

# Grundlagen zum CAN Bus

## Allgemeine Einführung

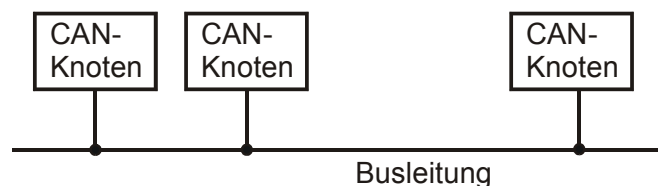
Schnittstellen dienen zur Übertragung von Informationen zwischen den einzelnen Komponenten eines Systems.

In einem Bussystem werden alle Komponenten über kurze Stichleitungen an eine gemeinsame Datenleitung angeschlossen.

Der Aufwand für die Verkabelung wird dadurch minimiert, und es können leicht zusätzliche Komponenten angeschlossen werden.

Der Datenfluss muss jedoch über ein Zugriffsverfahren (Protokoll) gesteuert werden, wenn alle Komponenten eine gemeinsame Busleitung benutzen. Dabei sollen möglichst auch Komponenten unterschiedlicher Hersteller zusammenarbeiten. Dies nennt man offenes Bussystem.

Das Controller Area Network (CAN) verbindet mehrere gleichberechtigte Komponenten (Knoten, Node) über einen 2-Draht Bus miteinander. Das CAN-Protokoll wurde 1983 von Bosch für den Einsatz in Kraftfahrzeugen entwickelt.



Aufgrund der hohen Störsicherheit, der geringen Kosten und der Echtzeitfähigkeit wird CAN auch in der Automatisierungstechnik, vor allem in Textilmaschinen, Aufzugssteuerungen und in Landmaschinen eingesetzt.

Die Organisation „Can in Automation“ (CiA) widmet sich der Weiterentwicklung des CAN Protokolls.

## Physikalische Beschreibung der CAN Schnittstelle

Die physikalischen Gegebenheiten für CAN und viele andere Busprotokolle, wie z.B. Profibus, sind in der ISO 11898 definiert. Zur Umsetzung dieser Spezifikation stehen viele Microchips zur Verfügung, wie z.B. der PCA82C250 von Philips.

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.

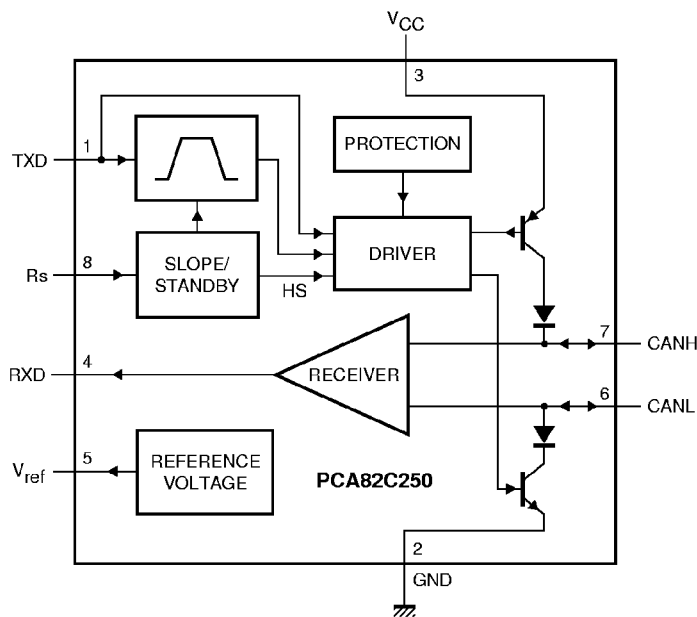
In die Leitung eingestreute Störungen wirken auf beide Leitungen in der gleichen Richtung. Da die beiden differentiellen Leitungen jedoch immer gegensinnige Pegel haben, bleibt die Differenz der Pegel auch bei Störungen weitgehend erhalten. Dies nennt man Gleichtaktunterdrückung, auf englisch „Common Mode Rejection Ratio“ (CMRR). Die

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

Durch die Ausführung als offener Collector (PNP auf VCC bei CAN-H und NPN auf GND bei CAN-L) 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 (Pegeldifferenz > 1,5 Volt); der Zustand mit zwei gleichen Pegeln wird als rezessiv bezeichnet (Pegeldifferenz < 1,0 Volt).

