

AN1201

Einführung in CANopen

Die Einführung in CANopen beschreibt die grundlegenden Kommunikationsmechanismen und die Verwendung der Identifier.

Autor: Uwe Koppe
MicroControl GmbH & Co. KG

Inhaltsverzeichnis

1	Was ist CANopen?	3
1.1	Objektverzeichnis.	4
2	Kommunikations-Mechanismen.	5
2.1	Service Data Objects	5
2.2	Process Data Objects	6
2.3	PDO-Mapping	7
2.4	Netzwerk-Management	8
2.5	Node-Guarding und Heartbeat	9
2.6	Emergency-Botschaften	9
3	Identifizier-Verteilung.	10
4	Gerätebeschreibung - EDS und DCF	12
5	Fazit	13
6	Referenzen	14
7	Dokument Versionen	15

1 Was ist CANopen?

Das CANopen Protokoll ist ein standardisiertes Schicht-7 Protokoll für den CAN Bus. Durch das Protokoll CANopen wird einerseits das „Wie“ der Kommunikation festgelegt, also mit welchen Telegrammen (d.h. Identifier) die Geräte angesprochen werden können. In CANopen sind Mechanismen zum Austausch von Prozessdaten in Echtzeit ebenso implementiert wie die Übertragung großer Datenmengen oder das Senden von Alarm-Telegrammen. Andererseits wird durch CANopen das „Was“ der Kommunikation festgelegt, das heißt ein Parameter zur Einstellung eines Gerätes wird über eine definierte Schnittstelle angesprochen (Geräteprofil oder Applikationsprofil).

Diese sogenannten CANopen-Profile sind in Tabellenform (Objektverzeichnis) organisiert. Allen Geräteprofilen gemeinsam ist das sogenannte „Kommunikationsprofil“ durch welches grundlegende Gerätedaten abgefragt bzw. eingestellt werden können. Zu diesen Daten zählen beispielsweise die Gerätebezeichnung, Hardware- und Software-Version, Fehlerstatus, verwendete CAN Identifier und viele weitere Parameter. Die Geräteprofile beschreiben die besonderen Fähigkeiten bzw. Parameter einer „Klasse“ von Geräten. Bislang wurden Geräteprofile definiert für digitale bzw. analoge E/A-Geräte, Antriebe, Sensoren und Regler, programmierbare Steuerungen, Encoder, Medizintechnik, Öffentlicher Nahverkehr, Batterien und Extrusionsanlagen, um nur einige zu nennen.

