$$\text{NodeID} = 40\text{H} + 1 * 2^0 + 0 * 2^1 + 0 * 2^2 + 1 * 2^3$$

$$\text{NodeID} = 40\text{H} + 1 * 1 + 1 * 8$$

$$\text{NodeID} = 40\text{H} + 1 + 8$$

$$\text{NodeID} = 49\text{H}$$

Die Einstellung 0FH an DIP1...DIP4 ist zum Rücksetzen des CANopen ChipF40 in den Auslieferungszustand reserviert.

Die Einstellung der Knotenadresse läßt sich folgender Tabelle entnehmen:

Knotenadresse	DIP-Schalter			
	DIP1	DIP2	DIP3	DIP4
40H	OFF	OFF	OFF	OFF
41H	ON	OFF	OFF	OFF
42H	OFF	ON	OFF	OFF
43H	ON	ON	OFF	OFF
44H	OFF	OFF	ON	OFF
45H	ON	OFF	ON	OFF
46H	OFF	ON	ON	OFF
47H	ON	ON	ON	OFF
48H	OFF	OFF	OFF	ON
49H	ON	OFF	OFF	ON
4AH	OFF	ON	OFF	ON
4BH	ON	ON	OFF	ON
4CH	OFF	OFF	ON	ON
4DH	ON	OFF	ON	ON
4EH	OFF	ON	ON	ON
reserviert	ON	ON	ON	ON

### **Einstellung Knotenadresse am DIP-Schalter der Version 3301002:**

Der CANopen ChipF40 Version 3301002 besitzt die Basis-Knotenadresse 40H. Mit Hilfe der DIP-Schalter DIP1 bis DIP4 und DIP6 bis DIP7 auf dem Modul kann diese Basis-Knotenadresse mit einem Offset im Bereich 0H bis 3EH beaufschlagt werden. Die absolute Knotenadresse ergibt sich wie folgt:

$$\text{NodeID} = 40\text{H} + \text{DIP1} * 2^0 + \text{DIP2} * 2^1 + \text{DIP3} * 2^2 + \text{DIP4} * 2^3 + \text{DIP6} * 2^4 + \text{DIP7} * 2^5$$

mit  $\text{DIP}_x=1$  bei geschlossenem Schalter (Stellung ON) und  $\text{DIP}_x=0$  bei offenem Schalter.

Damit ergibt sich ein Wertebereich von 40H (64dez) bis 07EH (126dez).

Die Einstellung 03FH an DIP1...DIP4 und DIP6...DIP7 ist zum Rücksetzen des CANopen ChipF40 in den Auslieferungszustand reserviert.

### **Einstellung Knotenadresse über CANopen:**

Der CANopen ChipF40 kann mittels LSS-Dienst (Layer Setting Services and Protocol, CiA DSP 305) über CANopen eine beliebige Knotenadresse im Bereich 1...127 (01H..07FH) erhalten. Danach sind die Einstellungen einer Knotenadresse am DIP-Schalter nicht mehr relevant.

Die Umstellung der NodeID per LSS hat ein Restaurieren des OD mit den Default-Werten zur Folge (*s. Ende dieses Kapitels*).

### **Einstellung Bitrate am DIP-Schalter:**

Die Einstellung der Bitrate erfolgt mit Hilfe der DIP-Schalter DIP5 und DIP6 auf dem Modul. Folgende Bitraten sind möglich:

Bitrate kBit/s	DIP5	DIP6
20	ON	OFF
125	OFF	OFF
500	OFF	ON
1000	ON	ON

Tabelle 2: *Einstellung der Bitraten*

### **Einstellung Bitrate über CANopen:**

Der CANopen ChipF40 kann mittels LSS- Dienst (Layer Setting Service and Protocol, CiA DSP-305) über CANopen eine Bitrate erhalten. Danach sind die Einstellungen einer Bitrate am DIP-Schalter nicht mehr relevant.

Index in LSS Tabelle	Timing	Bitrate kBit/s
0		1000
1		800
2		500
3		250
4		125
5		100
6		50
7		20
8		10

Tabelle 3: *mögliche Bitraten, die mittels LSS eingestellt werden können*

### **Einstellung Konfiguration am DIP-Schalter:**

Die Einstellung der Konfiguration erfolgt mit Hilfe der DIP-Schalter DIP7 und DIP8 auf dem Modul. Folgende Konfigurationen sind möglich:

Konfiguration	DIP7	DIP8
0	OFF	OFF
1	ON	OFF
2	OFF	ON
3	ON	ON

Tabelle 4: *Einstellung der Konfiguration*

In der Version 3301002 des CANopen Chip F40 ist eine Einstellung der Konfiguration mittels DIP-Schalter nicht möglich. Die Default-Konfiguration ist 1.

### **Einstellung Konfiguration über CANopen:**

Der CANopen ChipF40 kann über das Object Dictionary über CANopen eine Konfiguration erhalten. Die Konfiguration kann über den herstellerspezifischen Eintrag 0x2000 eingestellt werden. Als Wert ist die Konfigurationsnummer einzutragen, z.B. 5 für Konfiguration 5. Danach sind die Einstellungen einer Konfiguration am DIP-Schalter nicht mehr relevant.

Die Umstellung der I/O-Konfiguration hat ein Restaurieren des OD mit den Default-Werten zur Folge.

### **Default-Einstellungen:**

Im Auslieferungszustand sind alle beschriebenen DIP-Schalter offen, so daß der CANopen ChipF40 folgende Einstellungen besitzt:

Knotenadresse = 40H

Bitrate = 125 kBit/s

Konfiguration = 0

## **3.7 Zuordnung der IO-Pins zur Konfiguration F40**

<b>Konfiguration</b>	<b>Digitale Eingänge</b>	<b>Digitale Ausgänge</b>	<b>Analoge Eingänge</b>	<b>PWM Ausgänge</b>
0	14	8	2	4
1	8	8	8	4
2	16	8	-	4
3	8	16	-	4
4	16	-	8	4
5	24	-	-	4
6	16	4	4	4

*Tabelle 5: Anzahl der Ein-/Ausgänge je nach Konfiguration F40*

Pin-Nr.	Konfig. 0	Konfig. 1	Konfig. 2	Konfig. 3	Konfig. 4	Konfig. 5	Konfig. 6
1	DO 4	DO 0	DO 0	DO 0	DI 12	DI 20	DI 0
2	DO 5	DO 1	DO 1	DO 1	DI 13	DI 21	DI 1
3	DO 6	DO 2	DO 2	DO 2	DI 14	DI 22	DO 0
4	DO 7	DO 3	DO 3	DO 3	DI 15	DI 23	DO 1
5							
6							
7	PWM 0	PWM 0	PWM 0	PWM 0	PWM 0	PWM 0	PWM 0
8	PWM 1	PWM 1	PWM 1	PWM 1	PWM 1	PWM 1	PWM 1
9							
10							
11	AI 0	AI 0	DI 0	DI 0	AI 0	DI 0	AI 0
12	AI 1	AI 1	DI 1	DI 1	AI 1	DI 1	AI 1
13	DI 0	AI 2	DI 2	DI 2	AI 2	DI 2	AI 2
14	DI 1	AI 3	DI 3	DI 3	AI 3	DI 3	AI 3
15	DI 2	AI 4	DI 4	DI 4	AI 4	DI 4	DI 2
16	DI 3	AI 5	DI 5	DI 5	AI 5	DI 5	DI 3
17	DI 4	AI 6	DI 6	DI 6	AI 6	DI 6	DI 4
18	DI 13	AI 7	DI 7	DI 7	AI 7	DI 7	DI 5
19							
20							
21	DI 12	DO 4	DO 4	DO 4	DI 8	DI 16	DI 14
22	DI 11	DO 5	DO 5	DO 5	DI 9	DI 17	DI 15
23	DI 10	DI 0	DI 8	DO 8	DI 0	DI 8	DI 6
24	DI 9	DI 1	DI 9	DO 9	DI 1	DI 9	DI 7
25	DI 8	DI 2	DI 10	DO 10	DI 2	DI 10	DI 8
26	DI 7	DI 3	DI 11	DO 11	DI 3	DI 11	DI 9
27	DO 3	DI 4	DI 12	DO 12	DI 4	DI 12	DI 10
28	DO 2	DI 5	DI 13	DO 13	DI 5	DI 13	DI 11
29	DO 1	DI 6	DI 14	DO 14	DI 6	DI 14	DI 12
30	DO 0	DI 7	DI 15	DO 15	DI 7	DI 15	DI 13
31	DI 6	DO 6	DO 6	DO 6	DI 10	DI 18	DO 2
32	DI 5	DO 7	DO 7	DO 7	DI 11	DI 19	DO 3
33	PWM 2	PWM 2	PWM 2	PWM 2	PWM 2	PWM 2	PWM 2
34	PWM 3	PWM 3	PWM 3	PWM 3	PWM 3	PWM 3	PWM 3
35							
36							
37							
38							
39							
40							

Tabelle 6: Zuordnung Ein-/Ausgänge zur Konfiguration F40

DI - Digitaler Eingang

DO - Digitaler Ausgang

AI - Analoger Eingang

PWM - PWM Ausgang

**Konfiguration 0 ist als default eingestellt.**

**Vor einer Umstellung der I/O-Konfiguration ist dafür zu sorgen, dass die 28 I/O-Pins des Moduls durch die Umstellung nicht fehlbeschaltet werden. Am besten werden die I/O-Pins von der Außenbeschaltung getrennt.**

### 3.8 Zuordnung der IO-Pins zur Konfiguration F40 V3

Konfiguration	Digitale Eingänge	Digitale Ausgänge	Analoge Eingänge
0	18	8	2
1	12	8	8
2	20	8	-
3	12	16	-
4	20	-	8
5	28	-	-
6	20	4	4

*Tabelle 7: Anzahl der Ein-/Ausgänge je nach Konfiguration F40 V3*

Pin-Nr.	Konfig. 0	Konfig. 1	Konfig. 2	Konfig. 3	Konfig. 4	Konfig. 5	Konfig. 6
1	DO 4	DO 0	DO 0	DO 0	DI 12	DI 20	DI 0
2	DO 5	DO 1	DO 1	DO 1	DI 13	DI 21	DI 1
3	DO 6	DO 2	DO 2	DO 2	DI 14	DI 22	DO 0
4	DO 7	DO 3	DO 3	DO 3	DI 15	DI 23	DO 1
5							
6							
7	DI 14	DI 8	DI 16	DI 8	DI 16	DI 24	DI 16
8	DI 15	DI 9	DI 17	DI 9	DI 17	DI 25	DI 17
9							
10							
11	AI 0	AI 0	DI 0	DI 0	AI 0	DI 0	AI 0
12	AI 1	AI 1	DI 1	DI 1	AI 1	DI 1	AI 1
13	DI 0	AI 2	DI 2	DI 2	AI 2	DI 2	AI 2
14	DI 1	AI 3	DI 3	DI 3	AI 3	DI 3	AI 3
15	DI 2	AI 4	DI 4	DI 4	AI 4	DI 4	DI 2
16	DI 3	AI 5	DI 5	DI 5	AI 5	DI 5	DI 3
17	DI 4	AI 6	DI 6	DI 6	AI 6	DI 6	DI 4
18	DI 13	AI 7	DI 7	DI 7	AI 7	DI 7	DI 5
19							
20							
21	DI 12	DO 4	DO 4	DO 4	DI 8	DI 16	DI 14
22	DI 11	DO 5	DO 5	DO 5	DI 9	DI 17	DI 15
23	DI 10	DI 0	DI 8	DO 8	DI 0	DI 8	DI 6
24	DI 9	DI 1	DI 9	DO 9	DI 1	DI 9	DI 7
25	DI 8	DI 2	DI 10	DO 10	DI 2	DI 10	DI 8
26	DI 7	DI 3	DI 11	DO 11	DI 3	DI 11	DI 9
27	DO 3	DI 4	DI 12	DO 12	DI 4	DI 12	DI 10
28	DO 2	DI 5	DI 13	DO 13	DI 5	DI 13	DI 11
29	DO 1	DI 6	DI 14	DO 14	DI 6	DI 14	DI 12
30	DO 0	DI 7	DI 15	DO 15	DI 7	DI 15	DI 13
31	DI 6	DO 6	DO 6	DO 6	DI 10	DI 18	DO 2
32	DI 5	DO 7	DO 7	DO 7	DI 11	DI 19	DO 3
33	DI 16	DI 10	DI 18	DI 10	DI 18	DI 26	DI 18
34	DI 17	DI 11	DI 19	DI 11	DI 19	DI 27	DI 19
35							
36							
37							
38							
39							
40							

Tabelle 8: Zuordnung Ein-/Ausgänge zur Konfiguration F40 V3

DI - Digitaler Eingang

DO - Digitaler Ausgang

AI - Analoger Eingang

**Konfiguration 0 ist als default eingestellt.**

**Vor einer Umstellung der I/O-Konfiguration ist dafür zu sorgen, dass die 24 I/O-Pins des Moduls durch die Umstellung nicht fehlbeschaltet werden. Am besten werden die I/O-Pins von der Außenbeschaltung getrennt.**

