

Datenkommunikation im Automobil

Teil 2: Sicherer Datenaustausch mit CAN

Die immer komplexer werdenden elektronischen Systemen sorgen für ein höheres Maß an Sicherheit und Komfort beim Autofahren. Durch seine spezifischen Merkmale – so wird etwa der sichere Datenaustausch auch unter widrigen Umgebungsbedingungen sichergestellt – leistet dabei das serielle Bussystem CAN (Controller Area Network) einen entscheidenden Beitrag.

Von Eugen Mayer

Die von Bosch [1] entwickelte CAN-Technik ist seit 1993 genormt und liegt, gegliedert in mehrere Teile, als ISO-Norm 11898 vor (Bild 1). Der erste Teil umfasst das CAN-Protokoll und deckt vollständig den Data Link Layer (Framing, Adressierung, Buszugriff, Datensicherung) und teilweise den Physical Layer (Physical Signaling) des standardisierten Referenzmodells der Datenkommunikation (ISO 7498) ab. Das CAN-Protokoll wird in Hardware implementiert, wofür mittlerweile eine Vielzahl von kostengünstigen CAN-Controllern zur Verfügung steht.

Der zweite Teil beschreibt den CAN-High-Speed-Physical-Layer, der dritte Teil den CAN-Low-Speed-Physical-Layer. Beide decken den Physi-

cal Layer der ISO 7498 ab (unter anderem die physikalische Busankopplung, Datenraten und die Spannungspegel). Der CAN-High-Speed-Physical-Layer kommt vor allem in Antriebs- und Fahrwerksapplikationen zum Einsatz. Man realisiert ihn im Wesentlichen durch den CAN-High-Speed-Transceiver, der eine maximale Datenrate von 1 Mbit/s unterstützt. Für den hauptsächlich im Komfortbereich eingesetzten CAN-Low-Speed-Physical-Layer nutzt man in der Regel den CAN-Low-Speed-Transceiver mit einer maximalen Datenrate von 125 kbit/s.

Die CAN-Schnittstelle (Bild 2) besteht demnach aus einem CAN-Controller und einem CAN-Transceiver. Während der CAN-Controller das

CAN-Protokoll abwickelt, übernimmt der CAN-Transceiver die Aufgabe, den CAN-Controller physikalisch an den im Differenzsignalmodus betriebenen CAN-Bus zu koppeln. Die Differenzsignalübertragung verbessert die Störfähigkeit und erfordert zwei Kommunikationsleitungen (CAN-High- und CAN-Low-Leitung), die zur Vermeidung von Reflexionen an den Enden mit dem Wellenwiderstand abgeschlossen werden.

Zuordnung von Knoten und Nachrichten über Nachrichtenadressen und -filter

Die Nachrichtenadressen, üblicherweise als Identifier (ID) bezeichnet, bestimmen nicht die CAN-Zielknoten, sondern die Identität der Nachrichten selbst. Prinzipiell stehen so alle CAN-Nachrichten allen CAN-Knoten zum Empfang zur Verfügung (Nachrichtenverteilung). Über einen Filter selektiert jeder CAN-Knoten die für ihn relevanten CAN-Nachrichten aus dem Nachrichtenstrom (empfängerselektives System). Aufgrund des 11-bit-breiten Identifiers lassen sich in einem CAN-Netzwerk bis zu 2048 CAN-Nachrichten spezifizieren.

Diese Form der Nachrichtenverteilung bietet folgende Vorteile:

- ▶ Kosteneinsparung durch Mehrfachausnutzung von Sensoren.
- ▶ Einfache Realisierung und Synchronisation von verteilten Prozessen.
- ▶ Hohe Flexibilität hinsichtlich der Konfiguration.

Der Verzicht auf Knotenadressen erlaubt die Integration von weiteren Busknoten, ohne dass die Hard- oder Software vorhandener Busknoten modifiziert werden muss. Dies gilt allerdings nur, wenn es sich beim hinzukommenden Busknoten ausschließlich um einen Empfänger handelt.