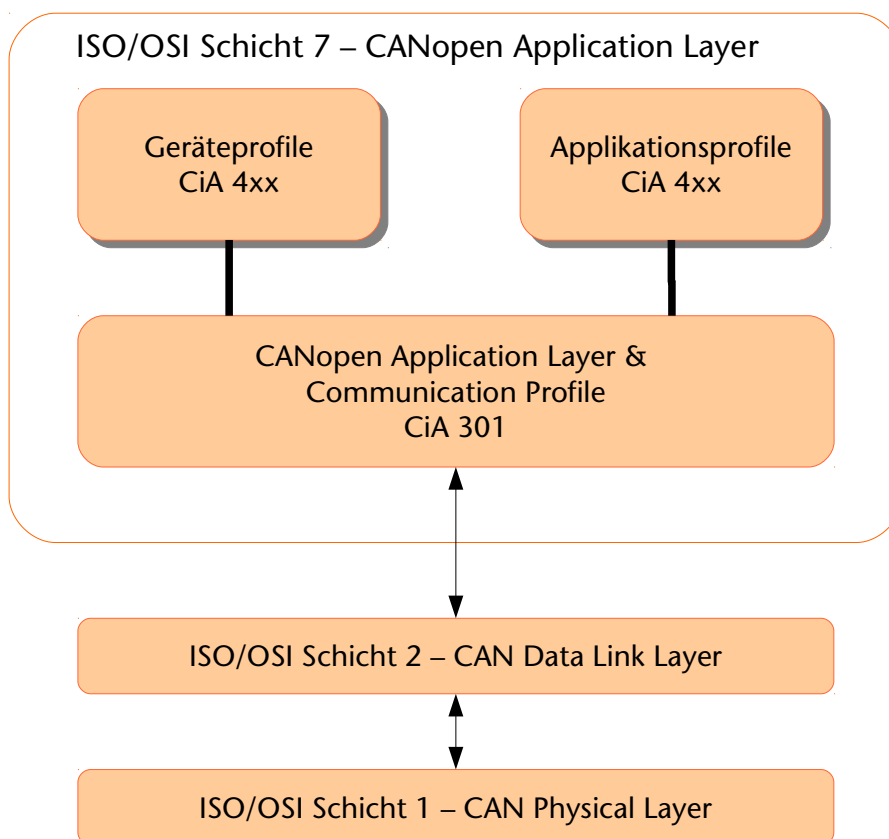


Abb. 1: Struktur der CANopen Geräteprofile

Die Geräteprofile setzen auf dem Kommunikationsprofil /1/ auf (siehe Abbildung 1). Der modulare Aufbau und das standardisierte Format hat für den Anwender zwei Vorteile: Erstens muß er sich bei der Anlagenkonzeption nur mit den Profilen auseinandersetzen, die für die Applikation relevant sind (z.B. digitale E/A Baugruppen). Zweitens kann man aus einer Vielzahl von am Markt verfügbaren Geräten eine Auswahl treffen, die alle in identischer Weise angesprochen werden und Prozessinformationen liefern.

1.1

Objektverzeichnis

Das Objektverzeichnis beschreibt den kompletten Funktionsumfang (Parameter) eines CANopen-Gerätes und ist in Tabellenform organisiert. Im Objektverzeichnis sind nicht nur die standardisierten Datentypen und Objekte des CANopen-Kommunikationsprofils sowie der Geräteprofile enthalten, sondern gegebenenfalls auch hersteller-spezifische Objekte und Datentypen. Die Adressierung der Einträge erfolgt mit Hilfe eines 16-Bit-Indizes (Reihenadresse der Tabelle, maximal 65536 Einträge) und eines 8-Bit-Subindizes (Spaltenadresse der Tabelle, maximal 256 Einträge). Somit lassen sich zusammengehörige Objekte leicht gruppieren. Die Struktur dieses CANopen Objektverzeichnisses ist in der folgenden Tabelle dargestellt:

Index (hex)	Objekt
0000	nicht verwendet
0001 - 001F	statische Datentypen
0020 - 003F	komplexe Datentypen
0040 - 005F	herstellerspezifische Datentypen
0060 - 007F	profilspezifische statische Datentypen
0080 - 009F	profilspezifische komplexe Datentypen
00A0 - 0FFF	reserviert
1000 - 1FFF	Kommunikationsprofil (CiA 301 & 302)
2000 - 5FFF	Herstellerspezifische Parameter
6000 - 9FFF	Parameter aus standardisierten Profilen (CiA 4xx)
A000 - AFFF	Netzwerk Variablen
B000 - FFFF	reserviert

Tabelle 1: Aufbau Objektverzeichnis

2 Kommunikations-Mechanismen

Die Kommunikation zwischen den Teilnehmern entspricht überwiegend dem Client-Server-Modell. Die Übertragung der Prozessdaten erfolgt nach dem Producer-Consumer-Modell.