### 3.9 Technische Daten

#### Elektrische Parameter:

- Betriebsspannung: 5V  $\pm$ 10%
- Stromaufnahme: typ. 65mA, 24MHz CPU-Takt, 25°C;  
max. 140mA abhängig von der Beschaltung der I/Os und CAN
- Taktversorgung: 24MHz CPU-Takt durch Anpassung  
der externen 4MHz-Quarzfrequenz
  
- Ausgangsspannung IO-Leitungen:
  - L-Pegel: < 0.4V
  - H-Pegel: > 4.5V
- Ausgangsstrom IO-Leitungen:
  - L-Pegel 1 Pin Maximalstrom: 15mA
  - L-Pegel 1 Pin Nennstrom: 4mA
  - L-Pegel alle Pins Maximalstrom: 100mA
  - L-Pegel alle Pins Nennstrom: 50mA
  - H-Pegel 1 Pin Maximalstrom: -15mA
  - H-Pegel 1 Pin Nennstrom: -4mA
  - H-Pegel alle Pins Maximalstrom: -100mA
  - H-Pegel alle Pins Nennstrom: -50mA
- Eingangsspannung IO-Leitungen:
  - L-Pegel: > VSS – 0.3V
  - H-Pegel: > 0.8V • VCC; < VCC + 0.3V
- Analoge Eingänge:
  - Eingangsspannung: VAGND ... VREF
  - Auflösung: 10 Bit
  - Eingangskapazität: 10.7pF
  - Referenzspannung: VAGND + 2.7V ... VCC
- Eigenschaften on-board CAN-Treiber:
  - max. Bitrate: 1000 kBit/s
  - Knotenanzahl: <110

#### Umgebungsbedingungen:

- Betriebstemperaturbereich: -40°C bis +85°C
- Lagertemperaturbereich: -40°C bis +90°C
- Luftfeuchtebereich: max. 90% r.F. nicht  
kondensierend



**Mechanische Eigenschaften:**

- Modulgröße LxBxH: 58,7mm x 24,0 mm x 11,8 ±0,3 mm
- Gewicht: ca. 10,5g
- Kontaktierung: 40poliger Dual-Inline-IC-Sockel,  
RM 2,54 x 15,24
- Stiftleisten Rastermaß 2,54 mm, Ø0,47 mm, Kontaktlänge  
3,2 mm

## 4 Inbetriebnahme

### 4.1 Versorgungsspannung

Der CANopen ChipF40 benötigt eine Versorgungsspannung von +5V DC  $\pm 10\%$ . Der Anschluß erfolgt über die Pins 6 (GND) und Pin40 (+5V) nach dem Standard für TTL-Bausteine. Zusätzlich ist eine GND-Anbindung an den Pins 20 und 35 vorgesehen. Zur besseren Montage des CANopen ChipF40 sollte ein DIL40 Präzisionssockel als Adapter verwendet werden.

### 4.2 CAN-Schnittstelle

Der CANopen ChipF40 verfügt über einen Microcontroller Fujitsu MB90F352 mit integrierter FULL-CAN-Schnittstelle und on-board CAN-Treiber. Die Leitungen "HI" (CAN\_HIGH) sowie "LO" (CAN\_LOW) sind auf die Stiftleisten des Bausteins gelegt. Dabei gilt folgende Zuordnung :

CAN_HIGH	→ PIN 38
CAN_LOW	→ PIN 37
CAN_GND	→ PIN 6, 20 oder 35

CAN\_GND und GND (Schaltungsmasse 0V) sind dabei identisch.

Diese Leitungen können laut Standard auf einen SUB-D9 Stecker angeschlossen werden :

Pin am Baustein	Leitung	Pin an SUB-D9
PIN38	→CAN_HIGH	→ PIN 7
PIN37	→CAN_LOW	→ PIN 2
PIN6, 20 oder 35	→CAN_GND	→ PIN 6 und/oder PIN 3

*(Standard nach CiA DS 301 for Communication Profile)*

Es besteht aber auch die Möglichkeit, die CAN-Leitungen extern optisch zu entkoppeln.

Die max. Übertragungsrate beträgt 1MBit/s. Dabei sollte die maximale Busauslastung 50% nicht überschreiten.

Damit ist der CANopen ChipF40 hardwareseitig in Betrieb genommen.

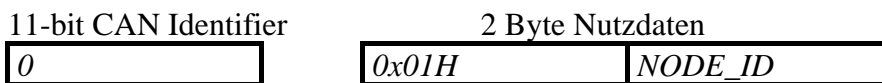


## 5 Schnelleinstieg

Dieses Kapitel ist für die Anwender gedacht, die mit CANopen bereits erste Erfahrungen gesammelt haben. Für Anwender, die die Anwendung des CANopen-Protokoll noch nicht kennen, sei auf das *Kapitel 7* verwiesen.

### 5.1 Start der Baugruppe (Boot-Up)

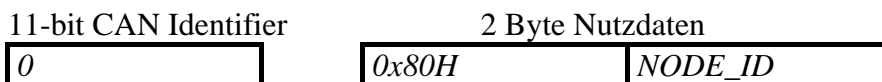
Da für alle Einträge im Objektverzeichnis (Object Dictionary, OD) Default-Werte vorhanden sind, kann in den meisten Fällen auf eine Konfiguration verzichtet werden. Die Baugruppe unterstützt das CANopen Minimum Boot-Up. Nach einem Reset und der Initialisierungsphase befindet sich die Baugruppe im Status "PRE-OPERATIONAL" (*siehe 8.3 PRE-OPERATIONAL*). Die Baugruppe läßt sich mit einem einzigen Telegramm (*Start\_Remote\_Node*) starten und geht damit in den Zustand "OPERATIONAL" (*siehe Kapitel8.4 OPERATIONAL*) über.



Dabei enthält das erste Nutzbyte das Startkommando, das zweite Datenbyte die Knotenadresse (*NODE\_ID*). Knotenadresse = 0 gilt für alle Knoten im Netzwerk (Broadcast).

### 5.2 Stop der Baugruppe

Die Prozessdaten-Kommunikation der Knoten kann mit dem Telegramm *Enter\_Pre\_Operational\_State* gestoppt werden. Das Telegramm hat folgenden Aufbau:



Mit *NODE\_ID* wird die Knotenadresse bezeichnet. *NODE\_ID* = 0 adressiert alle Baugruppen (Broadcast).

Die Baugruppe geht in den Status "PRE-OPERATIONAL" über. SDO-Transfer ist weiterhin möglich. Alle Parameter werden im aktuellen Zustand eingefroren.

**Hinweis :**

Konfigurationswerkzeuge für CANopen (z.B. ProCANopen der Firma Vector; CDM und CCM der Firma Port) arbeiten immer unter Verwendung des SDO-Transfers. Aufgrund dieser Tatsache funktionieren diese Tools auch im Zustand "Pre-Operational".

### 5.3 CAN-Identifizier / -Nachrichten

Für jede CAN-Nachricht mit Prozeßdaten (Prozess-Daten-Object, PDO) wird ein spezifischer CAN-Identifizier laut **CiA Draft Standard 301 V4.x** vergeben.

Die CAN-Identifizier für Ein- und Ausgangsdaten werden aus der Knotenadresse abgeleitet.

CAN-Identifizier (hex)	Datentyp
180H + Knotenadresse	1. Tx PDO
280H + Knotenadresse	2. Tx PDO
380H + Knotenadresse	3. Tx PDO
480H + Knotenadresse	4. Tx PDO
200H + Knotenadresse	1. Rx PDO
300H + Knotenadresse	2. Rx PDO
400H + Knotenadresse	3. Rx PDO (nicht F40 V3)
500H + Knotenadresse	4. Rx PDO (nicht F40 V3)

Tabelle 9: Zuordnung der CAN-ID zu den Prozeßdaten

### 5.4 Mapping der I/O's

Das defaultmässige Mapping der I/O's ist abhängig von der gewählten I/O-Konfiguration (siehe Tabelle 8: Zuordnung Ein-/Ausgänge zur Konfiguration).

Der CANopen ChipF40 unterstützt variables PDO-Mapping. Damit ist ein freies Mapping der Eingänge auf die Tx-PDOs und der Ausgänge auf die Rx-PDOs möglich. Diese Einstellungen können durch Schreiben auf den Index [1010H] auf dem Modul im EEPROM abgespeichert werden.

### 5.4.1 Default-Mapping CANopen ChipF40

Im Default-Mapping ist das 4. Tx PDO, sowie das 2. Rx PDO und das 4. Rx PDO ungültig. Daraus ergibt sich folgende Zuordnung von I/O und PDO:

Konfig.	1. Tx PDO	2. Tx PDO	3. Tx PDO	1. Rx PDO	3. Rx PDO
0	DI 0 ... DI 7 DI 8 ... DI 13	AI 0 ... AI 1	ungültig	DO 0 ... DO 7	Tastverhältnis PWM 0 ... PWM 3
1	DI 0 ... DI 7	AI 0 ... AI 3	AI 4 ... AI 7	DO 0 ... DO 7	Tastverhältnis PWM 0 ... PWM 3
2	DI 0 ... DI 7 DI 8 ... DI 15	ungültig	ungültig	DO 0 ... DO 7	Tastverhältnis PWM 0 ... PWM 3
3	DI 0 ... DI 7	ungültig	ungültig	DO 0 ... DO 7 DO 8 ... DO 15	Tastverhältnis PWM 0 ... PWM 3
4	DI 0 ... DI 7 DI 8 ... DI 15	AI 0 ... AI 3	AI 4 ... AI 7	ungültig	Tastverhältnis PWM 0 ... PWM 3
5	DI 0 ... DI 7 DI 8 ... DI 15 DI 16 ... DI 23	ungültig	ungültig	ungültig	Tastverhältnis PWM 0 ... PWM 3
6	DI 0 ... DI 7 DI 8 ... DI 15	AI 0 ... AI 3	ungültig	DO 0 ... DO 3	Tastverhältnis PWM 0 ... PWM 3

Tabelle 10: Mapping der I/O's F40

### 5.4.2 Default-Mapping CANopen ChipF40 V3

Im Default-Mapping ist das 4. Tx PDO und das 2. Rx PDO ungültig. Daraus ergibt sich folgende Zuordnung von I/O und PDO:

Konfiguration	1. Tx PDO	2. Tx PDO	3. Tx PDO	1. Rx PDO
0	DI 0 ... DI 7 DI 8 ... DI 13	AI 0 ... AI 1	ungültig	DO 0 ... DO 7
1	DI 0 ... DI 7	AI 0 ... AI 3	AI 4 ... AI 7	DO 0 ... DO 7
2	DI 0 ... DI 7 DI 8 ... DI 15	ungültig	ungültig	DO 0 ... DO 7
3	DI 0 ... DI 7	ungültig	ungültig	DO 0 ... DO 7 DO 8 ... DO 15
4	DI 0 ... DI 7 DI 8 ... DI 15	AI 0 ... AI 3	AI 4 ... AI 7	ungültig
5	DI 0 ... DI 7 DI 8 ... DI 15 DI 16 ... DI 23	ungültig	ungültig	ungültig
6	DI 0 ... DI 7 DI 8 ... DI 15	AI 0 ... AI 3	ungültig	DO 0 ... DO 3

Tabelle 11: Mapping der I/O's F40 V3

## 5.5 Baugruppenreset

Nach jedem Baugruppenreset sendet die Baugruppe ein Bootup-Telegramm. Temporäre Ausfälle der Baugruppe während des Betriebes und nachträgliches Einschalten der Baugruppe werden damit auch ohne Node-Guarding (*siehe Kapitel 7.8.1*) erkannt. Der Sender kann über den Identifier der Nachricht ermittelt werden.

11-bit CAN Identifier	1 Byte Nutzdaten
$700H + NODE\_ID$	0

## 5.6 Node Guarding

Für die Ausfallüberwachung des CANopen Netzwerkes stehen *Node Guarding* und *Life Guarding* Mechanismen zur Verfügung. Die dezentralen Geräte werden mittels Node Guarding überwacht, die ihrerseits über Life Guarding den Ausfall des Guardingmasters erkennen können.

Für das *Node Guarding* fordert der NMT-Master zyklisch ein Statustelegamm vom NMT-Slave (CANopen Geräte, z.B. CANopen ChipF40) an. Die Anforderung des Knotenstatus erfolgt mit einem Remote-Frame und enthält keine Nutzdaten. Das RTR-Bit (Remote-Transmit-Request-Bit) ist gesetzt.

11-bit CAN Identifier
$700H + NODE\_ID$

Daraufhin sendet der NMT-Slave das Statustelegamm, welches 1 Byte Nutzdaten enthält.

11-bit CAN Identifier	1 Byte Nutzdaten
$700H + NODE\_ID$	Guarding Protocol

Die Nutzdaten enthalten ein Toggle Bit, das nach jeder Nachricht wechseln muß, sowie den NMT-Zustand des Slave. Falls NMT-Zustand und Toggle Bit nicht mit den vom Master erwarteten Werten übereinstimmen oder falls keine Antwort erfolgt, geht der Master von einem Fehler des Gerätes aus. Wenn der Master die Guard Meldung zyklisch anfordert, kann der Slave den Ausfall des Masters daran erkennen, daß der Slave innerhalb der eingestellten *Life-Time* keine

Nachrichtenanforderung des Masters erhalten hat. Der Slave geht von einem Ausfall des Masters aus, setzt seine Ausgänge in den Fehlerzustand, sendet ein Emergency-Telegramm und geht in den Status "PRE-OPERATIONAL" über (Voraussetzung Index [67FEH] bzw. [1029H]=0).

