MicroControl
Systemhaus für Automatisierung

Einführung in CANopen

Autor	***:***	**Uwe Koppe**
Datum	***:***	**01.03.2000**
Veröffentlicht	***:***	
Schlüsselworte	***:***	**CANopen, Service Data Object, Process Data Object, Objektverzeichnis, EDS**

Das CANopen Protokoll ist ein standardisiertes Schicht-7 Protokoll für den CAN Bus. Diese Schicht basiert auf dem CAN Application Layer (CAL), welches als universelles Protokoll vom Steinbeis Transferzentrum für Prozessautomatisierung (stzp) entwickelt wurde. In der Praxis zeigt sich jedoch, daß Applikationen mit CAL für den Anwender zu komplex waren. Eine einheitliche, einfache Struktur für die Anbindung von CAN-Geräten der unterschiedlichen Hersteller sollte geschaffen werden: dies war die Geburtsstunde von CANopen im Jahr 1995.

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 (Profil).

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. Parmeter einer „Klasse“ von Geräten. Bislang wurden Geräteprofile definiert für digitale bzw. analoge E/A-Geräte, Antriebe, Bediengeräte, Sensoren und Regler, Programmierbare Steuerungen und Encoder. Viele weitere Profile (z.B. Medzintechnik, Marine, Öffentlicher Nahverkehr) sind in Vorbereitung.

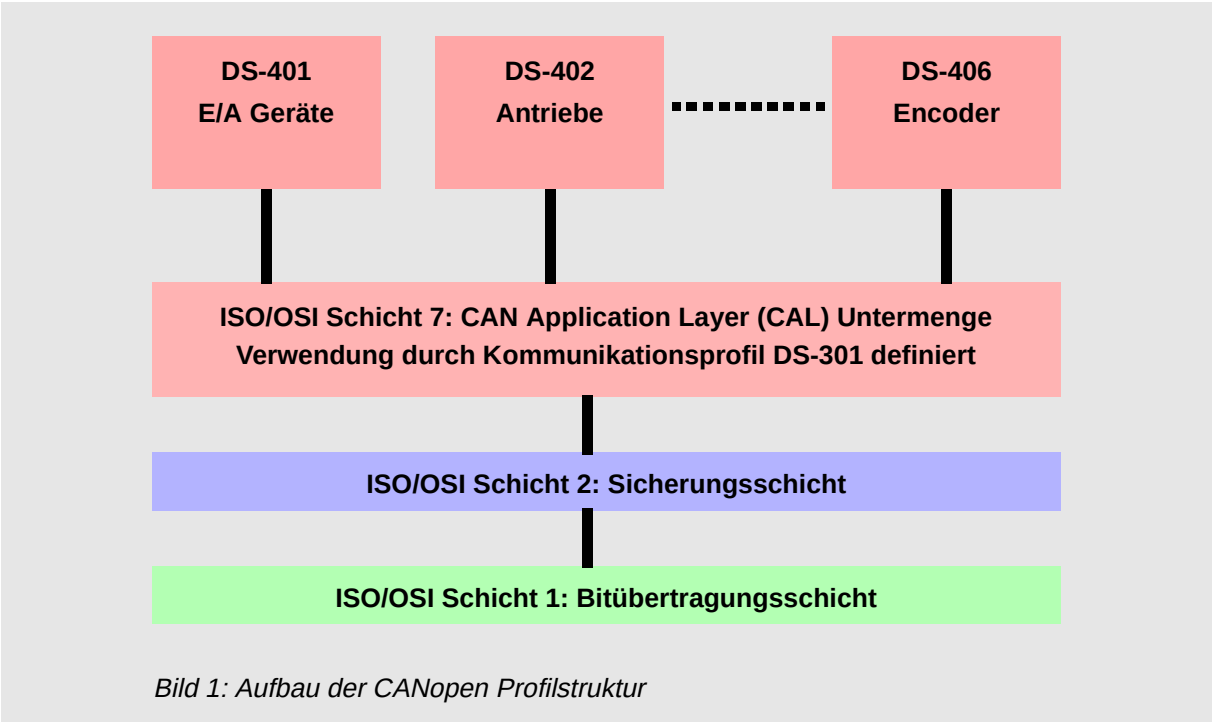


Bild 1: Aufbau der CANopen Profilstruktur

Die Geräteprofile setzen auf dem Kommunikationsprofil 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.

Objektverzeichnis

Das Objektverzeichnis beschreibt die komplette Funktionalität der CANopen-Geräte 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 (DS-301)
2000 - 5FFF	herstellerspezifische Parameter
6000 - 9FFF	Parameter aus den standardisierten Geräteprofilen
A000 - FFFF	reserviert

Tabelle 1: Aufbau Objektverzeichnis

Die Struktur des Objektverzeichnisses ist angelehnt an die Profile der Feldbusse Profibus und Interbus-S. Für die Objekte sind eine Reihe von Datentypen definiert. Aus diesen Standardtypen lassen sich u.a. Strukturen erstellen (ARRAY, RECORD), ähnlich wie in einer Programmiersprache.