2.1 Service Data Objects

Service Data Objects (SDOs) werden für Änderungen im Objektverzeichnis und für Statusabfragen verwendet. Jedes CANopen Gerät verfügt über mindestens einen SDO Kanal, dem zwei CAN-Identifizierer zugeordnet sind. Mit diesem Protokoll lassen sich Daten beliebiger Länge übertragen, wobei die Daten gegebenenfalls auf mehrere CAN-Nachrichten aufgeteilt (segmentiert) werden. In der ersten CAN-Nachricht des SDO sind vier der acht verfügbaren Bytes mit Protokollinformationen belegt. Für Zugriffe auf Objektverzeichniseinträge mit bis zu 4 Byte Länge (z.B. Integer16, Integer32, Float) genügt folglich eine einzige CAN-Nachricht (Expedited Transfer). Bei Datenlängen größer 4 Byte erfolgt eine segmentierte Übertragung, bei der alle auf die erste CAN-Nachricht folgenden Segmente des SDO jeweils 7 Byte Nutzdaten enthalten können. Das letzte Segment enthält eine Ende-Kennung. Ein SDO wird immer bestätigt übertragen, das heißt der Empfang einer jeden Nachricht wird durch den Empfänger quittiert. Mit Einführung der CANopen Spezifikation 4.0 ist auch ein beschleunigter SDO-Transfer möglich (Block Transfer). Bei diesem wird nicht mehr jedes Segment bestätigt, sondern es wird nur noch der Empfang einer Gruppe von Segmenten quittiert. Somit erhält man einen enormen Bandbreitenzuwachs für die Übertragung von großen Datenmengen.

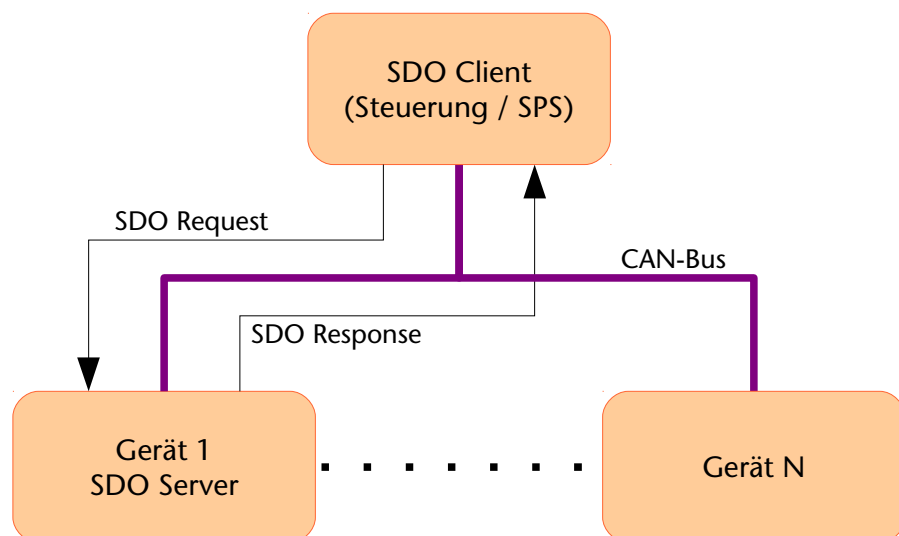


Abb. 2: SDO Client-Server Struktur

2.2 Process Data Objects

Für die Übertragung von Prozessdaten steht der Mechanismus des Process Data Object (PDO) zur Verfügung. Jedes CANopen-Gerät, welches Prozessdaten produziert und / oder konsumiert verfügt deshalb über mindestens ein PDO. Bei einem PDO stehen dem Anwender alle 8 Datenbytes einer CAN-Botschaft zur Verfügung. Die Übertragung einer PDO erfolgt unbestätigt, da letztendlich die CAN-Verbindungsschicht die fehlerfreie Übertragung einer Nachricht sicherstellt. Außerdem sind bestätigte Dienste in zeitkritischen Applikationen nicht wünschenswert, weil sie die Busbandbreite signifikant reduzieren. Somit sind PDOs "CAN pur", ohne den geringsten Protokoll-Overhead durch CANopen!

Die Übertragung der Prozessdaten kann auf verschiedene Arten durchgeführt werden:

Ereignis

Die Aussendung der PDO wird durch ein externes oder internes Ereignis gesteuert. Dieses Ereignis kann z.B. der Pegelwechsel eines digitalen Einganges sein oder aber der Ablauf eines Zeitgebers in dem Gerät.