Der *Life Time Factor* steht im Objekt [100DH] und wird mit dem Objekt [100CH] der *Guard Time* multipliziert und ergibt so die Lebenszeit für das "Node Guard Protokoll". Die Zeitbasis beträgt 1ms.

Die *Guard Time* sagt aus, wieviel Zeit zwischen 2 *Node Guarding* Telegrammen vergehen muss.

Der *Life Time Factor* sagt aus, wie oft die *Guard Time* ablaufen kann bevor ein Fehler generiert wird.

Default Einstellungen:

Life Time Factor	0
Guard Time	0 ms
Life Time	0 s

Beispieleinstellungen:

Life Time Factor	3
Guard Time	1000 ms
Life Time	3 s





## 6 Controller Area Network CAN

### 6.1 Kommunikation mit CANopen

CANopen ist eine vom CiA autorisierte Profildfamilie für industrielle Kommunikation mit interoperabel arbeitenden Automatisierungsgeräten auf der Basis von CAN.

Die Familie besteht durch die konsequente Umsetzung des ISO/OSI-Schichtenmodells.

CANopen bietet zahlreiche Vorteile, wie:

- herstellerneutrale Standards,
- offene Struktur,
- echtzeitfähige Kommunikation für Prozeßdaten ohne Protokolloverhead,
- modulare, skalierbare Struktur entsprechend den Anforderungen von einfachsten bis hin zu komplexen Automatisierungsgeräten.

CANopen bietet u.a. folgende Möglichkeiten

- Autokonfiguration von CAN-Netzwerken,
- einheitlichen Zugriff auf Geräteparameter,
- zyklische und ereignisgesteuerte Kommunikation,
- Synchronisation von Geräten vor allem für Mehrfachssysteme.

Durch die internationale Anwender- und Herstellervereinigung "CAN in Automation e.V." (CiA) wurde neben der physikalischen CAN-Schnittstelle auch das Schicht-7-Protokoll CANopen als herstellerübergreifender Standard im industriellen Steuerungsbereich für CAN definiert.

## 6.2 CANopen - offene, industrielle Kommunikation

CANopen basiert auf dem Kommunikationsprofil CiA DS 301 V4.x. CANopen wurde in verschiedenen Special Interest Groups (SIG) z.B.

- SIG Distributed I/O
- SIG Motion Control

und der Working Group (WG)

- WG Higher Layer Protocols

entwickelt.

Ausgangspunkt für den Standardentwurf CiA DS 301 waren Ergebnisse des ESPRIT-Projektes ASPIC.

Das Kommunikationsprofil beschreibt detailliert, wie mit Hilfe der von CANopen bereitgestellten Funktionalität Daten über den Bus gesendet werden.

Dabei erfolgt eine Einteilung der Daten in

1. Prozeßdaten und
2. Servicedaten

Prozeßdaten sind dabei Echtzeitdaten, wie Soll- und Istwerte, die ein Gerät ausgeben oder liefern soll. Sie werden jeweils über ein sogenanntes Prozeß-Daten-Objekt (PDO) übertragen. Dabei wird im Kommunikationsprofil festgelegt, wie ein solches PDO angelegt wird und welches Protokoll bei der Übertragung zu verwenden ist. PDO's können von mehreren Geräten gleichzeitig genutzt werden. Dadurch ist es möglich, auch Broadcast-Betrieb zu realisieren.

Mit Servicedaten kann ein Gerät konfiguriert und parametrierbar werden. Sie korrespondieren direkt mit Einträgen im sogenannten Objektverzeichnis eines Gerätes. Servicedaten werden über sogenannte Service-Daten-Objekte (SDO) übertragen. Das Kommunikationsprofil legt dabei ebenfalls fest, wie diese Objekte angelegt werden und welche Dienste ausführbar sind.

Ein SDO kann nur zwischen zwei Teilnehmern ausgebildet werden. Üblicherweise handelt es sich dabei um einen Konfigurationsmaster und um ein zu konfigurierendes Gerät. Der SDO-Transfer ermöglicht Quittungsbetrieb.

Jedes Gerät kann über mehrere PDO und SDO verfügen. Dadurch ist es möglich, neben reinen Master/Slave-Anwendungen auch Netzwerke zu konzipieren, in denen mehrere Master existieren.

Neben den Datenklassen definiert CANopen Kommunikationsklassen, mit denen

- Synchronisationsverhalten,
- Ereignisverarbeitung,
- Initialisierungsverhalten u.a.m.

beschrieben werden kann.

Neben der Spezifikation der Kommunikation enthalten die Standards Definitionen von sogenannten Device Profiles, die das Verhalten von Geräten beschreiben.

Diese Profiles bestehen prinzipiell aus zwei Beschreibungskomponenten.

#### 1. Funktionale Beschreibung

Die Funktionalität eines Gerätes wird durch Funktionsblöcke und Datenflüsse beschrieben. Die Parameter sind in einem sogenannten Objektverzeichnis angeordnet. Das Objektverzeichnis hat ein vordefiniertes Aussehen, das heißt, die Parameter für Geräte eines bestimmten Typs (z.B. E/A-Module oder Drives) befinden sich immer an der gleichen Stelle im Objektverzeichnis. Die Einträge sind als Mandatory-, Optional- und Manufacturer-Specific-Data klassifizierbar.

#### 2. Beschreibung des Betriebsverhaltens

Das Betriebsverhalten des Gerätes wird durch ein Zustandsübergangsdiagramm (*siehe Bild 5*) beschrieben.

Auswahl einiger Geräteprofile, die bisher standardisiert wurden:

- I/O Modules CiA DSP 401
  - digitale I/O's
  - analoge I/O's
- Drives and Motion Control CiA DSP 402
  - Servoantriebe,
  - Schrittmotoren und
  - Frequenzumrichter
- Encoder CiA DSP 406

Der aktuelle Stand der standardisierten Geräteprofile kann auf der Homepage des CiA eingesehen werden: <http://www.can-cia.de> .

Alle Profile orientieren sich am DRIVECOM-Profil, wobei CAN-spezifische Anpassungen die Multimasterfähigkeit von CAN besonders berücksichtigen.

Die Software für die CANopen-Slave Funktionalität basiert auf den im DS 301 definierten Diensten für den Datenaustausch und Diensten für das Netzwerkmanagement.

### 6.3 Aufbau und Verdrahtung

Als CAN-Bus-Kabel wird eine verdrehte, geschirmte Zweidrahtleitung eingesetzt, die an beiden Enden mit  $120\Omega$  abgeschlossen wird. Der Schirm ist einseitig auf PE zu legen. Der Wellenwiderstand des Kabels soll  $120\Omega$  betragen. Der Querschnitt richtet sich nach der Leitungslänge.

Leitungslänge [m]	Max. Bitrate [kBit/s]	Spezifischer Widerstand [m $\Omega$ /m]	Leitungsquerschnitt [mm <sup>2</sup> ]
0 ... 40	1000	70	0,25 ... 0,34
40 ...100	500	<60	0,34 ... 0,60
100 ... 500	100	<40	0,50 ... 0,60
500 ...1000	20	<26	0,75 ... 0,80

Tabelle 12: Empfohlene Leitungsparameter

## 7 CANopen-Kommunikation

### 7.1 Grundlagen

Mit der Entwicklung und Standardisierung der CANopen Profile wird dem Feldbusanwender möglich, CAN-Netzwerke in einfachster Art und Weise aufzubauen und dabei aus einer Vielzahl von Komponenten und Baugruppen unterschiedlicher Hersteller auszuwählen.

Realisierbar wird dies durch die herstellerübergreifende Standardisierung der Geräteprofile. Ein Geräteprofil beschreibt ein Standardgerät mit seiner Grundfunktionalität für ein Gerät seiner Klasse.

Diese grundlegende Funktionalität ist Voraussetzung für einfache, herstellerunabhängige Funktionen des Gerätes. Das Konzept der Gerätestandardisierung wird durch die Möglichkeit der Erweiterung um optionale Funktionalität ergänzt. Sie muß nicht von allen Herstellern implementiert werden, aber wenn sie realisiert wird, muß sie den Richtlinien des Geräteprofiles folgen.

Das Projektdesign, der Netzwerkaufbau und -test wird durch die einheitlichen Netzwerkparameter auf jedem Gerät stark vereinfacht. Leistungsfähige Netzwerkkonfiguratoren können den Anwender bei der Realisierung der Applikation unterstützen, indem sie zum Beispiel die Zuordnung der Netzwerkvariablen zu Prozeßdatenobjekten (PDO) und deren Konfiguration automatisieren und die konfigurierten Daten der Geräte archivieren.

### 7.2 Geräteprofile

Es existieren Profile für die Kommunikation CiA Draft Standard 301 V4.x, für I/O-Module CiA Draft Standard 401 V2.x, für Drives and Motion Control CiA Draft Standard 402 V2.x, für Encoder CiA Draft Standard 406. Andere Profile sind in Vorbereitung.

Die Profile eines Gerätes sind in Form eines Objektverzeichnisses (OD) in einer definierten Form im Gerät selbst hinterlegt. Das Objektverzeichnis verwaltet die Objekte unter einem 16-bit Index, der sich mit einem 8-bit Subindex weiter unterteilen läßt. Die Einträge sind zu Gruppen zusammengefaßt. Das Kommunikationsprofil liegt zum Beispiel ab Index [1000H] bis [1FFFH].

---

Eine Reihe von Einträgen sind Pflichteinträge (mandatory), andere sind optional oder herstellerspezifisch. Man unterscheidet verschiedene Typen von Objekten:

- Domain eine variable Anzahl von Daten
- Deftype ein Definitionseintrag z.B. unsigned16
- Defstruct ein Record-type wie z.B. PDOMapping
- Var eine einzelne Variable
- Array ein mehrfaches Datenfeld, wobei jedes einzelne Datenfeld eine einfache Variable vom gleichen Typ ist
- Record ein mehrfaches Datenfeld, wobei die Datenfelder eine beliebige Kombination von einfachen Variablen sind.

Bei strukturierten Einträgen steht auf Subindex 0 die Anzahl der noch folgenden Subindexe.

### 7.3 Kommunikationsprofil

Durch ein einheitliches Kommunikationsprofil ist die Schnittstelle zwischen Applikation und Gerät klar definiert. Das Kommunikationsprofil beschreibt die verschiedenen Kommunikationsobjekte und Dienste und die verfügbaren Modi der Sendenachrichtenauslösung. Das Kommunikationsmodell unterstützt die Übertragung von synchronen und asynchronen Sendungen. Mit den Mitteln der synchronen Nachrichtenübertragung ist der netzwerkweite, koordinierte Datenaustausch möglich. Die synchrone Übertragung wird mit dem vordefinierten Objekt SYNC-Message initiiert. Asynchrone Nachrichten können zu jeder Zeit übertragen werden.

### 7.4 Servicedatenobjekte

Das Netzwerkmanagement greift auf die Kommunikations- und Geräteprofile der Netzwerkteilnehmer zu. Für diese Zugriffe stehen Servicedatenobjekte (SDO) zur Verfügung. Die SDO stellen eine Punkt-zu-Punkt Verbindung zum Zugriff auf das Objektverzeichnis des Netzwerkknosens dar. Der Zugriff geschieht nach dem Multiplexed Domain Protokoll, wobei der Index und Subindex des

adressierten Objektes als Multiplexor verwendet werden. Dieses Protokoll beruht auf Quittungsbetrieb. Nachrichten, die kürzer als 5 Byte sind, können mit einer Sendung/Quittung übertragen werden. Der Inhaber des Objektverzeichnisses ist dabei der SDO-Server. Die Lese- und Schreibzugriffe über SDO's werden vom CANopen-Gerät überwacht und auf Gültigkeit überprüft. Es existieren eine Reihe von Zugriffsbeschränkungen, wie *read only*, *write only*, *no pdo mapping*. Fehlermeldungen geben detaillierte Auskunft über Zugriffskonflikte. SDO's dienen hauptsächlich Konfigurationszwecken.

## 7.5 Prozeßdatenobjekte

Der Datenaustausch benötigt keinen Busmaster. Der Prozeßdatenaustausch in Echtzeit zwischen den Knoten erfolgt über Prozeßdatenobjekte (PDO) direkt und ohne Protokolloverhead. Ein PDO ist eine CAN-Nachricht, deren Dateninhalt, Identifier, Sperrzeit, Übertragungstyp und Eventtimer über Einträge im Objektverzeichnis mittels SDO's konfigurierbar ist. Die Einträge befinden sich ab Index [1400H] für Empfangsobjekte und ab [1800H] für Sendeobjekte. CANopen erlaubt zyklische und ereignisgesteuerte Kommunikation. Der Übertragungstyp gibt die Art und Weise der Reaktion auf die SYNC-Nachricht an, die Sperrzeit (Inhibittime) ist die minimale Zeit, die zwischen zwei Sendungen des PDO verstreichen muß. Dadurch ist es möglich, die Buslast auf ein Minimum zu reduzieren und einen hohen Datendurchsatz zu erzielen. Der Eventtimer wird dann eingesetzt, wenn ein Gerät ein PDO regelmäßig in gleichbleibenden Abständen senden soll. Die Inhibittime wird bei dieser Übertragungsart nicht berücksichtigt. Weiterhin besteht die Möglichkeit, PDO's über Remote-Frames abzufragen. Ein einfaches Gerät unterstützt normalerweise vier PDO's. Diese sind mit voreingestellten Identifiern initialisiert. Weitere PDO's können auf dem Gerät vorgesehen, aber zur Vermeidung von Konflikten ungültig gesetzt sein. Dies geschieht mit Setzen des MSB (Bit 31) im Identifier des PDO.

