

# **Embedded Control**

**Dokumentation zu**

**CANopen**

**Januar 2006**

**Daniel Künzi**

**E3B**

**Dozent: Prof. Max Felser**

# Einleitung

CANopen ist ein auf CAN basierendes Schicht-7-Kommunikationsprotokoll, welches hauptsächlich in der Automatisierungstechnik verwendet wird. Es wurde von der CiA (CAN-in-Automation: Nutzer- und Herstellervereinigung) entwickelt und ist seit Ende 2002 als europäische Norm EN 50325-4 standardisiert.

CANopen verwendet als Übertragungstechnik die Schichten 1 und 2 des ursprünglich für den Einsatz im Automobil entwickelten CAN-Standards (ISO 11898-2). Diese werden in der Automatisierungstechnik durch die Empfehlungen des CiA Industrieverbandes hinsichtlich der Steckerbelegung, Übertragungsraten und der Anwendungsschicht erweitert. Der CANopen-Standard beschreibt den Austausch von Daten in einem CAN-basierenden Netzwerk. Dabei werden sowohl die grundlegenden Kommunikationsmechanismen (Kommunikationsprofil) als auch die Funktionalität der kommunizierenden Geräte (Geräteprofil) definiert. Das heißt, unter CANopen wird auch die Interpretation von Prozessdaten, die über den Bus übertragen werden festgelegt.

Ursprünglich wurde CANopen für bewegungsorientierte Geräte entwickelt und bietet eine hohe Konfigurationsflexibilität. Heute wird CANopen in den verschiedensten Gebieten eingesetzt. Dazu gehören unter anderen die Medizin, grössere Fahrzeuge, die Steuerungen auf Schiffen, öffentliche Verkehrsfahrzeuge und die Gebäudeautomation.



## Geschichte

### *Kurzgeschichte CAN*

Der CAN-Bus (Controller Area Network) gehört zu den Feldbussen. Es handelt sich dabei um ein asynchrones, serielles Bussystem, das seit **1983** von Bosch für die Vernetzung von Steuergeräten im Automobil entwickelt wird und **1985** zusammen mit Intel vorgestellt wurde, um die Kabelbäume (bis zu 2 km pro Fahrzeug) zu reduzieren und dadurch Gewicht zu sparen.



### *Geschichte CANopen*

Entwickelt wurden die Grundlagen dieses Standards in einem geförderten EU-Projekt unter der Führung von Bosch. Im Jahre **1995** wurden die bis dahin erstellten Spezifikationen an den CiA (CAN in Automation) übergeben. Seitdem wird der Standard im Rahmen der CAN-Nutzerorganisation

gepflegt und weiterentwickelt. Er ist seit **2002** als Europäische Norm (EN 50325-4 2002 Part 4: CANopen) verfügbar.

## Markt und Einsatzgebiete

### Anwendungsbereiche

CANopen erweitert die Vorteile von CAN - hohe Störsicherheit und tiefe Kosten – zusätzlich durch eine sehr flexible Struktur, die einen Einsatz in den unterschiedlichsten Anwendungsbereichen ermöglicht. Das Einsatzspektrum reicht dabei von der industriellen Automation über den medizintechnischen Bereich bis hin zu Anwendungen in Fahrzeugen.

In offenen Systemen bietet sich die einfache Erweiterung durch Fremdgeräte an, die CANopen unterstützen. Selbst in geschlossenen proprietären Netzen ist der Einsatz von CANopen sinnvoll. In diesem Fall dient CANopen quasi als Werkzeugkasten, dem man nur die passenden Kommunikationsmechanismen entnehmen muss.