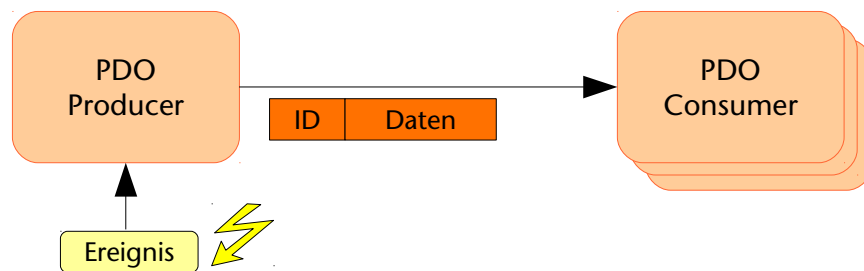


Abb. 3: Ereignis-gesteuerte PDO

Synchron

Bei der synchronen Übertragung werden durch einen Busteilnehmer Synchronisationstelegramme gesendet (Botschaft ohne Dateninhalt), auf deren Empfang hin ein PDO-Producer die Prozessdaten überträgt.

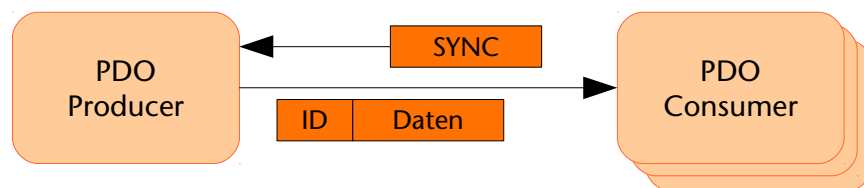


Abb. 4: Synchroner PDO Übertragung

2.3 PDO-Mapping

Wie bereits erläutert, stehen bei der Übertragung von Prozessdaten alle 8 Datenbytes zur Verfügung. Da keine Protokollinformationen vorhanden sind, muss das Format zwischen Producer und Consumer vereinbart werden - dies erfolgt durch das sogenannte PDO-Mapping. Es können nur die Daten übertragen werden, welche im Objektverzeichnis eines Gerätes verfügbar sind!

Bei dem festen Mapping sind die Prozessdaten in einer vordefinierten Reihenfolge in der PDO-Botschaft angeordnet. Durch den Anwender kann diese Anordnung nicht geändert werden, sie ist durch den Gerätehersteller vorgegeben.

Bei dem variablen Mapping können die Prozessdaten wahlfrei innerhalb der PDO-Botschaft angeordnet werden. Dazu werden von dem Eintrag aus dem Objektverzeichnis die Adresse (also Index und Sub-Index) sowie die Größe (Anzahl Bits) in das Mapping-Objekt eingetragen.

In dem folgenden Beispiel sind die Objekte "Error Register" ($1001_h:00_h$) und "Digital Input 1 - 8" ($6000_h:01_h$) in die Mapping-Tabelle eingetragen. Sobald die PDO gesendet werden muss, werden Daten der Objekte in die PDO Nachricht kopiert.