Der Identifier ist im Objektverzeichnis unter dem Eintrag Kommunikationsparameter, Subindex 1 zu erreichen. Bit 30 gibt bei Wert 0 an, daß Remote-Request für dieses PDO erlaubt sind. Bit 29 auf 0 steht für 11-Bit Identifier.



Bit	31	30	29	28 - 11	10 - 0
11-bit-ID	0/1	0/1	0	0000000000000000 0000	11-bit Identifizier
29-bit-ID	0/1	0/1	1	29-bit Identifizier	

Tabelle 13: Aufbau der COB-Identifizier

Die Übertragungstypen (transmission type) auf Subindex 2 können im Bereich von 0 bis 255 eingestellt werden. Die Werte 0 bis 240 bedeuten, daß die Übertragung der PDO in Relation zur SYNC-Nachricht steht. 0 bedeutet, daß abgetastete Eingangswerte bei Eintreffen des SYNC nur bei Veränderung gesendet werden, zwischen 1 und 240 gilt, daß das PDO nach dem Eintreffen der entsprechenden Anzahl von SYNC gesendet wird. Die Werte 241 bis 251 sind reserviert. Die Typen 252 und 253 sind nur für Remote-Objekte vorgesehen. Bei 253 werden die Daten bei Eintreffen des Remote-Request aktualisiert. 254 und 255 stehen für asynchrone PDO, deren Auslösung hersteller- oder geräteprofilspezifisch ist.

Die Sperrzeit (Inhibittime) wird in 100µs - Schritten als unsigned16-Wert auf Subindex 3 hinterlegt.

Auf Subindex 4 steht die CMS- Prioritätsgruppe. Dieser Mechanismus ist mit der Einführung des CiA DS301 V4.x entfallen. Der Subindex 4 wird aus Kompatibilitätsgründen erhalten und ist als reserviert gekennzeichnet. Auf die Funktion hat der Eintrag keinen Einfluß.

Subindex 5 enthält den Eventtimer. Dieser wird in 1ms Schritten als unsigned16-Wert hinterlegt.

In Abhängigkeit der unterstützten Subindexe muß Subindex 0 auf den entsprechenden Wert (5) gesetzt werden.

Für PDO's gelten folgende Regeln im I/O Profil:

Das erste Sende- und Empfangs-PDO wird für den Austausch digitaler Daten verwendet, das zweite Sende- und Empfangs-PDO für den Austausch analoger Daten. Wenn ein Gerät keine digitalen In- oder Outputs unterstützt, soll das erste Sende- oder Empfangs-PDO unbenutzt bleiben. Bei Geräten ohne analoge Interfacebaugruppen soll das zweite Sende- und Empfangs-PDO nicht verwendet werden.

## 7.6 PDO-Mapping

Zu jedem Kommunikationsparametereintrag eines PDO gehört ein Mappingeintrag, der im Objektverzeichnis 200H höher angesiedelt ist. Diese Mappingtabelle korrespondiert mit dem Inhalt der Daten des PDO. Grundlage für das Mapping ist, daß sich Variablen im Objektverzeichnis befinden, die mappbar sind, zum Beispiel digitale Eingänge auf Index [6000H] und digitale Ausgänge auf Index [6200H].

Diese Werte können natürlich über SDO gesetzt und ausgelesen werden. Um die Vorteile des CAN-Busses auszunutzen, werden die Variablen in PDO abgebildet.

Das geschieht folgendermaßen:

Die Einträge in der Mappingtabelle sind vier Byte groß. Auf Subindex 0 wird die Anzahl der zu mappenden Objekte geschrieben. Auf jeden folgenden Subindex wird ein Verweis auf den Index und Subindex der Variable und deren Länge in Bit abgelegt, zum Beispiel 60000108H für einen Verweis auf Index [6000H], Subindex 1, Länge 8 Bit. In diesem Fall wird der Wert des digitalen Einganges auf das erste Byte eines Sende-PDO's abgebildet. Das Mapping erfolgt bei den meisten Geräten mit einer Granularität von 8, das heißt, es sind maximal 8 Einträge zu einem Byte in der Mappingtabelle möglich.

Manchmal ist es sinnvoll, Bereiche aus dem Mapping auszuschließen. Ein Gerät soll zum Beispiel nur das 5. Byte eines PDO auswerten. In diesem Fall können 2 unsigned16 Dummy-Objekte im Mapping eingefügt werden, sofern vom Gerät unterstützt. Mit Hilfe der Mappingtabelle werden nun entsprechend der eingestellten Kommunikationsparameter das zu sendende PDO kodiert oder das empfangene PDO dekodiert.

## 7.7 Fehlerverarbeitung

Jeder Knoten im Netz ist in der Lage, Fehlerzustände zu signalisieren, soweit sie von der Hardware und Software erkannt werden. Dafür wird das Emergency-Objekt verwendet. Interne fatale Fehlerzustände werden in Fehlercodes codiert und nur einmal an die anderen Knoten geschickt. Treten weitere, andere Fehler auf, verbleibt der Knoten im Fehlerzustand und sendet ein neues Emergency-Objekt. Wenn der Fehler behoben ist, sendet der Knoten eine Fehlermeldung mit dem Code "No Error". Die Emergency-Message bestehen aus 8 Byte, wobei das erste und zweite Byte eine zusätzliche Information, die im Geräteprofile zu finden ist, das dritte Byte den Inhalt des Fehlerregisters ist und die restlichen 5 Byte herstellerspezifisch sind. Der Emergency Error Code wird in Objekt [1003H], dem *Pre-Defined-Error-Field* abgelegt. Dieser stellt ein Fehlerlogbuch dar, die Fehler werden zeitlich sortiert. Der älteste Fehler liegt auf dem höchsten Subindex.

Byte	0	1	2	3	4	5	6	7
Content	Emergency Error Code		Error Register, Object [1001H]	Manufacturer Specific Error Field				

Tabelle 14: Aufbau der Emergency-Nachricht

## 7.8 Netzwerkdienste

Neben Diensten für die Konfiguration und dem Datenaustausch existieren eine Reihe von Netzwerkdiensten zur Überwachung der Netzwerkteilnehmer. NMT-Dienste (Network Management) erfordern einen Knoten im Netz, welcher die Aufgaben des NMT-Masters übernimmt. Dazu gehört unter anderem die Initialisierung der NMT-Slaves, die Verteilung der Identifier, die Knotenüberwachung und das Booten des Netzes.

### 7.8.1 Life Guarding

Die optionale Knotenüberwachung wird durch das sogenannte *Life Guarding* erreicht. Der NMT-Master sendet zyklisch eine Lifeguard-Nachricht an das Gerät. Dieses antwortet darauf mit einer Nachricht, worin sein derzeitiger NMT-Zustand und ein zwischen zwei Nachrichten wechselndes Bit enthalten sind. Bei Ausbleiben der Antwort oder unerwartetem NMT-Zustand des Gerätes wird die NMT-Master Applikation informiert. Weiterhin kann das Gerät den Ausfall des Masters detektieren. Das *Life Guarding* wird mit dem Aussenden der ersten Sendung des Masters gestartet.

### 7.8.2 Heartbeat

Das Heartbeat ist, analog zum Life Guarding, ein Überwachungsdienst, für den jedoch kein NMT-Master benötigt wird. Die Aufgaben von Producer und Consumer können von allen CANopen Geräten ausgeführt werden, die diesen Dienst unterstützen.

#### 7.8.2.1 Heartbeat Producer

Der Heartbeat Producer sendet zyklisch eine Heartbeat Nachricht. Als Intervallzeit wird die auf Index [1017H] eingestellte Producer Heartbeat Time (unsigned 16 - Zeitwert in ms) verwendet. Ist diese Zeit abgelaufen, wird eine Nachricht folgenden Aufbaus gesendet:

11-bit CAN Identifier	1 Byte Nutzdaten
<span style="border: 1px solid black; padding: 2px;"><i>700H+NodeID</i></span>	<span style="border: 1px solid black; padding: 2px;"><i>Producer state</i></span>

*Tabelle 15 : Aufbau der Heartbeat-Nachricht*

Die verwendete COB-ID ist die 0700H + Knotennummer.

Der Heartbeat-Producer gibt im ersten Byte der Nachricht seinen NMT-Status (producer state) an. Dieser kann folgende Werte annehmen:

00H BOOTUP  
04h STOPPED  
05h OPERATIONAL  
7Fh PRE-OPERATIONAL

---

Der Heartbeat-Producer wird deaktiviert, wenn als Producer Heartbeat Time der Wert Null eingetragen wird.

### 7.8.2.2 Heartbeat Consumer

Der Heartbeat-Consumer wertet die vom Producer gesendete Heartbeat-Nachricht aus. Um den Producer zu überwachen, benötigt der Consumer zu jedem Producer, dessen Heartbeat-Nachrichten ausgewertet werden sollen, die Knotennummer sowie die Consumer Heartbeat Time.

Für jeden zu überwachenden Producer gibt es einen Untereintrag, der folgendermaßen aufgebaut ist:

	MSB		LSB
Bit	31-24	23-16	15-0
Value	00H	Node-ID	Consumer Heartbeat Time

*Tabelle 16 : Aufbau des Consumer Heartbeat Time – Eintrages*

Die Aktivierung des Consumers erfolgt, wenn eine Heartbeat-Nachricht empfangen wird und ein entsprechender Eintrag im OD vorhanden ist. Läuft bei aktivem Heartbeat-Consumer eine der aktivierten Heartbeat-Zeiten ab, ohne dass eine entsprechende Heartbeat-Nachricht empfangen wurde, wird der Consumer für diesen Producer deaktiviert und ein Event ausgelöst. Die Auswertung des Event ist vom Objekt [67FEH] (F40 V3) bzw. [1029H] (F40) "Error Behaviour" abhängig (*siehe Abschnitt 9.1*).

Der Heartbeat-Consumer wird komplett deaktiviert, wenn als erste Consumer Heartbeat Time der Wert Null eingetragen wird.

## 7.9 Netzwerk Book-Up

Der NMT-Master ist verantwortlich für das Booten des Netzwerkes. Das Booten erfolgt in mehreren Schritten. In Abhängigkeit von dem Typ der angeschlossenen Geräte werden die Identifier bei Minimum-Geräten über vordefinierte Einstellungen bestimmt. Die vordefinierten Einstellungen für die Identifier für Emergency, PDO's und SDO's errechnen sich aus der Knotenadresse, die zwischen 1 und 127 liegen kann, addiert zu einem Basisidentifier, der die Funktion festlegt.

Bit	10										0
COB-Identifier											
	Funktionscode				Modul-ID						

Tabelle 17: Ermittlung der COB-Identifier aus der Knotenadresse

Diese Basisidentifier liegen wie folgt:

Objekt	resultierende COB-ID [hex]	resultierende COB-ID [dezimal]	Kommunikationsparameter auf Index
EMERGENCY	80H + Modul-ID	129 – 255	
PDO1 (tx)	180H + Modul-ID	385 – 511	1800H
PDO1 (rx)	200H + Modul-ID	513 – 639	1400H
PDO2 (tx)	280H + Modul-ID	641 – 767	1801H
PDO2 (rx)	300H + Modul-ID	769 – 895	1401H
PDO3 (tx)	380H + Modul-ID	896 – 1022	1802H
PDO3 (rx)	400H + Modul-ID	1025 – 1050	1402H
PDO4 (tx)	480H + Modul-ID	1152 – 1278	1803H
PDO4 (rx)	500H + Modul-ID	1281 – 1406	1403H
SDO (tx)	580H + Modul-ID	1409 – 1535	
SDO (rx)	600H + Modul-ID	1537 – 1663	
Nodeguard/ Heartbeat/ Bootup	700H + Modul-ID	1793 – 1919	100EH

Tabelle 18: Basisidentifier

Über die vordefinierten SDO können die Konfigurationsdaten auf das Gerät geladen werden.

Nachdem das Gerät mit dem NMT-Dienst *Start\_Remote\_Node* vom NMT-Zustand "PRE-OPERATIONAL" in den NMT-Zustand "OPERATIONAL" versetzt worden ist, können PDO's versendet werden. Das Minimum-Device unterstützt außerdem die Dienste

*Stop\_Remote\_Node*, *Enter\_Pre-Operational\_State*, *Reset\_Node*, *Reset\_Communication*. Nach dem Einschalten und der Initialisierung geht das Gerät automatisch in den Zustand "PRE-OPERATIONAL". Mit *Reset\_Node* wird das Gerät komplett zurückgesetzt, *Reset\_Communication* bewirkt ein Rücksetzen der Kommunikationsparameter.