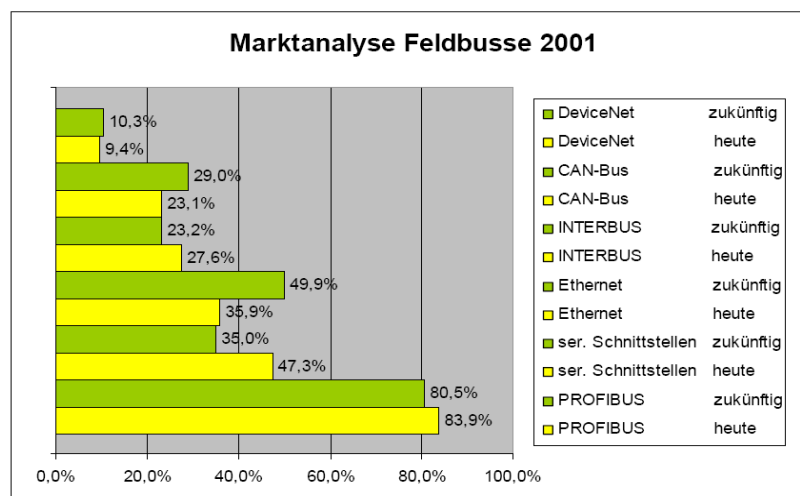
### Übersicht Anwendungsbereiche:

- **Industrie**
  - Antriebe
  - Bedienterminal
  - Industriewaagen
  - Steuerungen
- **Grosse Fahrzeuge, Schiffe, öffentlicher Nahverkehr**
- **Medizin**

### Anwendungsgebiete

- Automatisierungstechnik
- CAN-Embedded

### Markt



**Entwicklungstendenzen am Beispiel verkaufter Busbausteine, nach Angaben der Nutzerorganisationen (Orientierungswerte).**

## Anwendungsmodelle

Standardisierte Profile vereinfachen dem Entwickler die Arbeit ein CANopen-Netzwerk aufzubauen. Standardbausteine, Tools und Protokoll-Stacks sind zu einem vernünftigen Preis erhältlich. Für einen Entwickler eines Systems ist es wichtig, dass eine bereits erstellte Applikationssoftware wiederverwendet werden kann. Das bedeutet, dass nicht nur die Kommunikation standardisiert werden muss, sondern auch dass die Bausteine untereinander kompatibel und auswechselbar sind. Die CANopen Geräte- und Schnittstellenprofile wurden so festgelegt, dass ein Baustein ausgewechselt werden kann (zum Beispiel eine Motorsteuerung der Firma x durch eine der Firma y). CANopen ist sehr flexibel und offen. So kann zum Beispiel einer Grundfunktion eine herstellerspezifische Funktion im Geräteprofil hinzugefügt werden.

CANopen erleichtert dem Entwickler das Programmieren indem man sich nicht mehr um CAN-spezifischen Details wie Bittiming oder implementationsspezifischen Funktionen kümmern muss. Es werden standardisierte Kommunikationsobjekte für Echtzeitdaten, Konfigurationsdaten und Spezialfunktionen sowie Objekte für Netzwerkmanagement bereitgestellt.

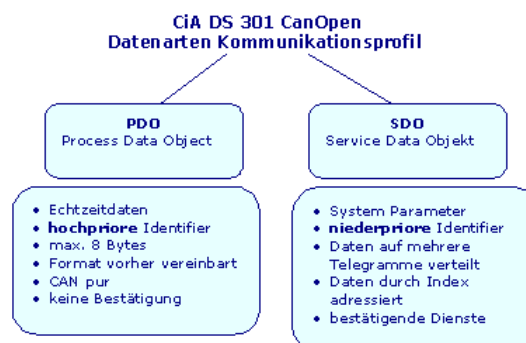
### CANopen Kommunikationsprofil

Das CANopen Kommunikationsprofil (dokumentiert in CiA DS-301) regelt wie die Geräte Daten miteinander austauschen. Wie bei allen anderen Feldbus-Protokollen werden hierbei Echtzeitdaten und Parameterdaten unterschieden. CANopen ordnet diesen, vom Charakter her völlig unterschiedlichen Datenarten, jeweils passende Kommunikationselemente zu.