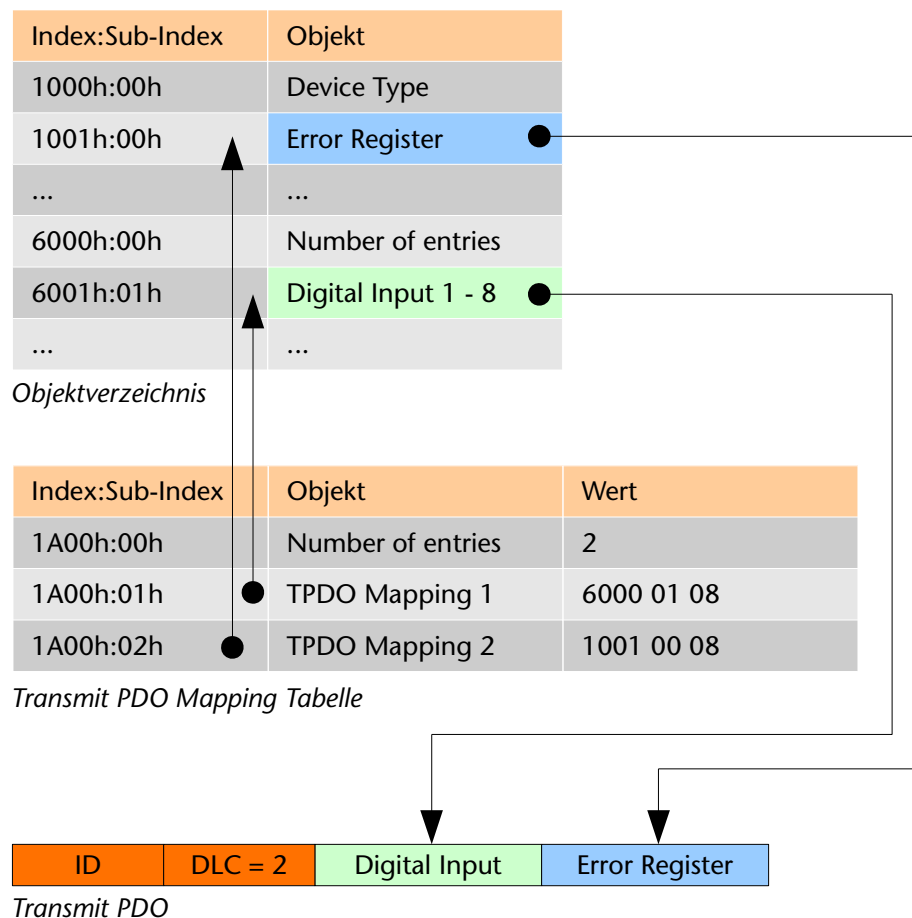


Abb. 5: Verwendung der PDO Mapping Tabelle

2.4 Netzwerk-Management

In einem CANopen-Netzwerk gibt es immer nur einen NMT-Master (NMT = Network Management), alle anderen Geräte sind NMT-Slaves. Der NMT-Master hat die komplette Kontrolle über alle Geräte und kann deren Zustand - im Sinne der Kommunikation - verändern.

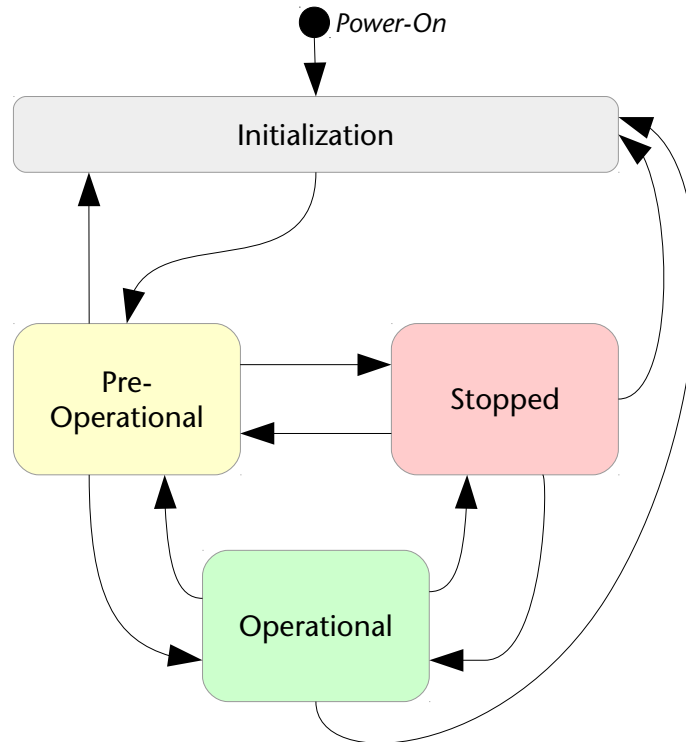


Abb. 6: NMT Zustandsdiagramm

Initialization	Dies ist der Zustand, den ein Knoten nach dem Einschalten durchläuft. Innerhalb dieser Phase erfolgt eine Initialisierung der Geräte-Applikation sowie der Geräte-Kommunikation (Bitrate und Knoten-Adresse). Anschließend geht der Knoten selbständig in den Zustand Pre-Operational.
Pre-Operational	In dem Zustand „Pre-Operational“ kann mit dem Knoten über SDOs kommuniziert werden. Der Knoten ist jedoch nicht in der Lage, eine PDO-Kommunikation durchzuführen.
Operational	In dem Zustand „Operational“ hat der CANopen-Knoten die volle Betriebsbereitschaft und kann selbständig Prozessdaten senden und empfangen.
Stopped	In dem Zustand "Stopped" wird ein Knoten vollständig vom Netz abgeschaltet, es sind weder SDO- noch PDO-Kommunikation möglich. Der Knoten kann nur durch ein entsprechendes Netzwerk-Kommando (z.B. Start-Node) in einen anderen Netzwerk-Zustand gebracht werden.

