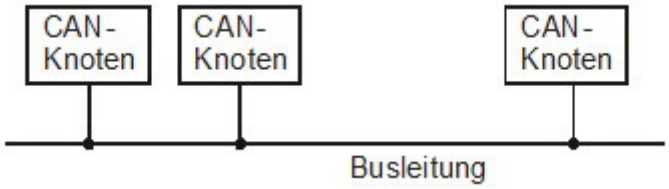
CAN-Bus

Allgemein

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. Das Controller Area Network (CAN) verbindet mehrere gleichberechtigte Komponenten (Knoten, Node) über einen 2-Draht Bus plus zusätzlicher Masseleitung miteinander. Das CAN-Protokoll wurde 1983 von Bosch für den Einsatz in Kraftfahrzeugen entwickelt und erstmals 1986 der Öffentlichkeit vorgestellt.

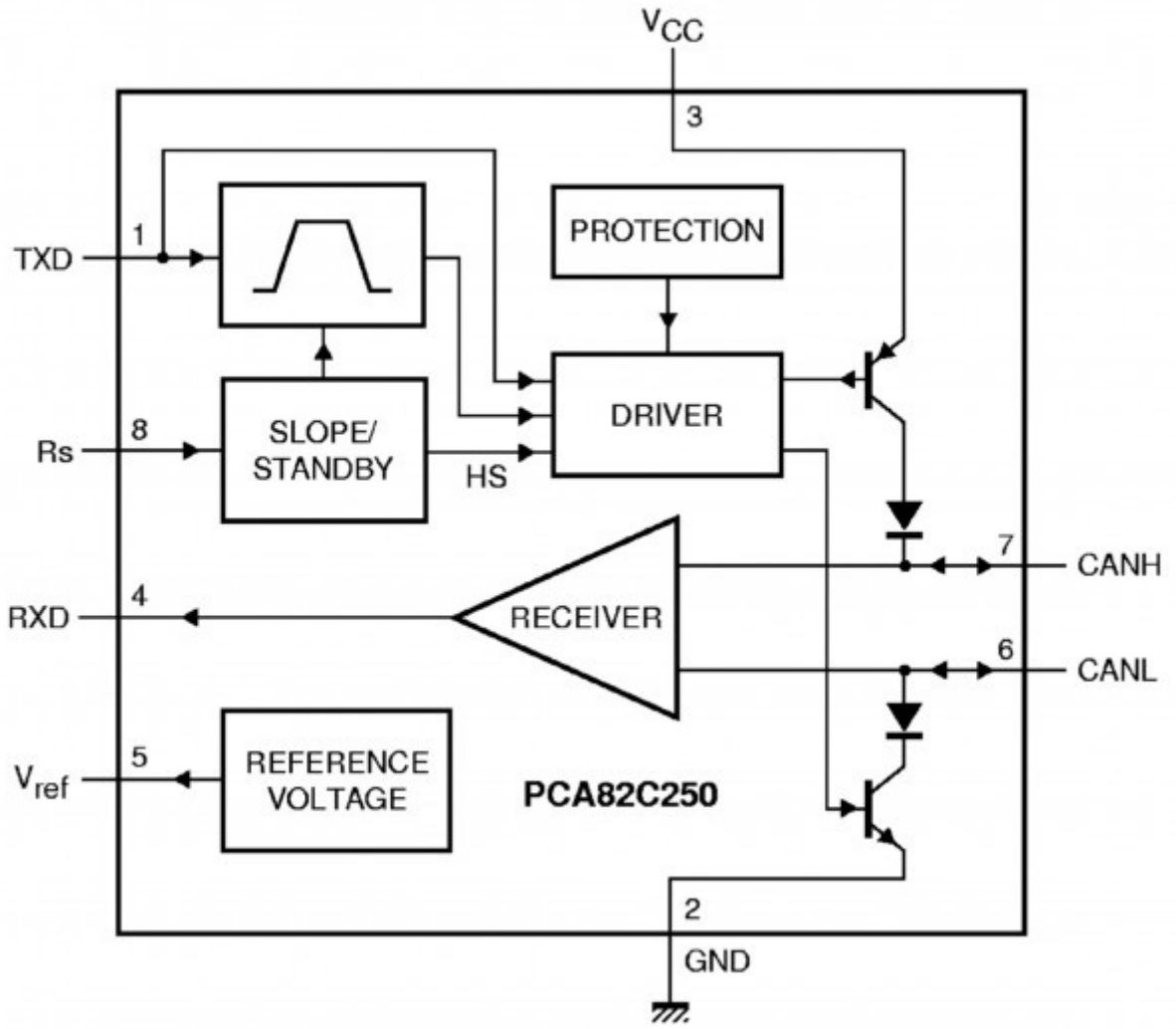


Aufgrund der hohen Störsicherheit, der geringen Kosten und der Echtzeitfähigkeit wird CAN nicht nur in der Automobilindustrie, sondern auch in vielen anderen Branchen (z. B. in Nutzfahrzeugen, mobilen Arbeitsmaschinen, Eisenbahnen, in der Medizintechnik, in der Industrieautomation, in Aufzügen, und als Maschinenbussystem) eingesetzt.
Die Organisation „CAN in Automation“ (CiA) widmet sich der Weiterentwicklung des CAN-Protokolls und der Spezifikation des CANopen-Anwendungsprotokolles sowie der CANopen-Profile.