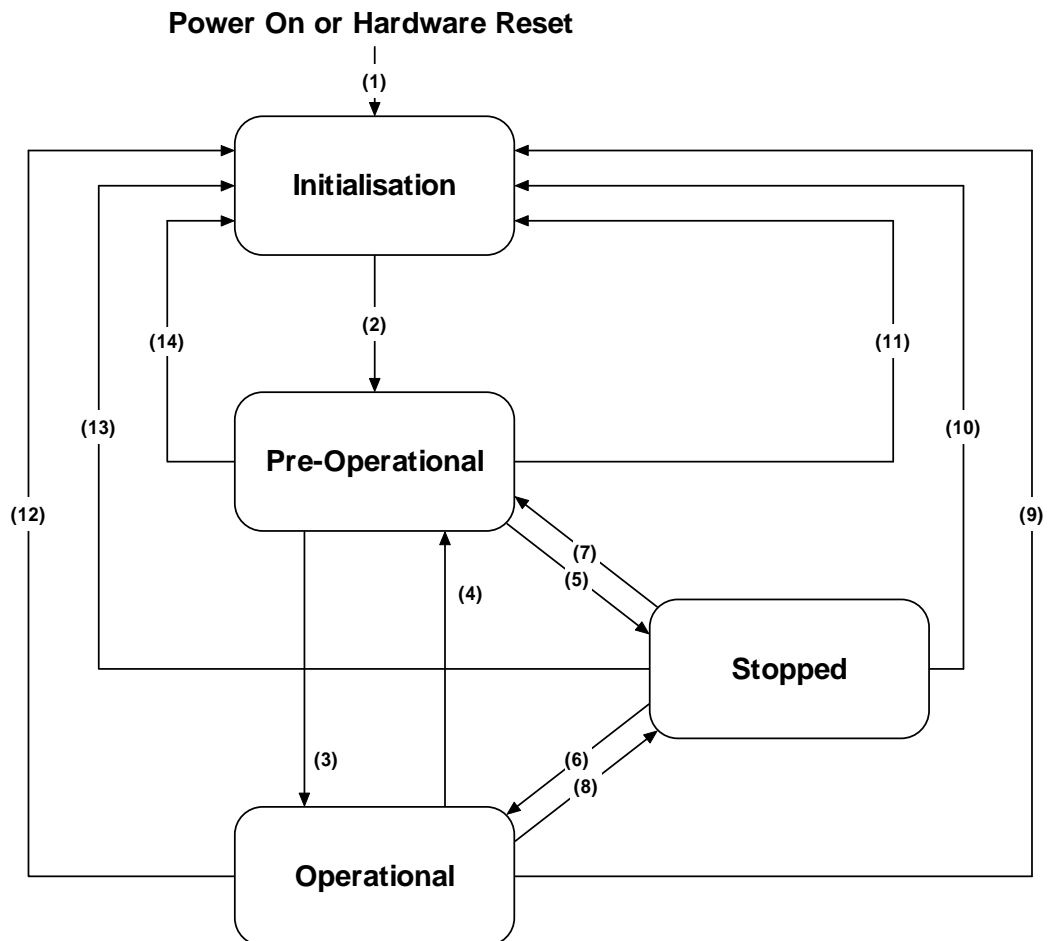


Bild 5: NMT-Zustandsdiagramm eines CANopen Gerätes

(1)	nach "Power On", automatischer Wechsel in die "Initialisation"
(2)	"Initialisation" beendet, automatischer Wechsel zu "Pre-Operational"
(3),(6)	NMT-Dienst "Start_Remote_Node"
(4),(7)	NMT-Dienst "Enter_Pre-Operational_State"
(5),(8)	NMT-Dienst "Stop_Remote_Node"
(9),(10),(11)	NMT-Dienst "Reset_Node"
(12),(13),(14)	NMT-Dienst "Reset_Communication"

Tabelle 19 : Legende zum Zustandsdiagramm

## 7.10 Objektverzeichniseinträge

Zu den Daten der Geräte gehören neben den Parametern für die PDO's eine Anzahl weiterer Einträge aus dem Objektverzeichnis. Zum Kommunikationsprofil gehören unter anderem der Gerätetyp auf Index [1000H], das Fehlerregister auf Index [1001H], das Pre-Defined-Error-Field auf [1003H], der Identifier der SYNC-Nachricht auf [1005H], der Gerätename, die Hardware- und Softwareversion des Herstellers auf [1008H], [1009H] und [100A], die Parameter Guard-Time und Life-Time-Factor auf [100CH] und [100DH]. Im Gerätetyp sind Informationen über das implementierte Geräteprofil und die Fähigkeiten des Gerätes verschlüsselt. Das Fehlerregister gibt Auskunft über interne Fehler des Gerätes, das Pre-Defined-Error-Field stellt ein Fehlerlogbuch bereit. Falls Guard-Time und Life-Time-Factor verschieden von Null parametrisiert sind, ergeben sie multipliziert die Lebenszeit des Gerätes für das Knotenüberwachungsprotokoll.

## 7.11 Beschreibung des PDO-Mapping am Beispiel

Alle Netzwerkvariablen können von PDO's übertragen werden. Ein PDO kann maximal 8 Byte transportieren. Die Zuordnung von Variablen zu PDO's ist über Mapping-Tabellen definiert. Diese sind über das Objektverzeichnis adressierbar. Das Lesen und Schreiben der Einträge des Objektverzeichnisses geschieht mittels SDO. Diese werden z.B. von einem Konfigurationswerkzeug verwendet, um das Netzwerk zu konfigurieren.

Der Vorgang soll an einem konkreten Beispiel veranschaulicht werden: Die Inputs 2 und 3 des Gerätes A sollen an die Outputs 1 und 3 des Geräts B übertragen werden. Beide Geräte unterstützen vollständiges Mapping.



## Device A:

1000H	Device Type
.....	
6000H,1	Input 1, 8 Bit
6000H,2	Input 2, 8 Bit
6000H,3	Input 3, 8 Bit
....	

## Transmit PDO Mapping Parameter

1A00H,0	NrOfEntries	2
1A00H,1	1.Map Object	60000208H
1A00H,2	2.Map Object	60000308H

## Transmit PDO Communication Parameter:

1800H,0	NrOfEntries	2
1800H,1	COB-ID	501
1800H,2	Trans.Type	255
....		

## Resultierendes PDO:

COB-ID	DATA	
501	Output 1	Output 3

Sende und Empfangs-PDO haben den gleichen CAN-Identifizier 501. Somit empfängt Device B automatisch das PDO, welches von Device A gesendet wird. Der Empfänger interpretiert die Daten gemäß seinem Mapping. Das heißt, er leitet das erste Byte an Output 1 weiter und das zweite Byte an Output 3. Der Sender wiederum hat in genau diese Bytes seine Inputs 2 und 3 gelegt, womit die Zuordnung korrekt gelöst ist.

## 7.12 Abbildung der Ein-/Ausgänge auf Objektverzeichniseinträge

Der CANopen ChipF40 ermöglicht eine einfache, nutzenorientierte Projektierung der CANopen-Anwendung.

Durch die feste Anzahl von Ein-/Ausgängen ist die Abbildung auf Prozeßdatenobjekte einfach. Die digitalen bzw. analogen Eingänge, sowie die digitalen Ausgänge werden auf die von der CiA vorgeschlagenen Prozessdatenobjekte abgebildet.

Die allgemeine Zuordnung nach Datentyp kann der *Tabelle 9* entnommen werden.

Die *Tabelle 20* und *Tabelle 21* zeigen die konkrete Zuordnung bei dem CANopen ChipF40 und CANopen ChipF40 V3:

Datentyp	Index / Subindex	Größe
<b>Digitale Eingänge</b>		
DI0 ... DI7	6000H / 1	BYTE
DI8 ... DI15	6000H / 2	BYTE
DI16 ... DI23	6000H / 3	BYTE
<b>Digitale Ausgänge</b>		
DO0 ... DO7	6200H / 1	BYTE
DO8 ... DO15	6200H / 2	BYTE
<b>Analoge Eingänge</b>		
AI0	6401H / 1	WORD
AI1	6401H / 2	WORD
AI2	6401H / 3	WORD
AI3	6401H / 4	WORD
AI4	6401H / 5	WORD
AI5	6401H / 6	WORD
AI6	6401H / 7	WORD
AI7	6401H / 8	WORD
<b>PWM Ausgänge</b>		
PWM0	6500H / 1; 6510H / 1	WORD
PWM1	6500H / 2; 6510H / 2	WORD
PWM2	6500H / 3; 6510H / 3	WORD
PWM3	6500H / 4; 6510H / 4	WORD

*Tabelle 20: Objektverzeichnis der Geräteparameter F40*

Datentyp	Index / Subindex	Größe
<b>Digitale Eingänge</b>		
DI0 ... DI7	6000H / 1	BYTE
DI8 ... DI15	6000H / 2	BYTE
DI16 ... DI23	6000H / 3	BYTE
DI24 ... DI27	6000H / 4	BYTE
<b>Digitale Ausgänge</b>		
DO0 ... DO7	6200H / 1	BYTE
DO8 ... DO15	6200H / 2	BYTE
<b>Analoge Eingänge</b>		
AI0	6401H / 1	WORD
AI1	6401H / 2	WORD
AI2	6401H / 3	WORD
AI3	6401H / 4	WORD
AI4	6401H / 5	WORD
AI5	6401H / 6	WORD
AI6	6401H / 7	WORD
AI7	6401H / 8	WORD

Tabelle 21: Objektverzeichnis der Geräteparameter F40 V3

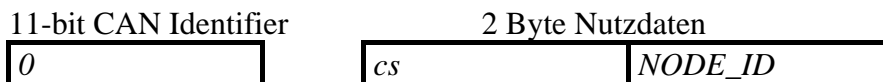
Anmerkung:

Nach dem Boot-Up der CANopen Baugruppe kann über SDO auf die Objekte zugegriffen werden. Befindet sich der Knoten im Status "OPERATIONAL", dann kann über PDO auf die Objekte zugegriffen werden. Dabei gelten die definierten Default-Mappingparameter. Eine Änderung der Mappingparameter erfolgt mit Hilfe eines Netzwerk-Konfigurators über SDO.

## 8 Funktionalität des CANopen ChipF40

### 8.1 CANopen Statusübergänge

Die NMT-Nachricht zur Änderung des Gerätestatus hat folgenden Aufbau:



*NODE\_ID* Knotenadresse ; *NODE\_ID* = 0 adressiert alle Baugruppen (Broadcast)

*cs* Kommando

In der *Tabelle 22* sind alle NMT-Master Telegramme zur Zustandssteuerung des CANopen Gerätes zusammengefaßt.

Kommando (cs)	Bezeichnung	Funktion	NMT-Status nach Ausführung
1 (01H)	Start_Remote_Node	startet das Gerät und die PDO-Übertragung, gibt Ausgänge frei	OPERATIONAL
2 (02H)	Stop_Remote_Node	stoppt die PDO-Übertragung, führt Ausgänge in den Fehlerzustand	STOPPED
128 (80H)	Enter_Pre_Operational_State	stoppt die PDO-Übertragung, SDO weiter aktiv	PRE-OPERATIONAL
129 (81H)	Reset_Node	führt einen Reset des Gerätes durch; Kaltstart, Ausgänge werden auf Ihre Defaultwerte zurückgesetzt	PRE-OPERATIONAL
130 (82H)	Reset_Communication	die Kommunikationsparameter werden auf Ihre Defaultwerte zurückgesetzt	PRE-OPERATIONAL

*Tabelle 22: NMT-Master Telegramme zur Zustandssteuerung*

## 8.2 Power On

Nach "Power-On" führt das Gerät die benötigten Initialisierungen durch und schaltet in den Zustand "PRE-OPERATIONAL".

## 8.3 PRE-OPERATIONAL

In diesem Zustand sind keine Prozess-Daten-Objekte (PDO's) aktiv. Die Default-Identifizier für die Service-Daten-Objekte (SDO's) stehen zur Verfügung. Über SDO können alle notwendigen Konfigurationen durchgeführt werden. Nach Abschluß der Konfiguration kann das Gerät in den Zustand "OPERATIONAL" versetzt werden. Dies erfolgt durch den NMT-Master oder durch den Anwender über einen Netzwerkkonfigurator.

## 8.4 OPERATIONAL

Im Zustand "OPERATIONAL" können Prozessdatenobjekte ausgetauscht werden. Auch ein Zugriff über SDO's ist möglich.

## 8.5 STOPPED

Im Zustand "STOPPED" wird die Kommunikation im ganzen angehalten. Dies gilt nicht für ein eventuell aktiviertes Node Guarding, dies funktioniert weiterhin. Ferner kann dieser Zustand dazu verwendet werden, die Applikation in eine Art „Sicherheitszustand“ zu bringen. In diesem Zustand funktioniert **keine** PDO-, SDO-, SYNC- und Emergency- Kommunikation. Man kann über eine NMT-Message diesen Zustand wieder verlassen.