Um zu signalisieren, dass ein Gerät nach dem Einschalten betriebsbereit ist, wird eine sogenannte „Boot-Up Message“ ausgesendet. Diese Botschaft benutzt den Identifier des NMT-Error-Control Protokolls (Tabelle 3 auf Seite 10) und ist fest an die eingestellte Geräteadresse gebunden ($700_h + \text{Geräteadresse}$).

2.5 Node-Guarding und Heartbeat

Die Prüfung eines CAN-Knotens ist insbesondere dann gefordert, wenn dieser nicht ständig Botschaften sendet (zyklische PDOs). Zur Überwachung von CANopen-Knoten sind zwei Mechanismen vorhanden, die alternativ genutzt werden können. Beim Node-Guarding-Protokoll werden durch den NMT-Master Botschaften an die vorhandenen CANopen-Slaves gesendet, die auf diese innerhalb einer bestimmten Zeit antworten müssen. Der Ausfall eines Knotens wird somit nur durch den NMT-Master registriert. Zudem basiert Node-Guarding auf CAN-Remote Frames und sollte daher vermieden werden [/3/](#) [/4/](#). Daher ist dem Heartbeat-Protokoll, welches mit der CANopen Version 4.0 veröffentlicht wurde, der Vorzug zu geben. Hierbei sendet jeder Knoten autark eine Botschaft in zyklischen Abstand aus. Diese Nachricht kann von jedem anderen Teilnehmer im Netzwerk überwacht werden.

2.6 Emergency-Botschaften

Botschaften vom Typ "Emergency" werden verwendet, um Fehler eines Gerätes zu signalisieren. In dem Emergency-Telegramm wird ein Code übertragen, der den Fehler eindeutig identifiziert (definiert im Kommunikationsprofil CiA 301 [/1/](#) sowie in den jeweiligen Geräteprofilen CiA 40x).

Code (hex)	Bedeutung
00xx	Kein Fehler
10xx	Nicht definierter Fehlertyp
20xx	Stromfehler
30xx	Spannungsfehler
40xx	Temperaturfehler
50xx	Fehler an der Hardware
60xx	Fehler in der Software
70xx	Zusatzmodule
80xx	Kommunikation
90xx	Externer Fehler
FF00	Gerätespezifisch

Tabelle 2: Codes Für Emergency-Botschaft

Die Tabelle zeigt einen Auszug der verfügbaren Fehlercodes. Die Emergency-Botschaft wird von jedem CANopen-Gerät selbständig gesendet. Mit der aktuellen Version des CANopen Kommunikationsprofils kann die Aussendung einer Emergency-Botschaft auch abgeschaltet werden.

3 Identifizier-Verteilung

Grundsätzlich werden bei der Kommunikation über CANopen Identifier mit 11 Bit Länge (Standard Frames) verwendet. Die somit zur Verfügung stehende Menge von möglichen Identifiern ist durch das sogenannte Pre-defined Connection Set in diverse Bereiche aufgeteilt. Die Identifier-Verteilung ist so ausgelegt, dass in einem CANopen-Netzwerk maximal 127 Geräte vorhanden sind.