Physikalische Beschreibung

Die physikalische CAN-Übertragung ist in ISO 11898-2 (high-speed) und ISO 11898-3 (low-speed) standardisiert. Zur Umsetzung dieser Spezifikation stehen Transceiver von verschiedenen Herstellern zur Verfügung, wie z. B. der PCA82C250 von NXP.
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 Leitungen 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: 2,0 Volt nominal); der Zustand mit zwei gleichen Pegeln wird als rezessiv bezeichnet (Pegeldifferenz: 0,0 Volt nominal).
Der dominante Zustand entspricht 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)	2V
1 oder hochohmig	rezessiv oder floating	Transistor gesperrt	Transistor gesperrt	0V

Eine weitere Maßnahme zur Erhöhung der Störsicherheit ist die NRZ-Kodierung, d. h., es gibt nicht in jedem Bit einen Flankenwechsel. Um zu vermeiden, dass die Teilnehmer die Synchronisation zum Sender verlieren, wird vom Sender nach fünf Bits gleicher „Polarität“ (rezessiv bzw. dominant) ein Bit der jeweils anderen Polarität eingefügt (englisch: „bit-stuffing). Die Empfänger entfernen diese Stuff-Bits automatisch, so dass die gesendete Bit-Sequenz und die zum Hostcontroller weitergeleitete identisch sind.

Steckerbelegung

Als CAN-Steckverbinder hat sich der von CAN in Automation (CiA) vorgeschlagene 9-polige Sub-D-Stecker in vielen Anwendungen durchgesetzt. In den Knoten werden sowohl „weibliche“ wie auch „männliche“ Steckverbinder gleichzeitig eingesetzt. Somit können ohne Unterbrechung weitere Knoten in die Busleitung integriert werden. („Daisy-chain“-Verdrahtung
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.
CiA hat auch für andere Steckverbinder die Pin-Belegung empfohlen (CiA 303-1): beispielsweise für den 5-poligen M12-Rundsteckverbinder und 4- sowie 5-polige „open style“-Stecker, ohne auf die Abmessungen einzugehen. Eine ebenfalls nützliche Sache ist der von CiA definierte 10-polige „multipole Connector“, bei dem Pin 10 reserviert ist und nicht genutzt werden darf.

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 1 Mbit/s übertragen. Alle CAN-Knoten müssen die Nachricht gleichzeitig verarbeiten können. Die maximale Kabellänge ist daher abhängig von der Bitrate. Die Tabelle zeigt empfohlene Bitraten und die entsprechende maximale Kabellänge.
Die erreichbare Länge ist daneben auch vom verwendeten Kabl, von der Netztopologie und vom Abtastzeitpunkt abhängig.

Bitrate	Kabellänge
10 kbits/s	6,7 km
20 kbits/s	3,3 km
50 kbits/s	1,0 km
125 kbits/s	500 m
250 kbits/s	250 m
500 kbits/s	125 m
1 Mbits/s	25 m

Busterminierung (Abschlusswiderstand)

Die Busterminierung erfolgt beim CAN-Bus bei einer Linientopologie mit 120 Ohm an beiden Enden des Netzwerkes.

Eine Terminierung ist auch schon bei kurzen Leitungen mit niedrigen Bitraten empfehlenswert. 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 (ID) 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.
Alle Nachrichten sind von allen Teilnehmer gehört (englisch: „broadcast“) und abhängig von der ID-Akzeptanzfilterung zur Verarbeitung an den Hostcontroller weiter geleitet. Messgrößen, die von mehreren Steuergeräten als Information benötigt werden, können über das CAN-Netz so verteilt werden, so dass nicht jedes Steuergerät einen eigenen Sensor benötigt.

Kollisionsprüfung

Jeder Teilnehmer darf Nachrichten ohne besondere Aufforderung eines anderen Teilnehmers (z. B. Master) verschicken. Wie bei Ethernet kann es dazu kommen, dass mehrere Teilnehmer gleichzeitig senden. Die Nachricht mit dem niedrigsten Identifier (dem niedrigsten Binärwert) erhält die Sendeerlaubnis.
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 eine dominantes Bit detektieren. Alle „Verlierer“ werden automatisch zu Empfängern der Nachricht mit der höchsten Priorität und versuchen erst dann wieder zu senden, wenn der Bus frei wird.
Gleichzeitige Buszugriffe mehrerer Knoten müssen immer zu einer eindeutigen Sendeerlaubnis führen, deshalb müssen die Identifier eindeutig vergeben werden, d. h. sie dürfen nicht von zwei Teilnehmern gleichzeitig genutzt werden. Durch das Verfahren der bitweisen Arbitrierung über die Identifier der zur Übertragung anstehenden Botschaften wird jede Kollision nach einer berechenbaren Zeit eindeutig aufgelöst: Bei Nachrichten im Basis-Format (11-bit-ID) sind es maximal 13 Bitzeiten (29-bit-ID), im erweiterten Format sind es maximal 33 Bitzeiten. Dabei sind die oben erwähnten Stuffbits nicht berücksichtigt.