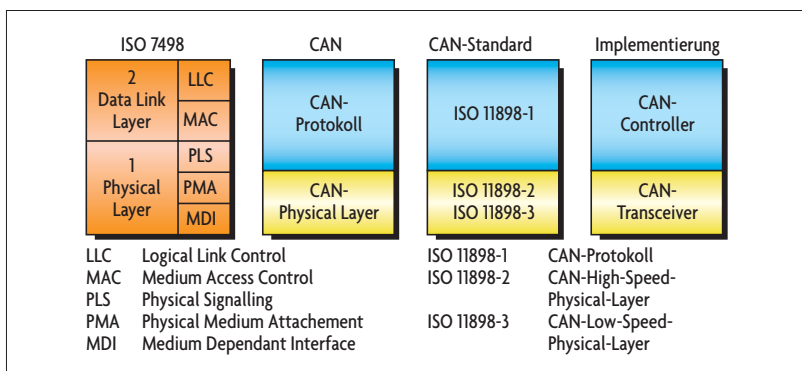


Bild 1. Die CAN-Technik ist seit 1993 genormt und liegt als ISO-Norm 11898 vor, die den Data Link Layer und teilweise den Physical Layer des standardisierten Referenzmodells der Datenkommunikation (ISO 7498) abdeckt.

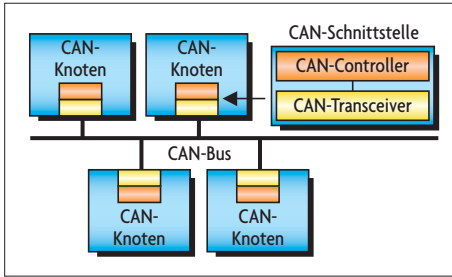


Bild 2. Die CAN-Schnittstelle besteht aus dem eigentlichen Controller zur Abwicklung des CAN-Protokolls und dem Transceiver zur physikalischen Ankopplung an das differenzielle CAN-Netzwerk.

Datenübertragung erfolgt ereignisgesteuert

Die übertragenen Nachrichten und deren Reihenfolge sind in einem CAN-Netzwerk nicht vom Fortschreiten der Zeit abhängig, sondern vom Auftreten spezieller Ereignisse. Jeder CAN-Knoten ist prinzipiell berechtigt, sofort nach Auftreten eines Ereignisses auf den CAN-Bus zuzugreifen. In Verbindung mit der vergleichsweise kurzen Nachrichtenlänge von maximal 130 bit im Standard-Format und der hohen Datenübertragungsrates bis zu 1 Mbit/s ermöglicht das Verfahren schnelle Reaktionen auf asynchrone Vorgänge. Dies ist eine wichtige Voraussetzung für eine echtzeitfähige Datenübertragung im Millisekundenbereich (1 bis 10 ms), die vor allem die Applikationen des Antriebs und des Fahrwerks verlangen.

Da der CAN-Kommunikation kein Zeitplan zugrunde liegt, ergibt sich der Nachrichtenverkehr stets erst zur Laufzeit und birgt deshalb implizit die Gefahr von Kollisionen. Diese steigt mit zunehmender Buslast und stellt die Echtzeit-Fähigkeit in Frage. Um trotz zufälligen Buszugriffs Echtzeit-Datenübertragung zu gewährleisten, kommt im CAN-Netzwerk das CSMA/CA-Buszugriffsverfahren (Carrier Sense Multiple Access/Collision Avoidance) zum Einsatz.

CSMA/CA-Zugriffsverfahren sorgt für zerstörungsfreien, prioritätsgesteuerten Buszugriff

Der Buszugriff beginnt damit, dass ein sendewilliger CAN-Knoten zunächst den CAN-Bus abhört (Carrier Sense).