Identifier	Service	Richtung	COB-ID Berechnung	Hinweis
000 _h	NMT	Receive	-	fest
080 _h	SYNC	Rcv. / Trm	-	variabel, Index 1005 _h
081 _h .. 0FF _h	EMCY	Transmit	080 _h + Node-ID	variabel, Index 1014 _h
100 _h	TIME	Rcv. / Trm.	-	variabel, Index 1012 _h
181 _h .. 1FF _h	TPDO1	Transmit	180 _h + Node-ID	variabel, Index 1800 _h
201 _h .. 27F _h	RPDO1	Receive	200 _h + Node-ID	variabel, Index 1400 _h
281 _h .. 2FF _h	TPDO2	Transmit	280 _h + Node-ID	variabel, Index 1801 _h
301 _h .. 37F _h	RPDO2	Receive	300 _h + Node-ID	variabel, Index 1401 _h
381 _h .. 3FF _h	TPDO3	Transmit	380 _h + Node-ID	variabel, Index 1802 _h
401 _h .. 47F _h	RPDO3	Receive	400 _h + Node-ID	variabel, Index 1402 _h
481 _h .. 4FF _h	TPDO4	Transmit	480 _h + Node-ID	variabel, Index 1803 _h
501 _h .. 57F _h	RPDO4	Receive	500 _h + Node-ID	variabel, Index 1403 _h
581 _h .. 5FF _h	SDO	Transmit	580 _h + Node-ID	fest
601 _h .. 67F _h	SDO	Receive	600 _h + Node-ID	fest
701 _h .. 77F _h	NMT-EC	Transmit	700 _h + Node-ID	fest
7E4 _h .. 7E5 _h	LSS	Rcv. / Trm.	-	fest

Tabelle 3: Pre-defined Connection Set

Der Begriff Master bzw. Slave sollte hier nicht missverstanden werden: Selbstverständlich können im Operational-Modus alle Geräte vollkommen autark Nachrichten auf den Bus legen. Der "Master" im Netzwerk besitzt die Fähigkeit, die Betriebsart der "Slaves" zu ändern, er hat also die Kontrolle über das CANopen-Netzwerk. Vielfach wird der „Master“ daher auch als CANopen-Network-Manager bezeichnet. Typischerweise wird ein CANopen-Master durch eine SPS oder einen PC realisiert. Die CANopen-Slaves können die Adressen 1 bis 127 belegen. Durch die Geräteadresse (die bei vielen Geräten durch ein DIP-Schalter eingestellt wird) ergibt sich dann automatisch eine Anzahl von Identifiern, welche dieses Gerät dann belegt. Ein Gerät mit der Moduladresse 3 würde die folgenden Identifier belegen (siehe Berechnungsformel in Tabelle 3 auf Seite 10):

Identifier	Funktion	Akronym
000 _h	Netzwerkmanagement	NMT
080 _h	Synchronisations Botschaft	SYNC
083 _h	Emergency Botschaft	EMCY
183 _h	Transmit PDO 1	TPDO1
203 _h	Receive PDO 1	RPDO1
283 _h	Transmit PDO 2	TPDO2
303 _h	Receive PDO 2	RPDO2
383 _h	Transmit PDO 3	TPDO3
403 _h	Receive PDO 3	RPDO3
483 _h	Transmit PDO 4	TPDO4
503 _h	Receive PDO 4	RPDO4
583 _h	Transmit SDO	SDO Response
603 _h	Receive SDO	SDO Request
703 _h	Node Guarding / Heartbeat	NMT-EC

Tabelle 4: Nutzung der CAN-Identifier für ein CANopen Gerät mit Adresse 3

Diese Vorgabe kann nachträglich noch geändert werden. Für den Anwender hat das Pre-defined Connection Set aber den Vorteil, dass ein CANopen Netzwerk schnell aufgebaut werden kann und eine doppelte Vergabe von Identifiern ausgeschlossen ist.

4

Gerätebeschreibung - EDS und DCF

Das "Electronic Data Sheet" (EDS) /2/ beschreibt die Funktionalität eines CANopen-Gerätes in maschinen-lesbarer Form. In dem EDS sind alle Objekte aufgeführt, die unterstützten Bitraten, der Hersteller und viele weitere Angaben. Die EDS ist jedoch nur eine Schablone für das Gerät, denn es sind nicht die Werte eines Objektes enthalten.

Das "Device Configuration File" (DCF) ist identisch zur EDS aufgebaut und enthält zusätzlich die Werte eines jeden Objektes.