Der dominante Zustand entspricht per CAN Definition einer logischen Null: Legt ein Knoten eine logische Null auf den Bus, überschreibt er möglicherweise den Zustand einer logischen Eins eines anderen Knotens. Die Kopplung der Knoten über die Busleitung stellt eine logische Und-Verknüpfung dar (Wired-And).



logischer Pegel	Zustand	CAN-H	CAN-L	Differenzpegel
0	dominant	Transistor durchgeschaltet (zieht Pegel auf VCC)	Transistor durchgeschaltet (zieht Pegel auf GND)	>1,5 Volt
1 oder hochohmig	rezessiv oder floating	Transistor gesperrt	Transistor gesperrt	0 Volt

Eine weitere Maßnahme zur Erhöhung der Störsicherheit ist das NRZI-Verfahren: Dabei wird nach maximal fünf Bits gleicher „Polarität“ (rezessiv bzw. dominant) ein Bit der jeweils anderen Polarität eingefügt.

## Steckerbelegung

Als Steckverbinder für den CANbus hat sich der vom CiA vorgeschlagene 9 polige Sub-D Stecker durchgesetzt. In den Knoten werden sowohl weibliche wie auch männliche Steckverbinder gleichzeitig eingesetzt. Somit können ohne Unterbrechung weitere Knoten

in die Busleitung integriert werden.

Für die Übertragung von CAN Signalen ist mindestens ein 3 poliges Kabel mit CAN-High, CAN-Low und Ground erforderlich. Die Verwendung eines geschirmten Kabels ist nicht vorgeschrieben. Bei größeren Leitungslängen sind Verdrillung des Leitungspaares und Schirmung zu empfehlen.

Die Organisation CiA definierte daneben auch noch einen 5 poligen Rundsteckverbinder und einen 4- oder 5-poligen "open style" Stecker, ohne auf die Abmessungen einzugehen. Eine ebenfalls nützliche Sache ist der von CiA definierte 10 polige "multipole Connector". Dahinter steckt eine 10 polige Doppelpfostenleiste. Pin 10 ist dabei reserviert und bleibt frei.

PIN	Signal	Beschreibung
1		reserviert
2	CAN-L	negiertes CAN-Signal (Dominant Low)
3	CAN-GND	Masse
4		reserviert
5	CAN-SHLD	Schirmung (optional)
6	GND	Geräte Masse (optional)
7	CAN-H	positives CAN Signal (Dominant High)
8		reserviert
9	VCC	Versorgungsspannung (optional)

## Bitrate und Leitungslängen

Das CAN Netzwerk kann prinzipiell Bitraten bis zu 1Mbit/s übertragen. Alle CAN-Knoten müssen die Nachricht gleichzeitig verarbeiten können. Die maximale Kabel-Länge ist daher abhängig von der Bitrate. Die Tabelle zeigt empfohlene Bitraten und die entsprechende maximale Kabellänge.

Bitrate	Kabellänge
10 kbits/s	6,7 km
20 kbits/s	3,3 km
50 kbits/s	1,3 km
125 kbits/s	530 m
250 kbits/s	270 m
500 kbits/s	130 m
1 Mbits/s	40 m

## Busterminierung (Abschlusswiderstand)

Die Busterminierung erfolgt beim CAN Bus mit 120 Ohm. Eine Terminierung ist auch schon bei kurzen Leitungen mit niedrigen Baudraten erforderlich. Ohne Terminierung gibt es Reflexionen. In der Praxis reicht bei kurzen Leitungen eine Terminierung an einem Ende, idealerweise wird der Bus aber an beiden Enden (und nur dort) mit jeweils 120 Ohm terminiert.

## Prinzip des Datenaustausches im CAN Netzwerk

Bei der Datenübertragung in einem CAN Bus werden keine Knoten adressiert, sondern der Inhalt einer Nachricht (z.B. Drehzahl oder Motortemperatur) wird durch einen eindeutigen Identifier gekennzeichnet. Neben der Inhaltskennzeichnung legt der Identifier auch die Priorität der Nachricht fest.