### Prozess Daten Objekte (PDO)

Prozessdatenaustausch mit den CANopen PDOs ist „CAN pur“, also ohne Protokoll-Overhead. Die Broadcast-Eigenschaften von CAN bleiben voll erhalten. Eine Nachricht kann von allen Teilnehmern gleichzeitig empfangen und ausgewertet werden. Die Kommunikation mit den CANopen PDOs entspricht dem „Producer-Consumer“ Modell. Daten werden von ihrem „Produzent“ (beispielsweise einer Ein-Ausgabebaugruppe) auf den Bus gelegt und von einer beliebigen Anzahl von „Konsumenten“ empfangen und verarbeitet. Die PDOs werden im allgemeinen ereignisorientiert, zyklisch oder auf Anforderung als Broadcast-Objekte übertragen. Innerhalb eines PDO können maximal 8 Byte Daten übertragen werden. In Verbindung mit einer Synchronisationsnachricht kann das Senden sowie die Übernahme von PDOs netzwerkweit synchronisiert werden ("synchrone PDOs"). Damit lassen sich sowohl die Busbelastung als auch die Reaktionszeit des Netzes auf ein Minimum reduzieren. CANopen erreicht eine hohe Kommunikationsleistung bei vergleichsweise geringer Baudrate. Die Zuordnung von Anwendungsobjekten auf ein PDO ist über eine, im Objektverzeichnis abgelegte Strukturbeschreibung ("PDO-Mapping") einstellbar und damit den jeweiligen Einsatzanforderungen eines Gerätes anpassbar.

### Beispiel:

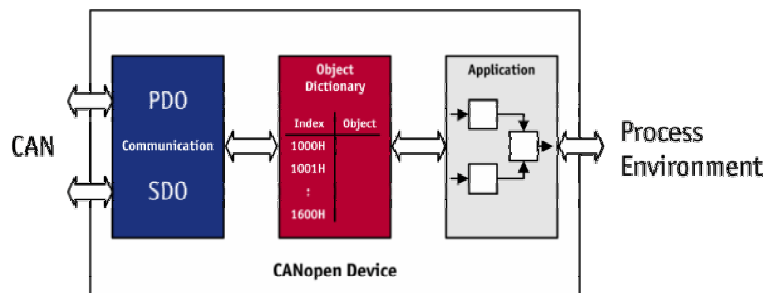


## Service Daten Objekte (SDO)

Die SDOs dienen in erster Linie zur Übertragung von Parametern während der Gerätekonfiguration sowie allgemein zur Übertragung längerer Datenbereiche. Die Übertragung von SDOs erfolgt als bestätigter Datentransfer mit jeweils zwei CAN-Objekten in Form einer Punkt-zu-Punkt Verbindung zwischen zwei Netzknoten. Die Adressierung des entsprechenden Objektverzeichniseintrages erfolgt durch Angabe von Index und Subindex des Eintrags. Es können Nachrichten unbegrenzter Länge übertragen werden, was allerdings mit einem zusätzlichen Protokolloverhead verbunden ist.

## Objektverzeichnis

Das Objektverzeichnis (OV) strukturiert die Daten eines CANopen Gerätes in einer übersichtlichen tabellarischen Anordnung. Es enthält sämtliche Geräteparameter als auch alle aktuellen Prozessdaten, die damit auch über das SDO zugänglich sind. Das Objektverzeichnis ist unterteilt in einen Bereich, welcher allgemeine Angaben über das Gerät, wie Geräteidentifikation, Herstellername, etc. sowie Kommunikationsparameter enthält, sowie einen Teil, der die spezifische Gerätefunktionalität beschreibt.



Die Identifizierung eines Eintrags ("Objekt") des Objektverzeichnisses erfolgt über einen 16-Bit Index und einen 8-Bit Subindex. Über die Einträge des Objektverzeichnisses werden die "Anwendungsobjekte" eines Gerätes, wie z.B. Ein- und Ausgangssignale, Geräteparameter, Gerätefunktionen oder Netzwerkvariablen in standardisierter Form über das Netzwerk zugänglich gemacht.

Index (hex)	Objekt
0000	Reserved
0001-001F	Static Data Types
0020-003F	Complex Data Types
0040-005F	Manufacturer Specific Data Types
0060-007F	Device Profile Specific Static Data Types
0080-009F	Device Profile Specific Complex Data Types
00A0-0FFF	Reserved for further use
1000-1FFF	Communication Profile Area
2000-5FFF	Manufacturer Specific Profile Area
6000-9FFF	Standardized Device Profile Area
A000-FFFF	Reserved for further use

## **CANopen Default Identifier**

CANopen-Geräte können ohne Konfiguration in ein CANopen-Netzwerk eingesetzt werden. Lediglich die Einstellung einer Busadresse und der Baudrate ist erforderlich. Aus dieser Knotenadresse leitet sich die Identifizierungsnummer für die Kommunikationskanäle ab.