## 8.6 Wiederanlauf nach Reset/ Power-On

Nach jedem Baugruppen Reset sendet das Gerät ein Bootup Telegramm. Temporäre Ausfälle der Baugruppe während des Betriebes und nachträgliches Einschalten der Baugruppe werden damit auch ohne Node Guarding (*siehe Kapitel 5.6*) erkannt. Der Sender kann über den Identifier der Nachricht ermittelt werden.

Der CANopen ChipF40 unterscheidet "Load"-Start und "Save"-Start.

Ein "Load"-Start ist notwendig :

- wenn das fabrikneue Gerät in Betrieb genommen wird.
- wenn die Geräteparameter (Objektverzeichnis-Einträge im RAM) mit Defaultwerten überschrieben werden sollen.

Beim "Load"-Start werden alle auf der CANopen ChipF40 existierenden Objektverzeichnis-Einträge nach Reset/Power-On in den RAM kopiert (Hersteller-Default-Werte).

Beim "Save"-Start (nach Schreiben von ("save") auf Objekt [1010H] Subindex 1) werden die gespeicherten Objektverzeichnis-Einträge nach Reset/Power-On in den RAM kopiert (gespeicherte Anwender-Werte). Beim CANopen ChipF40 kann dies bis zu 500 Millisekunden dauern, da das gesamte OD aus dem E<sup>2</sup>PROM in den RAM gelesen wird.

Nimmt der Bus-Master oder der Anwender (mit Netzwerkkonfigurator) Änderungen an den Objektverzeichnis-Einträgen vor, so werden die Änderungen beim nächsten RESTART erst berücksichtigt, wenn ein Schreiben von ("save") auf Objekt [1010H] Subindex 1 erfolgt ist. Das heißt nur die abgespeicherten Werte sind nach dem Reset bzw. Power-On des CANopen Gerätes gültig. Diese Werte werden dann in einem E<sup>2</sup>PROM gespeichert und gehen somit auch bei Spannungsausfall nicht verloren.

Über das Objekt [1010H] Subindex 1, können (wie bereits erwähnt) alle Geräte-Parameter in dem E<sup>2</sup>PROM gespeichert werden. Um ein versehentliches Abspeichern zu verhindern, muß eine bestimmte Signatur ("save") auf den Subindex 1 geschrieben werden. Im Hex-Code sieht die 32 bit-Signatur wie folgt aus:

MSB		LSB	
'e'	'v'	'a'	's'
65h	76h	61H	73H

Das Speichern der Geräteparameter im EEPROM kann bis zu 500 Millisekunden dauern.

Über das Objekt [1011H] Subindex 1 können alle Geräte-Parameter auf Herstellerdefaultwerte nach DS301 bzw. DS401 zurückgesetzt werden.

**Achtung!**  
 Dies gilt nicht für die Bitrate und Knotenadresse, die mittels LSS eingestellt wurde. Dies kann nur über die DIP Schalter DIP1 bis DIP4 erfolgen (*siehe Abschnitt 3.6*)

Um ein versehentliches Rücksetzen nach dem Abspeichern mit ("save") zu verhindern, muß die Signatur ("load") auf den Subindex 1 geschrieben werden. Im Hex-Code sieht die 32 bit-Signatur wie folgt aus:

MSB		LSB	
'd'	'a'	'o'	'l'
64h	61H	6fh	6ch

Um die Default-Werte gültig zu setzen, muß anschließend ein Reset bzw. Power-On durchgeführt werden.

## 8.7 NMT-Boot-Konfiguration

Der CANopen ChipF40 kann so konfiguriert werden, dass er als NMT-Boot-Master für alle im Netz befindlichen CANopen-Knoten fungiert. Dazu enthält das Object Dictionary zwei herstellerepezifische Einträge auf Index [2001H].

Subindex 1 ist der Eintrag für NMT-Boot-Enable, d.h. ist dort eine 0 eingetragen (default), so ist der Knoten kein Boot-Master, bei einer 1 ist der Knoten Bootmaster.

Subindex 2 ist der Eintrag für NMT-Start-Time, d.h. hier ist die Verzögerungszeit [ms] eingetragen, nach der der Knoten bei Systemstart die NMT-Boot-Nachricht verschickt (default 500ms, max. 65s).

Die Einträge für die NMT-Boot-Konfiguration werden unmittelbar beim Schreiben des Eintrages im EEPROM gesichert, und werden nicht beachtet beim Sichern und Wiederherstellen des OD (s.a. 8.6). Somit ist der Eintrag direkt mit den entsprechenden Werten zu beschreiben, um den CANopen ChipF40 als Boot-Master zu konfigurieren oder nicht.

Die Einstellungen werden jeweils beim nächsten Reset des CANopen ChipF40 aktiv.

Die SDOs zum Schreiben von NMT-Boot-Enable und NMT-Start-Time sehen beispielhaft wie folgt aus:

Dienst	CAN ID für Node 0x40	DLC	SDO Cmd	Index		Sub-index	Wert			
				01	20		01	00	00	00
NMT-Boot-Enable setzen	640	8	2F	01	20	01	01	00	00	00
NMT-Boot-Enable löschen	640	8	2F	01	20	01	00	00	00	00
NMT-Start-Time auf 0x3E8 setzen	640	8	2B	01	20	02	E8	03	00	00

Tabelle 23 : SDOs zum Setzen der NMT-Boot-Konfiguration

Der zu schreibende CAN-Index liegt bei 0x600 + NodeID. Die Werte werden gemäss CANopen-Standard LSB first abgelegt.

## 8.8 Funktionalität der analogen Eingänge

### 8.8.1 Ablage der analogen Werte

Dieser Abschnitt beschreibt die allgemeine Ablage der analogen Werte in einem CANopen-Frame.

Wie im CANopen Standard DS401 definiert, werden alle analogen Werte mit einer Auflösung bis zu 15-Bit als 16 Bit Wert "linksbündig" abgelegt. Der A/D-Wandler des CANopen ChipF40 besitzt eine Auflösung von 10 Bit. Dieser 10-Bit Wert wird um 5 Bit's nach links verschoben. Die unteren 5 Bits werden mit Null gefüllt. Das Vorzeichen (Bit 15) ist immer Null. Für den Empfänger der Nachricht ist der A/D-Wert somit ein positiver Wert mit 15-Bit Auflösung. Dies ist bei der Auswertung zu beachten.

Für die Übertragung eines analogen Einganges werden 2 Datenbytes verwendet. Diese werden wie folgt abgelegt und über den CAN-Bus übertragen :



Byte 2								Byte 1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+/-	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0	0	0	0	0

Tabelle 24 : Ablage der Analogdaten

Auf dem CAN-Bus wird zuerst das Byte 1 und dann das Byte 2 übertragen.

### 8.8.2 Berechnungsformel für den analogen Eingangswert

Um aus dem übertragenen Wert des Analog/Digital Wandlers einen Spannungswert des analogen Einganges zu berechnen, kommt folgende Formel zum Einsatz :

$$AIn_{/V} = \frac{\text{gewandelter ADWert}_{/hex} \cdot \text{Spannungsbereich}_{/V}}{2^{\text{AuflösungADC}}}$$

Zum besseren Verständnis im Umgang mit der Formel soll das folgende Beispiel dienen :

über CAN gesendeter A/D-Wert = 012A0H (4768 dez)  
 Spannungsbereich = 5V (Standard Versorgung an den Pins VAREF und VAGND)  
 logische Auflösung des CAN-Wertes = 15 Bit  
 analoger Eingangswert AIn = 0,727V

Für die kleinste Quantisierung des A/D- Wertes muß die tatsächliche Auflösung des A/D- Wandler verwendet werden.

A/D-Wert = 01H  
 Spannungsbereich = 5V (Standard Versorgung an den Pins VAREF und VAGND)  
 Auflösung des A/D-Wandlers = 10 Bit  
 kleinste Auflösung = 4,88 mV/Digit

### 8.8.3 Auswahl des Interrupt Triggers

Dieser Objekteintrag bestimmt, aufgrund welchem Ereignis ein Interrupt ausgelöst wird. Hierfür steht das Objekt [6421H] "Interrupt\_Trigger\_Selection" zur Verfügung. Das Auslösen eines Interruptes ist auch gleichzeitig die Veranlassung zum Senden des TX-PDO's für die analogen Eingänge, unter der Voraussetzung, daß der "Global\_Interrupt\_Enable" [6423H] aktiviert wurde.

Für jeden analogen Kanal existiert ein Subindex. Somit kann für jeden Kanal separat bestimmt werden, welches Ereignis das entscheidende ist.

Die Defaultwerte des Interrupttrigger betragen für alle analogen Eingänge 4.

Folgende Werte können eingetragen werden :

Bit-Nr.	Interrupt Trigger
0	Oberer Grenzwert überschritten
1	Unterer Grenzwert unterschritten
2	Eingangswert ändert sich um mehr als <i>DELTA</i> [6426H]
3	<b>wird nicht unterstützt !</b>
4	<b>wird nicht unterstützt !</b>
5 bis 7	<b>Reserviert</b>

Tabelle 25: Interrupt Trigger Bits

Beispiel :

**6421H,1 = 04H** bedeutet, daß sich der erste analoge Eingang um mehr als *DELTA* ändern muß, bevor das PDO gesendet wird.

### 8.8.4 Interrupt Quelle

In diesem Objekteintrag wird hinterlegt, welcher analoge Eingang den Interrupt ausgelöst hat. Hierfür steht das Objekt [6422H] "Analogue\_Input\_Interrupt\_Source" zur Verfügung. Dabei bezieht sich jedes einzelne Bit auf einen zugeordneten analogen Kanal. Die Bits werden automatisch rückgesetzt, wenn der Eintrag per SDO ausgelesen oder der Objekteintrag mittels PDO übertragen wurde.

Den Bits werden dabei folgende Wertigkeit zugeordnet :

"1" : Kanal hat Interrupt ausgelöst,

"0" : Kanal hat kein Interrupt ausgelöst.

Beispiel:

**6422H,1 = 01H**, d.h. der analoge Kanal 0 hat einen Interrupt ausgelöst.

### 8.8.5 Interrupt Aktivierung

In diesem Objekteintrag wird die Beachtung der Grenzwerte und der Deltafunktion der analogen Eingänge aktiviert oder gesperrt. Hierfür steht das Objekt [6423H] "Analogue\_Input\_Global\_Interrupt\_Enable" zur Verfügung. Der Defaultwert ist 0, d.h. die Verarbeitung der Grenz- und Deltawerte ist inaktiv. Durch Schreiben einer 1 auf diesen Eintrag wird die Auswahl des Interrupt Triggers (*siehe Kapitel 8.8.3*) aktiviert.

### 8.8.6 Interrupt oberer und unterer Grenzwert

Ein Interrupt wird ausgelöst, wenn der analoge Eingangswert größer [6424H] bzw. kleiner [6425H] als der angegebene Grenzwert im Subindex ist. Dies geschieht nur, wenn der OD-Eintrag [6423H] auf 1 steht.

Solange die Triggerbedingung erfüllt ist, wird jeder Wert der analogen Eingänge übertragen. Dies geschieht nur, wenn keine weitere Triggerbedingung, wie die Deltafunktion, gesetzt wurde. Die Grenzwerte sind als linksbündige 32 Bit Werte anzugeben.

Hierfür stehen die Objekte [6424H] "Analogue\_Input\_Interrupt\_Upper\_Limit\_Integer" und [6425H] "Analogue\_Input\_Interrupt\_Lower\_Limit\_Integer" zur Verfügung.

Die Defaultwerte für unteren und oberen Grenzwert betragen für alle analogen Eingänge 0.

Beispiel :

**6423H = 1H**, **6421H,1 = 05H** und **6424H,1 = 2000H** -> analoger Eingang 1 löst einen Interrupt aus, wenn der Grenzwert von 2000H (Auflösung 10 Bit CANopenChip) überschritten wird und sich dann der Wert um mehr als den Deltawert ändert.

### 8.8.7 Deltafunktion

Mit der Deltafunktion kann eingestellt werden, um wieviel sich ein analoger Eingangswert seit der letzten Übertragung ändern muß, damit er auf den Bus gesendet wird. Hierfür steht das Objekt [6426H] "Analogue\_Input\_Interrupt\_Delta" zur Verfügung. Die Eingabe erfolgt rechtsbündig und bezieht sich auf die Werte im OD, d.h. 15 Bit Auflösung.

Die Defaultwerte für Delta betragen für alle analogen Eingänge 0xA0.

### 8.8.8 Beispiel zu den Triggerbedingungen

An dem folgendem Beispiel soll das Zusammenspiel der Objekte 6424H bis 6426H dargestellt werden.