Ist der CAN-Bus frei, darf der CAN-Knoten sofort mit der Nachrichtenübertragung beginnen. Stellt er hingegen Busaktivitäten fest, muss er seinen Sendewunsch so lange zurückstellen, bis der CAN-Bus frei und die gerade laufende Nachrichtenübertragung abgeschlossen ist. Zudem hat er eine Pause von drei Bitzeiten (ITM – Intermission) abzuwarten. Eine laufende Nachrichtenübertragung wird nicht unterbrochen, der Buszugriff ist damit zerstörungsfrei. Im Falle mehrerer sendewilliger CAN-Knoten verhindert die bitweise Arbitrierung (Bild 3) trotz simultanen Bus-

CAN-Knoten weiter senden darf oder das Senden einstellen muss.

Am Ende der Arbitrierungsphase bekommt jener CAN-Knoten die Sendeberechtigung, der die CAN-Nachricht mit dem niederwertigsten Identifier überträgt. Unterlegene CAN-Knoten wechseln zunächst in den Empfangszustand und greifen für einen erneuten Sendeversuch auf den CAN-Bus zu, sobald dieser wieder frei ist. So verhindert die Bus- und Arbitrierungslogik nicht nur Kollisionen (Collision Avoidance), sondern sorgt auch für einen prioritätengesteuerten Buszugriff: Je niederwertiger ein Identifier ist, desto höher ist die Priorität der CAN-Nachricht und damit ein

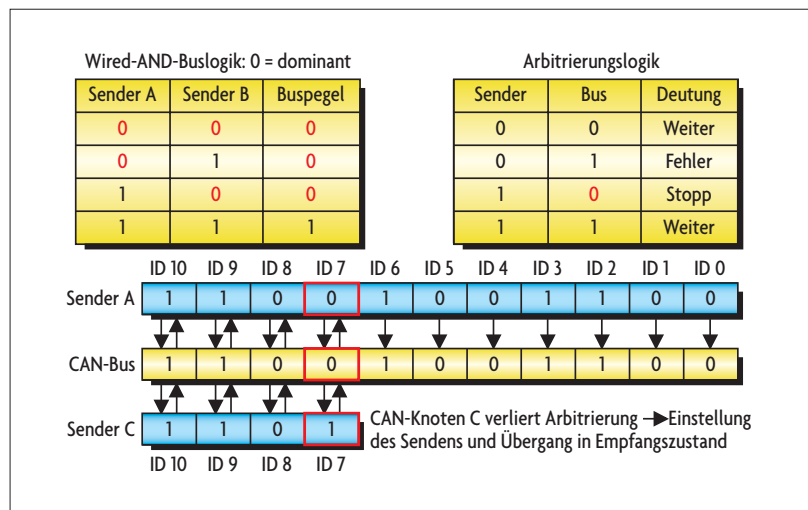


Bild 3. Die Wired-AND-Buslogik sorgt, hier am Beispiel von zwei CAN-Knoten, für einen eindeutigen Buspegel. Die Arbitrierungslogik entscheidet, ob Busteilnehmer senden dürfen.

zugriffs (Multiple Access – MA) das Auftreten von Kollisionen. Im Rahmen der bitweisen Arbitrierung legen alle sendewilligen CAN-Knoten den ID der zu übertragenden CAN-Nachrichten bitweise vom höchst- zum niederwertigsten Bit an. Die dem CAN-Netzwerk zugrunde liegende Wired-AND-Buslogik (0 = dominant) sorgt dafür, dass sich immer ein eindeutiger Buspegel einstellt. Nach dem Aufschalten eines ID-Bits vergleicht jeder CAN-Knoten den Buspegel mit dem angeschalteten Pegel. Die Arbitrierungslogik entscheidet, ob ein

schnellerer Buszugriff möglich. Die CAN-Nachricht mit dem kleinsten Identifier (ID = 0) wird deshalb ohne Verzögerung übertragen.