Das Format der beiden Beschreibungsdateien ist an das *.ini-Format von Windows angelehnt. Für den Anwender ist die Einbindung eines neuen Gerätes in das Netzwerk recht einfach: Hardware anschließen und die Diskette mit der zugehörigen EDS- oder DCF-Datei in den Master einlesen. Somit ist die Betriebsbereitschaft hergestellt.

```

.....

[MandatoryObjects]
SupportedObjects=3
1=0x1000
2=0x1001
3=0x1018

[1000]
ParameterName=DeviceType
ObjectType=0x7
DataType=0x7
AccessType=ro
DefaultValue=0x00020194
PDOMapping=0

[1001]
ParameterName=ErrorRegister
ObjectType=0x7
DataType=0x5
LowLimit=0x0
HighLimit=0xff
AccessType=ro
DefaultValue=0x0
PDOMapping=0

[1018]
SubNumber=5
ParameterName=Identity Object
ObjectType=0x9

[1018sub0]
ParameterName=Number of entries
ObjectType=0x7
DataType=0x5
.....

```

Abb. 7: Auszug aus einer EDS Datei

5 **Fazit**

CANopen wird aktiv von den Mitgliedern der internationalen Anwender- und Herstellervereinigung CAN in Automation e.V. (<http://www.can-cia.org>) weiterentwickelt. Die CANopen-Anwendungsschicht definiert verschiedene Kommunikationsdienste und Protokolle (z.B. Prozess- und Servicedaten) sowie ein Netzwerkmanagement.

CANopen wird hauptsächlich als eingebettetes Netzwerk in Maschinen, aber auch als generelles industrielles Kommunikationssystem verwendet. Es basiert auf den internationalen Normen ISO 11898-1 (CAN-Protokoll) und ISO 11898-2 (Schnelle physikalische Übertragungsschicht). Die CANopen Anwendungsschicht ist als EN 50325-4 von der CENELEC genormt.

Wenn Sie weitere Informationen zu CANopen suchen (oder sogar selbst an der Entwicklung von Profilen teilnehmen möchten), dann finden Sie auf der Homepage der CAN in Automation alle veröffentlichten Spezifikationen zum kostenlosen Download [/5/](#).

6

Referenzen

- /1/ CiA 301: CANopen application layer and communication profile, Version 4.2.0, CAN in Automation
<http://www.can-cia.org>
- /2/ CiA 306: CANopen electronic data sheet specification, Version 1.3.0, CAN in Automation
<http://www.can-cia.org>
- /3/ CiA 802: CANopen application note – CAN remote frames: Avoiding of usage, Version 1.1.0, CAN in Automation
<http://www.can-cia.org>
- /4/ MicroControl Blog: Finger weg von CAN Remote Frames!
<http://www.microcontrol-blog.net/2013/03/finger-weg-von-can-remote-frames/>
- /5/ CAN in Automation e.V. : CiA specifications
http://www.can-cia.org/index.php?id=specifications&no_cache=1

7 Dokument Versionen

<i>Revision</i>	<i>Datum</i>	<i>Beschreibung</i>
01	01.03.2000	Erste Version
02	18.08.2014	Überarbeitung gem. CiA 301, Version 4.2.0

Disclaimers

Life support — Products and software described in this application note are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. MicroControl customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify MicroControl for any damages resulting from such application.

Right to make changes — MicroControl reserves the right to make changes in the products - including circuits and/or software - described or contained herein in order to improve design and/or performance. MicroControl assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Copyright

No part of this application note may be copied, transmitted or stored in a retrieval system or reproduced in any way including, but not limited to, photography, magnetic, optic or other recording means, without prior written permission from MicroControl GmbH & Co. KG.

© 2014 MicroControl GmbH & Co. KG, Troisdorf

**MicroControl**
Systemhaus für Automatisierung

MicroControl GmbH & Co. KG
Junkersring 23
53844 Troisdorf
Germany
Fon: +49 / 2241 / 25 65 9 - 0
Fax: +49 / 2241 / 25 65 9 - 11
<http://www.microcontrol.net>