Kommunikations-Mechanismen

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

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-Identifier zugeordnet sind. SDOs nutzen das "Multiplexed Domain Transfer Protocol" der CAL Spezifikation. 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, daß 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. 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.

Process Data Objects

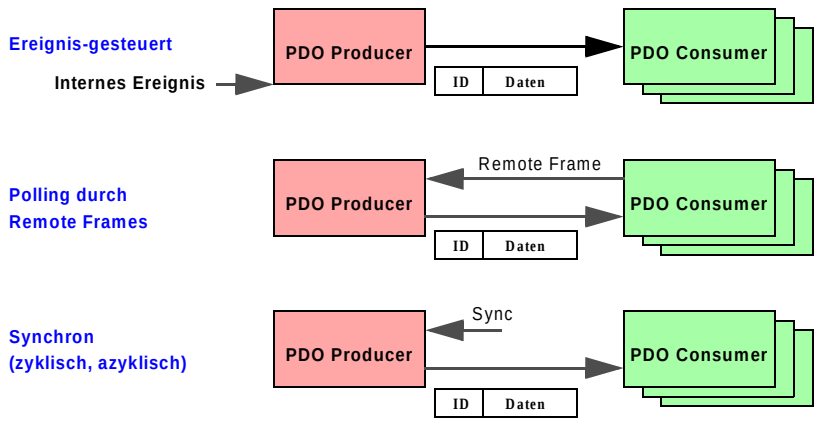
Für die Übertragung von Prozeßdaten steht der Mechanismus des Process Data Object (PDO) zur Verfügung. Jedes Prozeßdaten produzierende oder konsumierende CANopen-Gerät verfügt deshalb über mindestens ein PDO. PDOs verwenden das "Stored Event Protocol" des CAN Application Layers (siehe [1]). Dabei stehen 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 internes Ereignis gesteuert. Dieses Ereignis kann z.B. der Pegelwechsel eines digitalen Einganges sein oder aber der Ablauf eines Zeitgebers in dem Gerät.

- **Anforderung**
  In diesem Fall fordert ein anderer Busteilnehmer die Prozessdaten an, indem ein Remote-Transmission-Request gesendet wird.

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



Network 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 verändern. Es werden die folgenden Zustände unterschieden:

- **Initialization**
  Dies ist der Zustand, den ein Knoten nach dem Einschalten durchläuft. Innerhalb dieser Phase erfolgt ein Initialisierung der Geräte-Applikation sowie der Geräte-Kommunikation. Anschießend geht der Knoten selbständig in den Zustand Pre-Operational.

- **Pre-Operational**
  In diesem Zustand kann mit dem Knoten über SDOs kommuniziert werden. Der Knoten ist nicht in der Lage, eine PDO-Kommunikation durchzuführen und sendet auch keine Emergency-Botschaften aus.

- **Operational**
  In diesem Zustand hat der CANopen-Knoten die volle Betriebsbereitschaft und kann selbständig Nachrichten übertragen (PDOs, Emergency).

- **Prepared**
  In dem Zustand "Prepared" wird ein Knoten vollständig vom Netz abgeschaltet, es sind weder SDO- noch PDO-Kommunikation möglich. Der Knoten kann nur 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 Identifer des NMT-Error-Control Protokolls (siehe Tabelle 3) und ist fest an die eingestellte Geräteadresse gebunden (1792_{dez} + Geräteadresse).

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 DS-301 sowie in den jeweiligen Geräteprofilen DSP-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 4.0 des CANopen Kommunikationsprofils kann die Aussendung einer Emergency-Botschaft auch abgeschaltet werden.

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 anworten müssen. Der Ausfall eines Knotens wird somit nur durch den NMT-Master registriert. Der Ausfall des NMT-Masters würde zudem das gesamte Netzwerk lahmlegen. 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.

Identifier-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 128 Geräte vorhanden sind: ein NMT-Master und bis zu 127 NMT-Slaves.