Mit der dann folgenden Akzeptanzprüfung stellen alle Stationen nach korrektem Empfang der Nachricht anhand des Identifiers fest, ob die empfangenen Daten für sie relevant sind oder nicht. Durch die inhaltsbezogene Adressierung wird eine hohe Flexibilität erreicht: Es lassen sich sehr einfach Stationen zum bestehenden CAN-Netz hinzufügen.

Außerdem ergibt sich die Möglichkeit des Multicasting: Eine Nachricht kann von mehreren Teilnehmern gleichzeitig empfangen und ausgewertet werden. Messgrößen, die von mehreren Steuergeräten als Information benötigt werden, können über das CAN-Netz so verteilt werden, dass nicht jedes Steuergerät einen eigenen Sensor benötigt.

## Kollisionsprüfung

Jeder Teilnehmer darf Daten ohne besondere Aufforderung irgend eines Masters verschicken. Wie bei Ethernet kann es dazu kommen, dass mehrere Teilnehmer gleichzeitig senden. Die Nachricht mit dem niedrigsten Identifier setzt sich am Bus durch. Der Identifier mit der niedrigsten Binärzahl hat somit die höchste Priorität.

Den Vorgang zur Kollisionsprüfung über den Identifier nennt man „bitweise Arbitrierung“. Entsprechend dem "Wired-and-Mechanismus", bei dem der dominante Zustand (logisch 0) den rezessiven Zustand (logisch 1) überschreibt, verlieren all diejenigen Knoten den Wettstreit um die Buszuteilung, die rezessiv senden, aber auf dem Bus dominant beobachten. Alle "Verlierer" werden automatisch zu Empfängern der Nachricht mit der höchsten Priorität und versuchen erst dann wieder zu senden, wenn der Bus frei wird. Der CANbus ist somit ein Bussystem mit bedarfsabhängiger Buszuteilung.

Auch gleichzeitige Buszugriffe mehrerer Knoten müssen immer zu einer eindeutigen Busvergabe führen. Durch das Verfahren der bitweisen Arbitrierung über die Identifier der zur Übertragung anstehenden Botschaften wird jede Kollision nach einer berechenbaren Zeit eindeutig aufgelöst: Im CAN Standard Format sind es maximal 13 Bitzeiten, im erweiterten Format sind es maximal 33 Bitzeiten.

## Schichten der CAN-Software und CAN-Hardware

Die einzelnen Aufgaben des CANBus sind in sogenannten „Schichten“ (Layer) definiert.

1. Bitübertragungsschicht (Physical Layer). Diese Schicht beschreibt die physikalischen Eigenschaften, wie z.B. Stecker, Kabel, Signalpegel und wie ein Bit auf der Leitung dargestellt wird.
2. Übertragungsschicht (Transfer Layer)  
Die Übertragungsschicht hat die Aufgabe, das spezifizierte Busprotokoll abzuarbeiten. Dazu gehören die Erzeugung eines Übertragungsrahmens („Pakets“), die Anforderung des Busses mit der nötigen Erkennung des Buszustands (frei, belegt) und evtl. die Durch- bzw. Weiterführung des Zugriffs (dezentrale Buszuteilung). Dazu kommen die Aufgaben der Fehlererkennung und Fehleranzeige.
3. Objektschicht (Object Layer)  
Diese Schicht hat als Hauptaufgaben die Botschaftenverwaltung und Zustandsermittlung. Sie entscheidet, welche Botschaften momentan zu übertragen sind. Auf der Empfängerseite nimmt sie eine Botschaftenfilterung anhand der Kennung im Identifikationsfeld vor, d.h. eine Entscheidung, welche Botschaften vom Knoten

akzeptiert werden müssen und welche nicht.

#### 4. Anwendungsschicht (CAN Application Layer CAL)

In dieser Schicht werden die zu übertragenden Daten als Botschaften bereit gestellt und mit einer Kennung versehen, die eine inhaltsbezogene Adressierung ermöglichen. Durch die Wahl der Kennung wird jede Nachricht mit einer festgelegten Priorität versehen.

### Aufbau einer CAN Nachricht

Eine Nachricht wird in einer für den CAN-Bus eigenen Form verpackt. Diese Verpackung wird als „Frame“ bezeichnet.