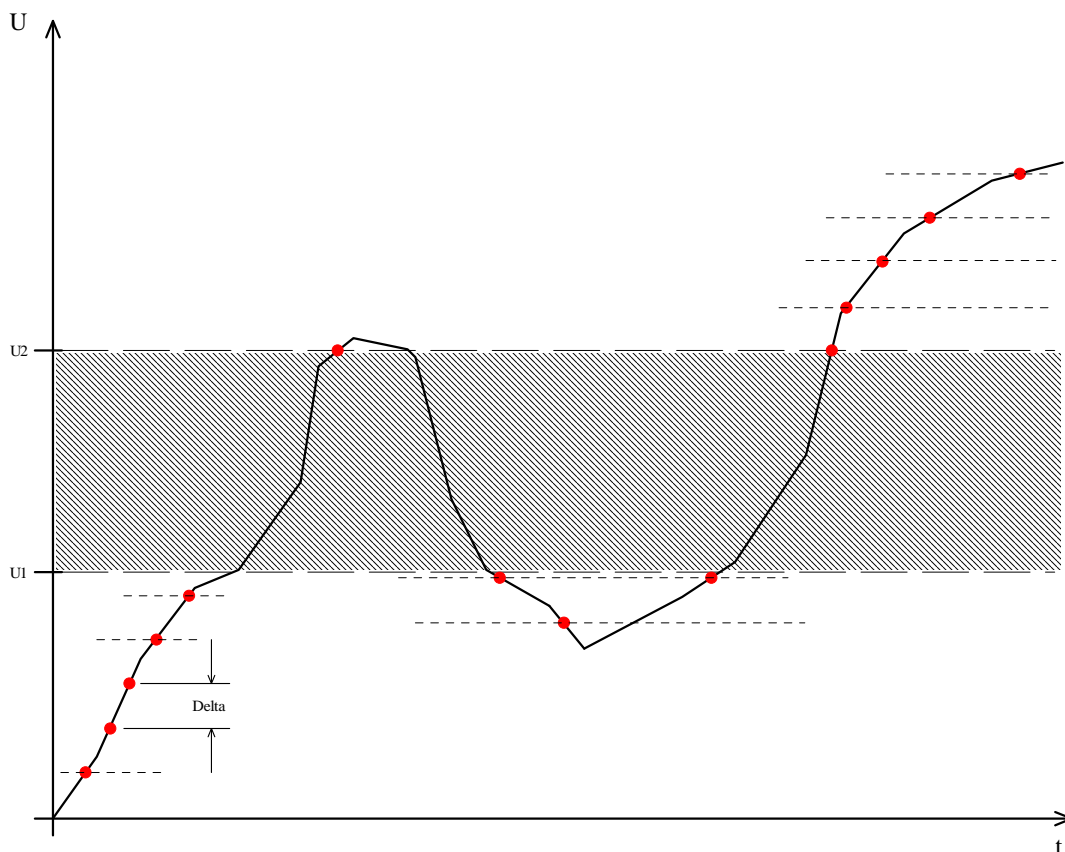


Bild 6: Beispiel Triggerbedingungen

Es wurden die Werte *U1* für LowerLimit und *U2* für UpperLimit im OD eingetragen. Weiterhin wurde der Wert *Delta* im Index 6426H eingetragen. Der dargestellte Spannungsverlauf liegt an einem analogen Eingang an. Zu den Zeitpunkten die mit einem  $\lambda$  versehen sind, wird ein entsprechendes PDO vom CANopen ChipF40 gesendet. Befindet sich jedoch der Analogwert im Bereich der schraffierten Fläche, so wird kein PDO gesendet.

## 8.9 Funktionalität der PWM Ausgänge (nicht F40 V3)

Mit dem CANopen ChipF40 können PWM-Signale erzeugt werden. Pro Ausgang existiert ein OD-Eintrag für Periodendauer (Index [6510H]) und Tastverhältnis (Index [6500H]). Beide Parameter haben das Format unsigned 16.

Das Tastverhältnis wird in Prozent angegeben. Dies bedeutet, daß der Wert 0H – 0% und 0FFFFH – 100% entspricht.

Die Default-Einstellung für die PWM- Ausgänge beträgt 0.

Der Periodendauer wird als Vielfaches von 1 $\mu$ s angegeben. Der Wert von 1000 entspricht also einer Frequenz von 1kHz.

Die Default-Einstellung für die PWM- Ausgänge beträgt 1000.

Der kleinste einstellbare Wert liegt bei 43 (CAN-Baudrate ungleich 10kBit), bzw bei 86 (CAN-Baudrate gleich 10 kBit).

## 8.10 Emergency Telegramm

Der Status des CANopen ChipF40 wird im Fehlerfall über hochpriore Notfall Telegramme (Emergency Telegramme) übermittelt. Diese Telegramme haben eine Datenlänge von 8 Bytes und enthalten Fehlerinformationen.

Das Emergency Telegramm wird übertragen, sobald einer der signalisierten Fehler aufgetreten ist. Das spezifische Emergency Telegramm wird immer nur einmal übertragen, auch wenn der Fehler längere Zeit ansteht. Sind alle Fehlergründe beseitigt, so wird erneut ein Emergency Telegramm mit Inhalt 0 (Fehler beseitigt) übertragen.

Die Struktur der Emergency Telegramme (8 Byte Daten) ist nachstehend dargestellt:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Error Code		Error-Register [1001H]	herstellerspezifischer Error Code				

Tabelle 26: Emergency Telegramm (BYTE-Feld)

### 8.10.1 Error Code

Der Error Code (Bytefeld 0+1, LSB, MSB) zeigt an, ob ein Fehler vorliegt, oder ob der Fehler bereits beseitigt ist (kein Fehler). Folgende Error Codes können auftreten :

0000H	kein Fehler
1000H	globaler Fehler
5000H	Hardwarefehler (Hardwarereset wurde ausgelöst)
6100H	Softwarefehler (Softwarereset wurde ausgelöst)
6101H	Softwarefehler (Watchdogreset wurde ausgelöst)
6102H	Softwarefehler (Stacküberlauf des Mikrocontrollers)
8110H	CAN-Nachricht verloren (Buslast zu hoch)
8120H	Gerät befindet sich im Error passiv Mode
8130H	Lifeguard oder Heartbeat Fehler Bei Heartbeat Consumer-Fehler wird im herstellerspezifischen Teil (Byte 3) die Knotenadresse des ausgefallenen Gerätes übertragen.
8140H	Gerät hatte CAN-BUSOFF erkannt
8210H	Das PDO konnte aufgrund falscher Länge nicht verarbeitet werden.

### 8.10.2 Error Register

Das Error Register (Byte 2 im Bytefeld) kann folgende Werte annehmen :

- 81H: ein herstellerspezifischer Fehler ist aufgetreten .
- 11H: CAN-Kommunikationsfehler
- 01H: ein allgemeiner Fehler ist aufgetreten
- 00H: es besteht kein Fehler mehr – Fehlerreset

## 8.11 Statusanzeige über Run und Error LED

Der aktuelle Zustand vom CANopen ChipF40 wird zur Laufzeit an den beiden Status-Leuchtdioden D1 und D2 dargestellt. Die Funktion der beiden LED's ist im Standard **CiA DR-303-3 V1.0** definiert.

### 8.11.1 Run LED

Die grüne Run LED (D2) zeigt den NMT Status der Baugruppe an. Die *Tabelle 27* zeigt die verschiedenen Anzeigezustände und deren Bedeutung.

RUN LED	Zustand	Beschreibung
An	OPERATIONAL	CANopen ChipF40 befindet sich im Zustand OPERATIONAL
Blinken Verhältnis 50:50	PRE-OPERATIONAL	CANopen ChipF40 befindet sich im Zustand PREOPERATIONAL
kurzes Blinken (Single Flash)	STOPPED	CANopen ChipF40 befindet sich im Zustand STOPPED
Dreifaches Blinken (Tripple Flash) mit der Error LED	Stacküberlauf	Die Software verursachte einen Stacküberlauf des Mikrocontrollers.
Abwechselndes Blinken mit der Error LED	Zugriff per LSS	Es wird momentan ein LSS Service durchgeführt.
Synchrones schnelles Blinken mit der Error LED	Reset nach Default	Am DIP-Switch wurde das Rücksetzen auf Defaultwerte eingestellt.

*Tabelle 27: Run LED Zustände*



### 8.11.2 Error LED

Die rote Error LED (D1) zeigt den aktuellen Fehlerzustand des CANopen ChipF40 an. Die *Tabelle 28* beschreibt die verschiedenen Zustände und deren Bedeutung:

ERROR LED	Zustand	Beschreibung
Aus	kein Fehler	Gerät arbeitet einwandfrei
Kurzes Blinken (Single Flash)	Warning Limit erreicht	Das Warning Limit im CAN Controller wurde erreicht (zu viele Error Frames auf dem CAN Bus).
Abwechselndes Blinken mit der Run LED	Zugriff per LSS	Es wird momentan ein LSS Service durchgeführt.
Doppeltes Blinken (Double Flash)	Error Control Event	Ein Fehler beim Lifeguard, Nodeguard oder Heartbeat wurde erkannt.
Dreifaches Blinken (Tripple Flash) mit der Run LED	Stacküberlauf	Die Software verursachte einen Stacküberlauf des Mikrocontrollers.
An	Bus Off	Der CAN Controller befindet sich im Zustand "Bus Off".
Synchrones schnelles blinken mit der Run LED	Reset nach Default	Am DIP-Switch wurde das Rücksetzen auf Defaultwerte eingestellt.

*Tabelle 28: Error Led Zustände*

## 9 Systemverhalten im Fehlerfall

### 9.1 Zustand des CANopen ChipF40 im Fehlerfall

Über den Eintrag „Error Behaviour“ im Objektverzeichnis (Index [1029] beim F40 und Index [67FEH] beim F40 V3) kann eingestellt werden, in welchen NMT-Zustand der CANopen ChipF40 im Fehlerfall wechselt.

Mögliche Einträge:

SubIndex 1: Communication Error

- 0: Zustandswechsel nach PRE-OPERATIONAL
- 1: kein Zustandswechsel
- 2: Zustandswechsel nach STOPPED

Die Einstellung gilt dann für alle möglichen Fehlerarten. (Ausnahme Error Code: 8120H).

Die Einträge Output Error (SubIndex 2) und Input Error (SubIndex 3) werden nicht unterstützt.

Der Error Code 6102H führt immer zum Schalten der Ausgänge in den Fehlerzustand. Dieser Fehler ist kritisch und kann nur durch Reset verlassen werden.

## 9.2 Verhalten der Ausgänge im Fehlerfall

Der Anwender kann für jeden Ausgang festlegen, wie er sich im Fehlerfall verhalten soll. Die Ausgänge werden nur verändert, wenn im Index „Error Behaviour“ ein Zustandswechsel aktiviert wurde. Alle Fehler, die nicht zu einem Zustandswechsel führen, setzen auch die Ausgänge nicht in den Fehlerzustand.

### 9.2.1 Digitale Ausgänge

Bei digitalen Ausgängen kann über die Objekte [6206H] (*"Error\_Mode\_Output\_8-Bit"*) und [6207H] (*"Error\_Value\_Output\_8-Bit"*) das Ausgangsverhalten im Fehlerfall vordefiniert werden. Dies kann mit Hilfe des Netzwerkkonfigurators geschehen. In der Default-Einstellung ändern sich die Ausgänge im Fehlerfall nicht.

Eine "1" an der Bitstelle des entsprechenden Ausganges im Objekt [6206H] bewirkt, daß der im Objekt [6207H] stehende Wert für die Bitstelle ("0" oder "1") auf den entsprechenden Ausgang geschrieben wird.

Beispiel :

Digitale Ausgänge

Index	Sub-Index	DO 3	DO 2	DO 1	DO 0	
6206H	1	0	0	1	1	Error Mode Output 8-Bit
6207H	1	X	X	0	1	Error Value Output 8-Bit

Tabelle 29: Beispiel für Verhalten der Ausgänge im Fehlerfall

Im Fehlerfall werden die digitalen Ausgänge DO 0 auf 1 sowie DO 1 auf 0 gesetzt. Die Zustände der Ausgänge DO 2 und DO 3 bleiben unverändert.

### 9.2.2 PWM Ausgänge

Bei PWM-Ausgängen kann über die Objekte [6543H] ("*PWM Output Error Mode*") und [6544H] ("*PWM Output Error Value*") das Ausgangsverhalten im Fehlerfall vordefiniert werden. Dies kann mit Hilfe des Netzwerkkonfigurators geschehen. In der Default-Einstellung ändern sich die Ausgänge im Fehlerfall nicht.

Eine "1" im Objekt [6543H] des entsprechenden Ausganges bewirkt, daß der im Objekt [6544H] stehende Wert auf den entsprechenden Ausgang geschrieben wird.

Beispiel :

Index	Sub-Index	PWM 0 Fall 1	PWM 0 Fall 2	
6543H	1	0	1	PWM Output Error Mode
6544H	1 = 0x4000	keine Änderung	0x4000 also 25%	PWM Output Error Value

*Tabelle 30: Beispiel für Verhalten der Ausgänge im Fehlerfall*

Im Fehlerfall bleibt der PWM-Ausgang0 im Fall 1 unverändert. Im Fall 2 wird das Tastverhältnis auf 25% eingestellt.

### 9.3 Wechsel von Fehlerfall zum normalen Betrieb

Die Ausgänge behalten die im Fehlerfall eingenommenen Zustände, bis sie durch einen neuen Ausgangswert (mittels PDO oder SDO) überschrieben werden. Voraussetzung für das Schreiben per PDO ist, daß zuvor die Ursache beseitigt wurde ("Busoff-", „Heartbeatconsumer-" oder "Life Guarding-" Fehler) und die Baugruppe in den Zustand "OPERATIONAL" (Master: "*Start\_Remote\_Node*") gesetzt wurde.