Identifier	Funktion	Berechnung
0	Netzwerkmanagement (NMT)	-
1 .. 127	frei	-
128	Synchronisation (SYNC)	-
129 - 255	Emergency Message	128 + Modul-ID
256	Timestamp Message	-
257 .. 384	frei	-
385 .. 511	Transmit PDO 1	384 + Modul-ID
512	frei	-
513 .. 639	Receive PDO 1	512 + Modul-ID
640	frei	-
641 .. 767	Transmit PDO 2	640 + Modul-ID
768	frei	-
769 .. 895	Receive PDO 2	768 + Modul-ID
896	frei	-
897 - 1023	Transmit PDO 3	896 + Modul-ID
1024	frei	-
1025 - 1151	Receive PDO 3	1024 + Modul-ID
1152	frei	-
1153 - 1279	Transmit PDO 4	1152 + Modul-ID
1280	frei	-
1281 - 1407	Receive PDO 4	1280 + Modul-ID
1408	frei	-
1409 - 1535	Transmit SDO	1408 + Modul-ID
1536	frei	-
1537 - 1663	Receive SDO	1536 + Modul-ID

Tabelle 3: Belegung der Identifier

Identifier	Funktion	Berechnung
1664 - 1772	frei	-
1793 - 1919	NMT Error (Node Guarding, Heartbeat, Boot-Up)	1792 + Modul-ID
1920 - 2014	frei	-
2015 - 2031	NMT, LMT, DBT	-

Tabelle 3: Belegung der Identifier

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 Identifer belegen (siehe Berechnungsformel in Tabelle 3):

Identifier	Funktion	Richtung
0	Netzwerkmanagement (NMT)	Empfang
131	Emergency Message	Senden
387	Transmit PDO 1	Senden
515	Receive PDO 1	Empfang
643	Transmit PDO 2	Senden
771	Receive PDO 2	Empfang
899	Transmit PDO 3	Senden
1027	Receive PDO 3	Empfang
1155	Transmit PDO 4	Senden
1283	Receive PDO 4	Empfang
1411	Transmit SDO	Senden
1539	Receive SDO	Empfang
1795	Node Guarding	Senden

Tabelle 4: Belegung der Identifier für ein CANopen-Gerät mit Knoten-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.

PDO-Mapping

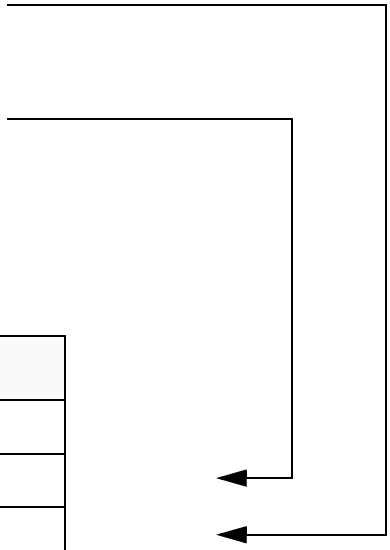
Wie bereits erläutert, stehen bei der Übertragung von Prozeßdaten alle 8 Datenbytes zur Verfügung. Da keine Protokollinformationen vorhanden sind, muß das Format zwischen Producer und Consumer vereinbart werden - dies erfolgt durch das sogenannte PDO-Mapping. Es können nur die Daten übertragen werden, welche im Objektverzeichnis eines Gerätes verfübar 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.Variables Mapping
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 werden die Objekte "Error Register" (1001_{hex}) und "Digital Input 1 - 8" (6000_{hex} ,1) in die Mapping-Tabelle eingetragen.

Index	Objekt
1000	Device Type
1001	Error Register
...	...
6000,0	Number of Entries
6000,1	Digital Input 1 - 8
...	...

Tabelle 5: Vom Objektverzeichnis des Gerätes ...



Index	Objekt	Wert
1A00,0	Number of Entries	2
1A00,1	Transmit PDO Mapping 1	6000 01 08
1A00,2	Transmit PDO Mapping 2	1001 00 08
...	...	

Tabelle 6: ... zum Eintrag in das PDO-Mapping

Gerätebeschreibung - EDS und DCF

Das "Electronic Data Sheet" (EDS) beschreibt die Funktionalität eines CANopen-Gerätes in maschinen-lesbarer Form. In dem EDS sind alle Objekte aufgeführt, die unterstützten Baudraten, 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.

Fazit

Neben dem Profibus hat sich in Europa CANopen in der Automatisierungstechnik und vielen Embedded-Control-Bereichen etabliert. Neben den bislang vorgestellten Standards wird an einer Vielzahl von weiteren Profilen gearbeitet. Wenn Sie als Leser sich für weitere Informationen zu CANopen interessieren (oder sogar selbst an der Entwicklung von Profilen teilnehmen möchten), dann sollten Sie die Homepage der Anwender- und Nutzervereinigung CiA (http://www.can-cia.de) besuchen. Informationen über spezielle Profile finden Sie auch auf der Homepage von MicroControl (http://www.MicroControl.net).

Weiterführende Literatur

[1] CAN Application Layer Spezifikation (CiA DS-201 bis DS-207), Version 1.1, Erlangen, Februar 1996
[2] CANopen Communication Profile (CiA DS-301), Version 4.0, Erlangen, Juni 1999