Ist die Buslast nicht zu hoch, sorgt diese Art des zufälligen, zerstörungsfreien und prioritätengesteuerten Buszugriffs für einen gerechten und sehr schnellen Buszugriff. Allerdings muss einerseits berücksichtigt werden, dass mit zunehmender Buslast die Verzögerungen vor allem von niederpriorigen CAN-Nachrichten anwachsen. Das kann so weit führen, dass CAN-Nachrichten im schlimmsten Fall zu spät bei Empfängern ankommen oder völlig unterdrückt werden. Andererseits verursacht das CSMA/CA-Buszugriffsverfahren einen reziproken Zusammenhang zwischen Netzwerkausdehnung und maximaler Datenrate. Während der bitweisen Arbitrierung

ist ein Gleichlauf zwischen Sender und Empfänger. Da aus Kosten- und Aufwandsgründen auf eine Taktleitung verzichtet wird, realisiert man den Gleichlauf mittels Signalfanken und einem definierten Re-Synchronisationsmechanismus. Jede Nachrichtenübertragung beginnt mit dem Übertragen des dominanten Synchronisationsbits (SOF, Start of Frame) und erzeugt so die erste Signalfanke (Bus-Idle entspricht einem rezessiven Buspegel). Der Empfänger stellt während der gesamten Übertragung durch Auswertung jeder ankommenden Signalfanke die Synchronisation sicher und passt notfalls sein eigenes Bit-Timing entsprechend an. Das Bit-Stuffing-Verfahren sorgt dafür, dass spätestens nach fünf homogenen Bits ein komplementäres Bit und somit eine Signalfanke erscheint.

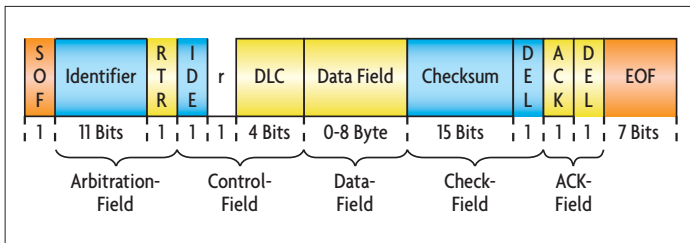


Bild 4. Frames enthalten einen 11-bit- (Standard) oder 29-bit-Identifier (Extended). Bei Remote-Frames ist, im Gegensatz zu Data-Frames, die Länge des Data-Field gleich Null.

muss ein rezessiv sendender CAN-Knoten einen dominanten Pegel sicher erkennen. Die Größe des Bit-Zeitintervalls ist deswegen so zu wählen, dass die Signallaufzeiten auf dem CAN-Bus vollständig kompensiert werden. Steigende Netzausdehnung bedingt somit ein verlängertes Bit-Zeitintervall, woraus schließlich eine maximal nutzbare Datenrate resultiert.

■ Datenübertragung erfolgt mittels Data-Frames

Neben den für die Datenübertragung hauptsächlich eingesetzten Data-Frames (Bild 4) gibt es auch Remote-Frames zur Anforderung von Daten. Sie kommen aber kaum zur Anwendung, da die Datenübertragung im Automobil nicht auf Nachfrage, sondern im Wesentlichen auf Informationserzeugerinitiative basiert. Beide Frametypen sind identisch aufgebaut. Allerdings entfällt beim Remote-Frame das Data-Field.

Grundvoraussetzung für die Übertragung von Data- und Remote-Frames

ist ein Anschluss an das SOF folgt der ID, der entweder 11 bit (Standard-ID) oder 29 bit (Extended-ID) lang sein kann. Im Automobil dominiert das Standardformat. Das Extended-Format spielt typischerweise im Zusammenhang mit höheren Protokollen wie SAE J1939 eine Rolle. Mittels IDE-Bit (Identifier Extension) wird das ID-Format angezeigt. Ein anderer Bitschalter (RTR-Bit, Remote Transmission Request) zeigt an, ob es sich um ein Data- oder ein Remote-Frame handelt.

Zur Übertragung von Nutzinformationen steht das 64 bit breite Data-Field zur Verfügung, bei dem die genaue Anzahl der Nutzbytes mittels DLC (Data Length Code) angegeben wird. Dem Data-Field folgt die CRC-Sequence (Cyclic Redundancy Check). Anhand aller zu übertragenden Bits, eines Generatorpolynoms und eines definierten Algorithmus generiert der Sender die CRC-Sequence. Unabhängig von der Nachrichtenfilterung geschieht dasselbe empfängerseitig mit den ankommenden Bits. Es folgen der Vergleich der beiden Sequenzen und die Quittierung nach dem rezessiven CRC-Delimiter im Acknowledge-Slot (ACK-Slot). Am Ende eines Data-Frames steht nach dem rezessiven

ACK-Delimiter das 7 bit lange und rezessive EOF (End of Frame).