## **CANopen – Geräteprofile**

Die CANopen Geräteprofile beschreiben das „was“ der Kommunikation. In ihnen wird die Bedeutung der übertragenen Daten eindeutig und hersteller-unabhängig festgelegt. So lassen sich die Grundfunktionen einer jeden Geräteklasse (z.B. Ein- Ausgabebaugruppen, Servoantriebe oder Frequenzumrichter) einheitlich ansprechen. Für alle wichtigen, in der industriellen Automatisierungstechnik eingesetzten Gerätetypen, wie digitale und analoge Ein/Ausgabemodule, Antriebe, Regler, programmierbare Steuerungen, oder Encoder sind in der CiA Geräteprofile erarbeitet worden. In Geräteprofilen werden sowohl Funktionalität, als auch Parameter von Standardgeräten des jeweiligen Typs festgelegt. Auf der Grundlage dieser standardisierten Profile kann auf identische Art und Weise über den Bus auf CANopen Geräte zugegriffen werden. Damit sind Geräte, die dem gleichen Geräteprofil folgen, weitgehend untereinander austauschbar. Obwohl bereits viele optionale Parameter in den Profilspezifikationen beschrieben sind, bieten die Geräteprofile zusätzlich Raum für herstellereigene Funktionserweiterungen.

## **Netzwerkmanagement**

Zur Kontrolle des Gerätezustandes steht eine vollständige Netzwerkmanagement Funktionalität (NMT) zur Verfügung, welche auf der Basis einer Master-Slave Beziehung realisiert ist. Für die Überwachung der Kommunikationsfähigkeit von CANopen Geräten sind weiterhin zwei alternative Dienste in Form von "Node-Guarding" beziehungsweise "Heartbeat-Nachrichten" spezifiziert.

## **Elektronisches Gerätedatenblatt – EDS Datei**

Funktionalität und Eigenschaften eines CANopen-Gerätes werden vom Gerätehersteller in Form von einem standardisierten elektronischen Datenblatt (Electronic Data Sheet, EDS) im ASCII-Format beschrieben. Hierbei ist das EDS als eine Art Formblatt zu verstehen, welches alle Daten und Funktionalitäten eines Gerätes wie sie über das Netzwerk zugänglich sind beschreibt. Die tatsächliche Geräteeinstellung wird mit dem sog. "Device Configuration File" (DCF) beschrieben. EDS und DCF können in Form eines Datenträgers, abrufbar über Internet oder im Gerät gespeichert zur Verfügung gestellt werden.

## **Technologie**

Layer	Name	Beschreibung	Definition
7	Application Layer	Anwendungsschicht	<b>CANopen</b>
6	Presentation Layer	Darstellungsschicht	
5	Session Layer	Kommunikationsschicht	
4	Transport Layer	Transportschicht	
3	Network Layer	Vermittlungsschicht	
2	Link Layer	Sicherungsschicht	<b>CAN</b>
1	Physical Layer	Bitübertragungsschicht	

Wie bereits erwähnt basiert CANopen auf CAN welches die Bitübertragungsschicht (Layer 1) und die Sicherungsschicht (Layer 2) im ISO/OSI-Modell definiert.

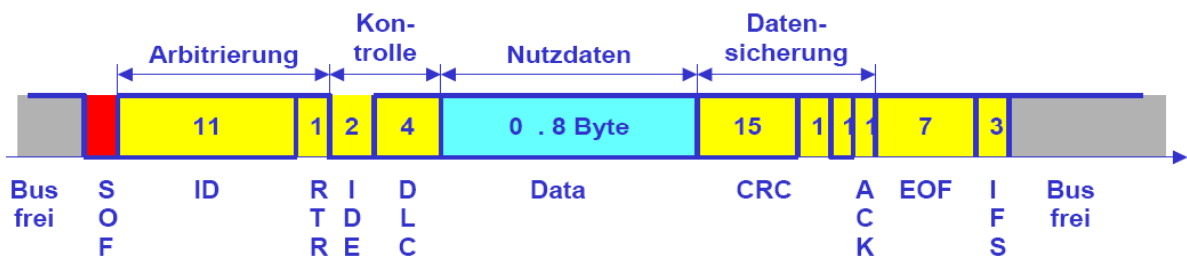
## Sicherungsschicht (Layer 2)