Ein Frame besteht aus 7 Kennfeldern::

- Start-Condition
- Message Identifier
- Steuerbits
- Daten (0-8 Bytes)
- Prüfbits
- Acknowledge-Bit
- Stop-Condition

Man unterscheidet außerdem die Frames nach der Länge des Identifiers:

- Standard Frame (11 Bit Identifier)
- Extended Frame (29 Bit Identifier)

Nach der Art des Frames unterscheidet man den

- Data Frame (Daten werden ohne spezielle Aufforderung gesendet)
- Remote Data Frame (Daten werden angefordert – Ein Empfänger, der den REMOTE identifiziert, sendet daraufhin seine Nachricht)

Standard-Frame nach Standard CAN 2.0A:

Start	Identifier	RTR	IDE	r0	DLC	DATA	CRC	ACK	EOF+IFS
1 Bit	11 Bit	1 Bit	1 Bit	1 Bit	4 Bit	0...8 Byte	15 Bit	2 Bit	10 Bit

Extended-Frame nach Standard CAN2.0B:

Start	Identifier	SRR	IDE	Identifier	RTR	r1	r0	DLC	DATA	CRC	ACK	EOF+IFS
1 Bit	11 Bit	1 Bit	1 Bit	18 Bit	1 Bit	1 Bit	1 Bit	4 Bit	0...8 Byte	15 Bit	2 Bit	10 Bit

- **Start:** dominant, dient der Synchronisation,
- **Identifier:** Information für den Empfänger und Prioritätsinformation für die Busarbitrierung,
- **RTR:** rezessiv, unterscheidet zwischen Daten- (dominant) und Datenanforderungstelegramm (rezessiv),
- **IDE:** Identifier Extension,
- **r0:** reserviert,
- **DLC:** enthält die Längeninformation der nachfolgenden Daten,
- **DATA:** enthält die Daten des Telegramms,
- **CRC:** kennzeichnet den Fehlercode für die vorangegangenen Informationen. Die CRC Prüfsumme wird zur Fehlererkennung verwendet,
- **ACK:** enthält Rückmeldung von anderen Teilnehmern bei korrektem Empfang der Nachricht,

- **EOF**: kennzeichnet das Ende des Datentelegramms (7 rezessive Bits),
- **IFS**: kennzeichnet den Zeitraum für das Übertragen einer korrekt empfangenen Nachricht,
- **SRR**: ersetzt im Extended Frame das RTR Bit des Standard Frames,
- **IDE**: zeigt an, dass noch weitere 18 Bits folgen,
- **r1, r0**, reservierte Bits,
- **DLC**: Längeninformation der nachfolgenden Daten.

## Fehlererkennung im CAN Netzwerk

Das CAN-Protokoll kann Fehler selbst erkennen und signalisieren. Um Fehler zu erkennen, sind im CAN-Protokoll drei Mechanismen auf der Nachrichtenebene implementiert:

### 1. Cyclic Redundancy Check (CRC)

Der CRC sichert die Integrität des Rahmens, indem sendeseitig redundante Prüfbits hinzugefügt werden. Empfangsseitig werden diese Prüfbits aus den empfangenen Bits neu berechnet und mit den empfangenen Prüfbits verglichen. Bei Nichtübereinstimmung liegt ein CRC-Fehler vor.

### 2. Frame-check

Dieser Mechanismus überprüft die Struktur des übertragenen Rahmens. Die durch Frame-Check erkannten Fehler werden als Formatfehler bezeichnet.

### 3. ACK-Fehler

Von allen Empfängern werden die empfangenen Rahmen durch positives Acknowledgement quittiert. Wird am Sender kein Acknowledgement erkannt (ACK-Fehler), so deutet dies auf einen möglicherweise nur von den Empfängern erkannten Übertragungsfehler, auf eine Verfälschung des ACK-Feldes oder auf nicht vorhandene Empfänger hin.

Außerdem sind im CAN-Protokoll zwei Mechanismen zur Fehlererkennung auf der Bitebene implementiert.

### 1. Monitoring

Jeder Knoten der sendet, beobachtet gleichzeitig den Busspegel. Er erkennt dabei Differenzen zwischen gesendetem und empfangenen Bit. Dadurch können alle globalen Fehler und lokal am Sender auftretenden Bitfehler sicher erkannt werden.