■ Fehlererkennungsmechanismen sorgen für hohe Datensicherheit

Die Wahrscheinlichkeit, dass verfälschte CAN-Nachrichten unerkannt bleiben, ist außerordentlich gering. Sie liegt bei $4,7 \times 10^{-11}$ [2]. Dafür verantwortlich sind die im CAN-Protokoll definierten Fehlererkennungsmechanismen. Dazu gehören auf der Empfängerseite neben dem von der Nachrichtenfilterung unabhängigen CRC, mit dem sich bis zu fünf Fehler innerhalb einer CAN-Nachricht detektieren lassen, die Überprüfung des Formats (Form Check) und die Bit-Stuffing-Regel (Stuff Check). Der Sender führt ein Bit-Monitoring durch und wertet den ACK-Slot aus.

Legt man einem CAN-Netzwerk eine Fehlerrate von 10^{-3} zugrunde, dann treten bei einer jährlichen Betriebsdauer von 1000 Stunden, einer Datenrate von 500 kbit/s, einer mittleren Buslast von 25 % und einer mittleren Nachrichtenlänge von 80 bit statistisch knapp alle 4000 Jahre vom CAN-Protokoll unerkannte verfälschte CAN-Nachrichten auf. Unter der Fehlerrate versteht man das Verhältnis gestörter CAN-Nachrichten zur Anzahl aller übertragenen CAN-Nachrichten.

Sobald ein Fehlererkennungsmechanismus einen Übertragungsfehler signalisiert, bricht der CAN-Knoten, der den Fehler erkannt hat, die Nachrichtenübertragung ab, indem er ein Error-Flag (sechs dominante Bits) auf den CAN-Bus auflegt. Das Error-Flag verletzt bewusst die Bit-Stuffing-Regel, so dass netzwerkweit jeder CAN-Knoten den bis dahin lokalen Fehler wahrnimmt und infolgedessen die Nachrichtenübertragung mit dem Aufschalten eines Error-Flags abbricht. Diese Methode stellt die für die Datenintegrität verteilter Anwendungen so wichtige netzweite Datenkonsistenz sicher.

Die Fehlerkorrektur besteht darin, die abgebrochene CAN-Nachricht durch denselben Sender zu wiederholen, sobald der CAN-Bus wieder frei ist (nach dem Error-Delimiter und ITM). Bei der Auslegung des Systems

muss man berücksichtigen, dass es aufgrund des CSMA/CA-Buszugriffverfahrens keine Garantie für das sofortige Wiederholen gibt. Die Fehlerholzeit hängt von der Nachrichterpriorität und der Buslast ab.

Die Fehlersignalisierung via Error-Flag versetzt jeden CAN-Knoten in die Lage, laufende Nachrichtenübertragungen abzubrechen. Da dies auch für defekte CAN-Knoten gilt, sind solche in der Lage, die gesamte CAN-Kommunikation zum Erliegen zu bringen.

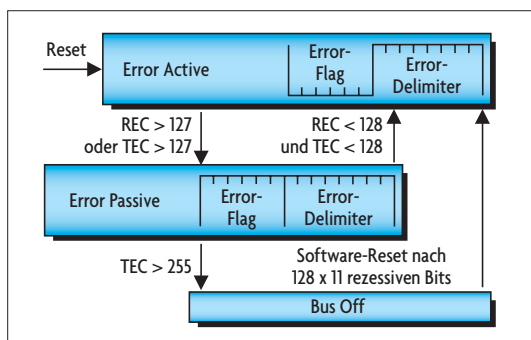


Bild 5. Jeder CAN-Knoten verfügt über eine Netzknottenüberwachung, die mittels Fehlerzähler defekte Knoten erkennen und abschalten kann.

gen. Um dies zu verhindern, verfügt jeder CAN-Knoten über eine Netzknottenüberwachung (Bild 5), die mittels Fehlerzähler und Regeln zur Steuerung der Fehlerzähler den defekten Knoten abschalten kann (Bus Off).