### Rahmenstruktur

Bei CAN wird jede zu übertragende Nachricht über eine Nachrichtenennung (Identifier) eindeutig gekennzeichnet. Im Gegensatz zur Teilnehmeradressierung wird dabei kein Steuergerät, sondern die Nachricht selbst adressiert. Dadurch steht eine Nachricht grundsätzlich jedem Busteilnehmer zum Empfang zur Verfügung. Die Übernahme einer Nachricht hängt einzig von der Entscheidung der Steuergeräte ab (empfängerselektives System). Diese Entscheidung wird über sogenannte Akzeptanz- bzw. Nachrichtenfilter durchgeführt. Es ist somit möglich, dass eine Nachricht von einem, mehreren oder allen Steuergeräten zur Weiterverarbeitung übernommen wird.

Im Standard-Format umfasst die Nachrichtenadresse 11 Bits. Daher können in einem CAN-Netzwerk 2048 verschiedene CAN-Botschaften voneinander unterschieden werden. Neben dem Standard-Format existiert ein Extended-Format. Dieses Format sieht 29 Bits für die Nachrichtenadresse vor. Es können dann über 536 Millionen CAN-Botschaften spezifiziert werden.

- **Standard Format (Version 2.0 A)**



- **Extended Format (Version 2.0 B):**

Erweiterung des Identifiers von 11 auf 29 Bit. Kennzeichnung über das IDE Bit.

<b>SOF</b>	Start of Frame	Startbit
<b>ID</b>	Identifier	Identifier
<b>RTR</b>	Remote Transmission Request	Anforderungstelegramm
<b>IDE</b>	Identifier Extension	Extended Format (Version 2.0 B)
<b>DLC</b>	Data Length Code	Anzahl der Nutzdaten in Byte
<b>DATA</b>	Data	Nutzdaten
<b>CRC</b>	Cyclic Redundancy Check	Prüfsumme
<b>ACK</b>	Acknowledge	Empfangsbestätigung
<b>EOF</b>	End of Frame	Stopbit
<b>IFS</b>	Inter-frame Spacing	Anzahl der Bits die aufeinanderfolgende Botschaften trennt

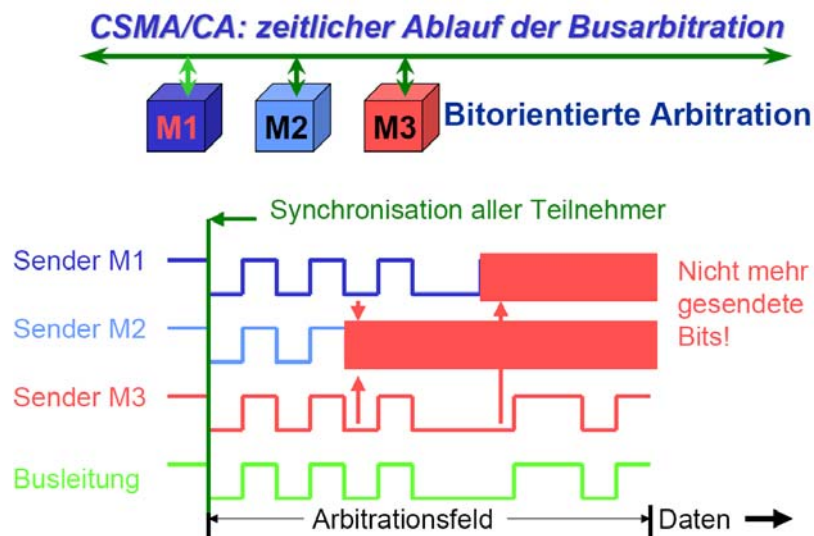
CAN basiert auf einer Multi-Master-Architektur: alle Steuergeräte sind gleichberechtigt und besitzen die Masterfunktionalität. Der große Vorteil liegt darin, dass jedes Steuergerät aus eigener Kraft das gemeinsame Übertragungsmedium belegen kann. Somit ist jedes Steuergerät in der Lage, bei Bedarf bzw. bei Ereignis eine entsprechende CAN-Nachricht abzuschicken. Zudem wirkt sich die Multi-Master-Architektur positiv auf die Verfügbarkeit aus: der Ausfall eines Steuergerätes zieht keinen Totalausfall des Bussystems nach sich.