Schichten der CAN-Software und CAN-Hardware

Die einzelnen Aufgaben der CAN-Kommunikation erfolgt entsprechend dem IOS/OSI-Referenzmodell in „Schichten“ (Layer).

- Bitübertragungsschicht (Physical Layer): Diese Schicht beschreibt die physikalischen Eigenschaften, wie z. B. Signalpegel, Übertragungsgeschwindigkeit, Abtastzeitpunkt, Stecker, Kabel, usw. Sie ist partiell im CAN-Controller und im CAN-Transceiver realisiert.

- Übertragungsschicht (Data Link Layer): Dies ist das eigentliche CAN-Protokoll mit seinem Nachrichtenformaten (Datentelegramme, Remote-Request-Telegramm, Fehlertelegramm und Überlasttelegramm) sowie dem Fehlerverhalten (englisch: „fault confinement“).
- Die höhere Protokolle: Die darüber liegenden Schichten sind in der Regel nicht einzeln ausgewiesen und werden normalerweise in Software auf dem Hostcontroller implementiert. In einigen Branchen sind diese höheren Protokolle standardisiert (z. B. CANopen, DeviceNet, SAE J1939). Die Automobilindustrie hat ein Transportprotokoll in ISO 15675 international genormt, mit dem man lange Nachrichten mit mehr als 8 byte auf der Senderseite segmentieren und auf der Empfängerseite wieder zusammenbauen kann.

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 sieben Kennfeldern:

- Startfeld (Start-of-frame bit)
- Arbitrationsfeld (CAN-Identifier plus RTR-Bit)
- Steuerfeld (enthält den Datenlängencode)
- Datenfeld (0 bis 8 Byte)
- CRC-Feld (enthält eine 15-bit-Prüfsumme sowie eine Endemarkierung)
- Acknowledge-Feld (ACK-Bit plus Endemarkierung)
- Endefeld (End-of-frame)

Man unterscheidet zwei Frame-Formate, die sich vor allem durch die Länge des Identifiers unterscheiden:

- Basis-Format (11-bit-Identifier)
- Extendet-Format (29-bit-Identifier)

Bei den Frames unterscheidet man folgende Arten:

- Data Frame (Nachricht wird ohne spezielle Aufforderung gesendet)
- Remote Transmission Request (RTR) Frame (Nachricht wird angefordert – der Empfänger, der die Nachricht mit dem angeforderten Identifier „besitzt“, liefert das korrespondierende Data Frame)

Basis-Frame nach ISO 11898-1 (früher auch als CAN 2.0A bezeichnet):

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	16 Bit	2 Bit	10 Bit

Extended-Frame nach ISO 11898-1 (früher auch als CAN2.0B bezeichnet):

Start	Identifier	SRR	IDE	r1	r0	DLC	DATA	CRC	ACK	EOF+IFS
1 Bit	11 Bit	1 Bit	1 Bit	1 Bit	1 Bit	4 Bit	0-8 Byte	16 Bit	2 Bit	10 Bit

- **Start:** dominant, dient der Synchronisation
- **Identifier:** Information für den Empfänger und Prioritätsinformation für die Busarbitrierung
- **RTR:** unterscheidet zwischen Daten- (dominant) und RTR-Telegramm (rezessiv)
- **IDE:** Identifier Extension

- **r0, r1:** reserviert
- **DLC:** enthält die Längeninformation des nachfolgenden Datenfeldes
- **DATA:** enthält die Daten des Telegramms
- **CRC:** enthält die CRC-Prüfsumme und die CRC-Endeerkennung für die vorangegangenen Bitsequenz
- **ACK:** enthält eine Bestätigung von anderen Teilnehmern bei korrektem Empfang der Nachricht
- **EOF:** kennzeichnet das Ende des Datentelegramms (sieben rezessive Bits)
- **IFS:** 3-bit-langer Zwischenraum zwischen CAN-Frames
- **SRR:** ersetzt im Extended-Frame das RTR-Bit
- **IDE:** zeigt an, dass noch weitere 18 Bits folgen

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 Information 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 (das recessive Bit des Senders wird durch dominante Bits der Empfänger „überschrieben“). 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.

From:
https://wiki.gps-watch.de/ - **Wiki**

Permanent link:
https://wiki.gps-watch.de/informationen/can-bus

Last update: **2019/07/24 16:48**