### 2. Bit-stuffing

Auf der Bitebene wird die Codierung der Einzelbits überprüft. Das CAN-Protokoll nutzt die NRZ-Codierung (Non-Return-to Zero), die eine maximale Effizienz bei der Bitcodierung gewährleistet. Dabei werden die Synchronisationsflanken nach der Methode des Bit-stuffings erzeugt, indem vom Sender nach fünf aufeinanderfolgenden gleichwertigen Bits ein Stuff-Bit mit komplementärem Wert in den Bitstrom eingefügt wird, welches die Empfänger automatisch wieder entfernen. Werden ein oder mehrere Fehler mit Hilfe der oben beschriebenen Mechanismen von mindestens einem Knoten entdeckt, so wird die laufende Übertragung durch Senden eines "Error flag" abgebrochen. Dadurch wird die Annahme der übertragenen Nachricht durch andere Stationen verhindert und somit die netzweite Datenkonsistenz sichergestellt. Nach Abbruch der Übertragung einer fehlerhaften Botschaft beginnt der Sender automatisch, seine Nachricht erneut zu senden (Automatic Repeat Request).

Tritt ein Fehler mehrmals aufeinanderfolgend auf, führt dies zur automatischen Abschaltung des Knotens.

## Effektive Übertragungsrate für Datenbytes

Trotz des selbsttätigen Zugriffs eines CAN-Knotens auf die Busleitung können für einen Knoten der höchsten Priorität Anhaltswerte für die effektive Übertragungsrate angegeben werden. Eine Nachricht im Standard-Format mit acht Datenbytes benötigt maximal 130 Bits Dabei geht man von einer maximalen Anzahl von 19 Stuff-Bits und 3

Zwischenraumbits aus:

1 Start bit  
 +11 Identifier bits  
 + 1 RTR bit  
 + 6 Control bits  
 + 64 Data bits  
 + 15 CRC bits  
 + 19 (maximum) Stuff bits  
 + 1 CRC delimiter  
 + 1 ACK slot  
 + 1 ACK delimiter  
 + 7 EOF bits  
 + 3 IFS (Inter Frame Space) bits  
 = 130 bits

Im Extended-Format sind maximal 154 Bits zu übertragen.

Die effektive Übertragungsrate bei 1000 kBit/s ergibt sich aus der Anzahl der Datenbits dividiert durch die Gesamtanzahl der Bits mal 1000 kBit/s.

Die folgende Tabelle listet Anhaltswerte für die effektive Übertragungsrate bei unterschiedlicher Anzahl von Datenbytes auf. Die variable Anzahl der Stuff-Bits ist dabei nicht berücksichtigt:

Anzahl Datenbytes pro Frame	effektive Übertragungsrate in kBit/s bei 1000 kBit/s	
	mit Standard ID	mit Extended ID
0	0	0
1	145	107
2	254	193
3	338	264
4	405	323
5	460	374
6	505	417

	effektive Übertragungsrate in kBit/s bei 1000 kBit/s	
	7	432
8	576	489

Der Messverstärker GSV-3CAN, der pro Frame einen Messwert mit drei Datenbytes überträgt, kann über einen CANbus mit 1000 kBit/s maximal  $338 \cdot 1024 / 24 = 14421$  Messwerte pro Sekunde übertragen.

Im Packed-Format des GSV-3CAN wird dagegen ein Frame mit 8 Datenbytes genutzt, der vier Messwerte zu je zwei Bytes enthält. Dadurch erhöht sich die Anzahl auf  $576 \cdot 1024 / 64 \cdot 4 = 36864$  Messwerte pro Sekunde, die über den CANbus erfasst werden kann.

## Maximale Mess-Datenfrequenz mit dem Dehnmessstreifen Messverstärker GSV-3CAN

Zur Auswertung der Signale von Dehnmessstreifen (DMS) steht der Messverstärker GSV-3CAN zur Verfügung.

Die folgende Tabelle listet Anhaltswerte für die maximale Anzahl von Messdaten, die pro Sekunde im CAN Netzwerk übertragen werden können.

CAN Bus Übertragungsrate	Messfrequenz im Standard Format	Messfrequenz im Packed Format
125 kBit/s	1,5 kHz	3,7 kHz
250 kBit/s	3,0 kHz	7,5 kHz
500 kBit/s	6,0 kHz	15 kHz
1000 kBit/s	12,0 kHz	30 kHz