## Zugriffsverfahren

Einem CAN-Netzwerk liegt eine Buszugriffskontrolle in Form des CSMA/CA-Verfahrens zugrunde. Dabei handelt es sich um ein zufälliges, zerstörungsfreies, prioritätengesteuertes, dezentrales Buszugriffsverfahren. Im Rahmen einer bitweisen Arbitrierungsphase wird entschieden, welches sendebereite Steuergerät seine CAN-Botschaft übertragen kann. Eine wesentliche Rolle spielt dabei die Nachrichtenadresse.

Über die Nachrichtenadresse ist einer CAN-Botschaft eine individuelle Priorität zugeordnet. Im Falle eines simultanen Buszugriffs wird sich jenes Steuergerät am Bus durchsetzen, welches die CAN-Botschaft mit der höchsten Priorität absetzen möchte. Diese Methode garantiert die Übertragung der CAN-Botschaft mit der höchsten Priorität. Zu jedem Zeitpunkt wird demnach immer die höchstpriorste CAN-Botschaft übertragen.



## Priorisierung

Die Priorisierung von CAN-Botschaften ist die Basis für die Sicherstellung geringer Latenzzeiten. Je höher die Priorität einer CAN-Botschaft, desto weniger Zeit vergeht, bis das entsprechende Steuergerät diese auf dem Bus übertragen kann. In Verbindung mit einer maximal einstellbaren Baudrate (1 Mbaud im Entfernungsbereich bis zu 40 Meter) und der begrenzten maximalen Datenblocklänge einer CAN-Botschaft kann eine maximale Latenzzeit für die höchstpriorste CAN-Botschaft von ca. 150 Mikrosekunden garantiert werden. Die maximale Datensegmentlänge einer CAN-Botschaft ist auf acht Byte begrenzt. Die kurze Datenblocklänge ist in Verbindung mit den Datensicherungsmaßnahmen Voraussetzung für eine Datenübertragung auch in stark gestörter Umgebung sowie eine kurze Latenzzeit sendebereiter CAN-Botschaften.

## Fehlerbehandlung

Da CAN für den Einsatz in stark elektromagnetisch verseuchter Umgebung vorgesehen ist, sieht das CAN-Protokoll eine Reihe von Maßnahmen vor, die der Erkennung von Übertragungsfehlern beitragen. Stellt ein Steuergerät einen Übertragungsfehler fest, wird die laufende Übertragung durch Aussendung eines Fehlerflags (Error-Flag) abgebrochen und eine Wiederholung der Übertragung der betroffenen CAN-Botschaft eingeleitet.

Diese Form der Fehlersignalisierung trägt zu sehr kurzen Fehlererholzeiten bei. Im optimalsten Fall wird eine abgebrochene CAN-Botschaft nach spätestens 31 Bitzeiten vom entsprechenden Steuergerät wiederholt. Falls allerdings zwischenzeitlich eine CAN-Botschaft höherer Priorität sendebereit geworden ist, wird diese eingefügt, und die Fehlererholzeit wird länger.



Über das bei einem lokalen Fehler ausgesendete Error-Flag wird die Übernahme der CAN-Botschaft durch andere Steuergeräte verhindert und somit netzweite Datenkonsistenz sichergestellt. Um eine möglichst hohe Verfügbarkeit des Busses zu gewährleisten, beinhaltet das CAN-Protokoll einen Mechanismus zur automatischen Fehlereingrenzung bis hin zur Abschaltung defekter Steuergeräte.

## Installationstechnik

### **Bitübertragungsschicht (Layer 1)**

In einem CANopen Netzwerk können maximal 127 Teilnehmer angeschlossen werden. Die Übertragungsgeschwindigkeit ist im Bereich von 10 Kbit/s bis 1 Mbit/s in festgelegten Stufen einstellbar. Die Länge eines CANopen Netzwerkes ist abhängig von der Übertragungsgeschwindigkeit und in der nachfolgenden Tabelle dargestellt:

Bitrate (kbit/s)	Buslänge (m)
1000	30
500	100
250	250
125	500
62,5	1000

### **Bitcodierung**

Das CAN-Protokoll nutzt die NRZ-Codierung (Non-Return-to Zero), die eine maximale Effizienz bei der Bitcodierung gewährleistet (Nachteil: mit DC-Anteil).