## 10 Das CANopen ChipF40 Objektverzeichnis

Index [hex]	Objekt	Name	Datentyp	in PDO mappbar
1000	Var	Geräte Typ	Unsigned32	-
1001	Var	Fehler Register	Unsigned8	-
1003	Array	Fehler Meldung	Unsigned32	-
1005	Var	Identifizier SYNC-Nachricht	Unsigned32	-
1007	Var	SYNC window length	Unsigned32	-
1008	Var	Gerätebezeichnung	String	-
1009	Var	Hardware Version	String	-
100A	Var	Software Version	String	-
100C	Var	Guard Time	Unsigned16	-
100D	Var	Life Time Factor	Unsigned8	-
1010	Array	User-Parameter save	Unsigned32	-
1011	Array	Default-Parameter reload	Unsigned32	-
1014	Var	Identifizier Emergency	Unsigned32	-
1016	Array	Consumer Heartbeat Time	Unsigned32	-
1017	Var	Producer Heartbeat Time	Unsigned16	-
1018	Record	Identity Object	Identity	-
1029	Array	Error Behaviour	Unsigned8	-
1200	Record	1 <sup>st</sup> Server SDO Parameter	SDO Parameter	-
1201	Record	2 <sup>nd</sup> Server SDO Parameter	SDO Parameter	-
1400	Record	RxPDO1 Kommunikationsparameter Digital Output	PDOComPar	-
1401	Record	RxPDO2 Kommunikationsparameter Digital Output	PDOComPar	-
1402	Record	RxPDO3 Kommunikationsparameter Digital Output	PDOComPar	-
1403	Record	RxPDO4 Kommunikationsparameter Digital Output	PDOComPar	-
1600	Record	RxPDO1 Mappingparameter Digital Output	PDOMapping	-
1601	Record	RxPDO2 Mappingparameter Digital Output	PDOMapping	-
1602	Record	RxPDO3 Mappingparameter Digital Output	PDOMapping	-
1603	Record	RxPDO4 Mappingparameter Digital Output	PDOMapping	-
1800	Record	TxPDO1 Kommunikationsparameter Digital Input	PDOComPar	-
1801	Record	TxPDO2 Kommunikationsparameter Analog Input	PDOComPar	-
1802	Record	TxPDO3 Kommunikationsparameter Analog Input	PDOComPar	-
1803	Record	TxPDO4 Kommunikationsparameter Analog Input	PDOComPar	-

1A00	Record	TxPD01 Mappingparameter Digital Input	PDOMapping	-
1A01	Record	TxPD02 Mappingparameter Analog Input	PDOMapping	-
1A02	Record	TxPD03 Mappingparameter Analog Input	PDOMapping	-
1A03	Record	TxPD04 Mappingparameter Analog Input	PDOMapping	-
2000	Var	I/O Configuration	Unsigned8	-
2001	Var	NMT-Boot-Configuration	Unsigned8	-
6000	Array	PDO Digital Input	Unsigned8	x
6200	Array	PDO Digital Output	Unsigned8	x
6206	Array	Error Mode Digital Output	Unsigned8	-
6207	Array	Error State Digital Output	Unsigned8	-
6401	Record	PDO Analog Input	Integer16	x
6421	Array	Interrupt Trigger Auswahl	Unsigned8	-
6422	Array	Interrupt Quelle	Unsigned32	x
6423	Var	Globaler Interrupt freigeben	Boolean	-
6424	Array	Interrupt oberes Limit	Integer32	-
6425	Array	Interrupt unteres Limit	Integer32	-
6426	Record	Input Interrupt Delta	Unsigned32	-
6500	Array	PWM Pulse	Unsigned16	x
6510	Array	PWM Periode	Unsigned16	x
6543	Array	PWM Output Error Mode	Unsigned8	-
6544	Array	PWM Output Error Value	Unsigned16	-

Tabelle 31: Objektverzeichnis (Object Dictionary) des F40

## 11 Das CANopen ChipF40 V3 Objektverzeichnis

Index [hex]	Objekt	Name	Datentyp	in PDO mappbar
1000	Var	Geräte Typ	Unsigned32	-
1001	Var	Fehler Register	Unsigned8	-
1003	Array	Fehler Meldung	Unsigned32	-
1005	Var	Identifizier SYNC-Nachricht	Unsigned32	-
1007	Var	SYNC window length	Unsigned32	-
1008	Var	Gerätebezeichnung	String	-
1009	Var	Hardware Version	String	-
100A	Var	Software Version	String	-
100C	Var	Guard Time	Unsigned16	-
100D	Var	Life Time Factor	Unsigned8	-
1010	Array	User-Parameter save	Unsigned32	-
1011	Array	Default-Parameter reload	Unsigned32	-
1014	Var	Identifizier Emergency	Unsigned32	-
1016	Array	Consumer Heartbeat Time	Unsigned32	-
1017	Var	Producer Heartbeat Time	Unsigned16	-
1018	Record	Identity Object	Identity	-
1200	Record	1 <sup>st</sup> Server SDO Parameter	SDO Parameter	-
1201	Record	2 <sup>nd</sup> Server SDO Parameter	SDO Parameter	-
1400	Record	RxPDO1 Kommunikationsparameter Digital Output	PDOComPar	-
1401	Record	RxPDO2 Kommunikationsparameter Digital Output	PDOComPar	-
1600	Record	RxPDO1 Mappingparameter Digital Output	PDOMapping	-
1601	Record	RxPDO2 Mappingparameter Digital Output	PDOMapping	-
1800	Record	TxPDO1 Kommunikationsparameter Digital Input	PDOComPar	-
1801	Record	TxPDO2 Kommunikationsparameter Analog Input	PDOComPar	-
1802	Record	TxPDO3 Kommunikationsparameter Analog Input	PDOComPar	-
1803	Record	TxPDO4 Kommunikationsparameter Analog Input	PDOComPar	-
1A00	Record	TxPD01 Mappingparameter Digital Input	PDOMapping	-
1A01	Record	TxPD02 Mappingparameter Analog Input	PDOMapping	-
1A02	Record	TxPD03 Mappingparameter Analog Input	PDOMapping	-
1A03	Record	TxPD04 Mappingparameter Analog Input	PDOMapping	-
2000	Var	I/O Configuration	Unsigned8	-



*Das CANopen ChipF40 V3 Objektverzeichnis*

2001	Var	NMT-Boot-Configuration	Unsigned8	-
6000	Array	PDO Digital Input	Unsigned8	x
6200	Array	PDO Digital Output	Unsigned8	x
6206	Array	Error Mode Digital Output	Unsigned8	-
6207	Array	Error State Digital Output	Unsigned8	-
6401	Record	PDO Analog Input	Integer16	x
6421	Array	Interrupt Trigger Auswahl	Unsigned8	x
6422	Array	Interrupt Quelle	Unsigned32	x
6423	Var	Globaler Interrupt freigeben	Boolean	-
6424	Array	Interrupt oberes Limit	Integer32	x
6425	Array	Interrupt unteres Limit	Integer32	x
6426	Record	Input Interrupt Delta	Unsigned32	x
67FE	Array	Error Behaviour	Unsigned8	-

*Tabelle 32: Objektverzeichnis (Object Dictionary) des F40 V3*

## 12 Änderungen im Dokument

<b>Datum</b>	<b>Versionsnummer</b>	<b>Änderungen</b>
02.11.2004	Handbuch L1062d_1	erste Handbucherstellung
22.04.2005	Handbuch L1062d_2	Errorcode für Stacküberlauf und Anzeige dessen an der LEDs eingefügt
23.01.2008	Handbuch L1062d_3	Ergänzung der CANopen Chip F40 Version 3301002
26.06.2009	Handbuch L1062d_4	Änderung der maximalen PWM-Frequenz bei 10kBit CAN
03.03.2009	Handbuch L1062d_6	Maximalströme der digitalen Ausgänge hinzugefügt
26.11.2012	Handbuch L1062d_7	Hinweis für Verwendung von TxDC und RxDC.

---

**Index**
**I**

1000 .....	42
1003 .....	37
1005 .....	42
100C.....	42
100D.....	42
1400 .....	34
1800 .....	34

**6**

6000 .....	36
6200 .....	36

**A**

Array .....	33
-------------	----

**B**

Basisidentifizier .....	40
Baugruppenreset .....	25
Berechnungsformel.....	51
Betriebsspannung.....	18
Betriebstemperaturbereich .....	18
Bitrate.....	20
Bitraten.....	12, 13
Bus Off.....	62
Buslast.....	34

**C**

CAN-Identifizier.....	23, 43
CAN-Nachricht.....	23
CAN-Schnittstelle.....	10, 20
CiA DSP 401 .....	31
CiA DSP 402 .....	31
CiA DSP 406 .....	31
CiA, CAN in Automation .....	28

**D**

Datenflash .....	48
Defaultwerte.....	14
Defstruct.....	33
Deftype.....	33
Deltafunktion .....	54
Device Profile .....	32
DIPmodul-Connector .....	5, 7
Domain.....	33
Dummy16 .....	36

**E**

Elektrische Parameter .....	18
Emergency .....	37
Emergency Telegramme .....	56
Emergency-Objekt .....	37
Empfangsobjekt .....	34
Enter_Pre_Operational_State....	22
Error Code.....	57
Error LED .....	59
Error Register.....	57
Eventtimer .....	35

**F**

Fehlermeldung .....	37
---------------------	----

**G**

Geräteprofile .....	32
Gerätstatus .....	46
Gerätetyp.....	42
Gewicht .....	19
GuardTime .....	42

**H**

Heartbeat .....	38
Heartbeat Nachricht .....	38
Heartbeat Producer.....	38

---

Heartbeat-Consumer.....	39	OPERATIONAL .....	40, 47
<b>I</b>		<b>P</b>	
Identifizier .....	34	PDO .....	34, 36, 42
Index .....	32, 36	PDO-Mapping .....	36
Inhibittime .....	34	Power-On .....	47, 48, 49
Interrupt Enable .....	53	Pre-defined-Error-Field .....	42
Interrupt oberer Grenzwert .....	53	PRE-OPERATIONAL .....	41, 47
Interrupt Quelle .....	52	Prioritätsgruppe.....	35
Interrupt Trigger .....	52	Prozeßdatenobjekte.....	34, 44
Interrupt unterer Grenzwert.....	53	<b>Q</b>	
<b>K</b>		Quittungsbetrieb .....	34
Knotenadresse.....	11, 12, 23, 40	<b>R</b>	
Kommunikationsprofil .....	33	Record.....	33
Konfigurationsmöglichkeiten.....	9	Remote-Frames .....	34
Kontaktierung .....	19	RESET .....	10
<b>L</b>		Reset_Communication.....	41
Lagertemperaturbereich.....	18	RESTART .....	48
Life Guarding .....	25, 38	RTR-Bit .....	25
LIFEGUARDING .....	62	<b>Run- LED</b> .....	58
LifeTimeFactor.....	42	<b>S</b>	
Load-Start .....	48	Save-Start.....	48
Luftfeuchtebereich.....	18	SDO .....	29, 33, 36, 40, 42
<b>M</b>		Sperrzeit .....	34, 35
mandatory .....	33	Standardeinstellungen.....	14
Mapping.....	36, 42	Start der Baugruppe .....	22
Mechanische Eigenschaften .....	19	Start_Remote_Node.....	22, 40
Minimum Boot-Up .....	22	<b>Statusanzeige</b> .....	58
Minimum-Device .....	40	Stiftleisten .....	19
Modulgröße .....	19	Stop_Remote_Node.....	41
Multiplexed Domain Protokoll.	33	Stromaufnahme.....	18
<b>N</b>		Subindex .....	34, 36
Network Management .....	37	SYNC.....	42
NMT-Master.....	37, 38, 40	SYNC Message.....	33
Node Guarding .....	25, 47	<b>T</b>	
<b>O</b>		Taktversorgung .....	18
Objektverzeichnis ...	29, 30, 32, 42	transmission type .....	35
OD .....	29, 32, 42		

---

---

<b>U</b>		
Übertragungstypen.....	35	
Umgebungsbedingungen .....	18	
<b>V</b>		
Var .....	33	
Versorgungsspannung.....	20	
		vordefinierte Identifier .....
		40
		<b>Z</b>
		zertifiziert .....
		3
		Zustandsdiagramm .....
		40



---

**Dokument:** CANopen ChipF40  
**Dokumentnummer:** 7., Auflage Juni 2012

---

**Wie würden Sie dieses Handbuch verbessern?**

---

---

---

---

**Haben Sie in diesem Handbuch Fehler entdeckt?** Seite

---

---

---

---

**Eingesandt von:**

Kundennummer: \_\_\_\_\_

Name: \_\_\_\_\_

Firma: \_\_\_\_\_

Adresse: \_\_\_\_\_

---

**Einsenden an:** SYS TEC electronic GmbH  
August-Bebel-Str. 29  
D-07973 Greiz  
GERMANY  
Fax : +49 (0) 36 61 / 62 79 99

Veröffentlicht von

---

© SYS TEC electronic GmbH 2010

**SYS TEC**  
ELECTRONIC

Best.-Nr. L-1062d\_7  
Printed in Germany