Die elektrische Störsicherheit wird unter anderem dadurch erreicht, dass ein Bit auf zwei Leitungen gleichzeitig mit einer gegensinnigen Potenzialänderung abgebildet wird. Man spricht hier auch von einem differentiellen Signal.

Auf einer zweiten Leitung wird also eine redundant invertierte Übertragung des logischen Signals vorgenommen. In die Leitung eingestreute Störungen wirken auf beide Leitungen in der gleichen Richtung. Da die beiden differentiellen Leitungen jedoch immer gegensinnige Pegel haben, bleibt die Differenz der Pegel auch bei Störungen weitgehend erhalten.

Die Leitung CAN-High und CAN-Low enthalten das invertierte und das nicht invertierte serielle Datensignal.

Durch die Ausführung als Open-Collector können ausserdem mehrere Teilnehmer auf dem Bus parallelgeschaltet werden, ohne dass im Konfliktfall elektrische Kurzschlüsse entstehen.

Der Zustand mit zwei unterschiedlichen Pegeln auf CAN-H und CAN-L wird als der dominante Zustand genannt; der Zustand mit zwei gleichen Pegeln wird als rezessiv bezeichnet.

Optional kann auch eine 24 Volt Versorgungsspannung mitgeführt werden.

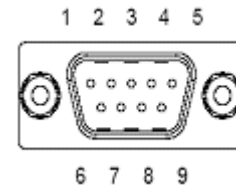
### **Installationstechnik**

Das CANopen System wird in Bustopologie mit Abschlusswiderständen (120 Ohm) am Anfang und am Ende verkabelt. Stichleitungen sollten möglichst vermieden werden. Das Kabel ist als geschirmtes Twisted Pair Kabel auszuführen und sollte eine Impedanz von 120 Ohm und einen Widerstand von 70 mOhm/m haben.

Der Anschluss der Teilnehmer an das CANopen Netzwerk erfolgt meist über 9 polige D-Sub Stecker.

Die Steckerbelegung lautet wie folgt:

Pin	Signal	Beschreibung
1	---	Reserviert
2	CAN_L	CAN_L Bussignal (dominant low)
3	CAN_V (-)	Externe Busspeisung Minus (optional)
4	---	Reserviert
5	CAN_SHLD	CAN Kabelabschirmung
6	CAN_V (-)	Externe Busspeisung Minus (optional)
7	CAN_H	CAN_H Bussignal (dominant high)
8	---	Reserviert
9	CAN_V (+)	Externe Busspeisung Plus (optional)
Gehäuse	CAN_SHLD	CAN Kabelabschirmung



**D-Sub Stecker**

Auf dem Markt sind verschiedene Schnittstellenkarten erhältlich: für PCI, ISA, PC-104 und USB. Beispiele:



### Kontrollwerkzeuge

Mit den Schnittstellenkarten kann der CAN-Bus passiv „abgehört“ werden. Man braucht also nur noch eine Analysesoftware auf einem PC um das Netzwerk zu testen.

## Quellenverzeichnis

<a href="http://www.can-cia.org/canopen/">http://www.can-cia.org/canopen/</a>	CAN-in-Automation
<a href="http://de.wikipedia.org/wiki/CANopen">http://de.wikipedia.org/wiki/CANopen</a>	Wikipedia - freie Enzyklopädie
<a href="http://www.canopensolutions.com/">http://www.canopensolutions.com/</a>	Infoseite der IXXAT Automation GmbH
<a href="http://www.feldbusse.de/CanOpen/canopen.htm">http://www.feldbusse.de/CanOpen/canopen.htm</a>	Infoseite von HMS Industrial Networks
<a href="http://www.vector-informatik.de">http://www.vector-informatik.de</a>	Vector Informatik: Softwarehersteller
<a href="http://www.emg.ing.tu-bs.de/pdf/IKF/ikf_folien_SS04.pdf">http://www.emg.ing.tu-bs.de/pdf/IKF/ikf_folien_SS04.pdf</a>	Vorlesungsfolien der TU Braunschweig