Quittierung empfangener CAN-Nachrichten

In einem CAN-Netzwerk wird unabhängig von der Nachrichtenfilterung jede Nachrichtenübertragung von allen Empfängern gleichzeitig im ACK-Slot quittiert (In-Frame-Acknowledgement). Ein dominanter Pegel entspricht einer positiven, ein rezessiver Pegel einer negativen Quittierung. Da der Sender den ACK-Slot rezessiv belegt, reicht eine positive Quittierung aus, um die Korrektheit der Nachrichtenübertragung zu bestätigen. Aufgrund dieses knottenneutralen positiven Acknowledgement werden negativ quittierende CAN-Knoten überschrieben und bleiben ungehört. Deshalb senden diese nach dem ACK-Delimiter ein Error-Flag.

Liegt keine einzige positive Quittierung vor, wird also der ACK-Slot

von keinem Empfänger überschrieben, detektiert der Sender einen ACK-Fehler und bricht die laufende Nachrichtenübertragung mit dem Aufschalten eines Error-Flags ab.

Forderung nach Echtzeitfähigkeit und hoher Datenübertragungsrate überlastet CAN

Noch bis vor einigen Jahren war CAN die am meisten gefragte Bustechnik im

Automobil. Mit voranschreitender Elektrifizierung stößt CAN zunehmend an seine Grenzen. Speziell bei echtzeitkritischen und höchst sicherheitskritischen Kfz-Anwendungen, die zudem eine hohe Datenübertragungsrate erfordern wie beispielsweise das Fahrerassistenzsystem „Lane Keeping Assistance“, kommen zunehmend neue Techniken

zum Einsatz, aber auch in kostensensitiven Komfortanwendungen stellen die Fahrzeugentwickler den CAN-Bus in Frage.

Deswegen haben sich im Laufe der Zeit neben CAN zwei andere Busse für den Einsatz im Automobil etabliert oder sind auf dem besten Wege dorthin: LIN und FlexRay. LIN (Local Interconnected Network) wird bereits zur kostengünstigen Vernetzung von Sensoren und Aktoren im Komfortbereich eingesetzt. FlexRay ist aufgrund der zeitgesteuerten Kommunikationsmethode, einer Datenrate von bis zu 20 Mbit/s und der Möglichkeit, auf zwei Kommunikationskanälen zu senden, auf dem Vormarsch. Zum weltweit ersten Serieneinsatz kommt FlexRay im neuen BMW X5 in einem aktiven Dämpferkontrollsystem.

Zuverlässige Steuergerätevernetzung

Die Spezialisten der Vector Informatik [3] unterstützen Fahrzeughersteller und -zulieferer nicht nur bei der CAN-Vernetzung, sondern auch bei den Bussystemen LIN, FlexRay und MOST.

Für Kundenprojekte sind durchgängige Werkzeugketten aus Design- und Entwicklungstools wahlweise mit Softwarekomponenten und Basissoftware für AUTOSAR-Steuergeräte verfügbar. Für den Einstieg in die Steuergerätevernetzung bzw. den Datenaustausch im Automobil bietet das Stuttgarter Unternehmen das eintägige Seminar „Serielle Bussysteme im Automobil“ an. Grundlagenseminare zu CAN, LIN, FlexRay und MOST vermitteln das notwendige Basiswissen, um sich schnell mit den vielfältigen Entwicklungstätigkeiten rund um die Automobilelektronik vertraut zu machen [4].

Der erste Teil dieser Beitragsreihe [5] befasste sich allgemein mit seriellen Bussystemen im Automobil. In den Folgen drei bis fünf werden die seriellen Bussysteme LIN, FlexRay und MOST behandelt. Der interessierte Leser findet zu den bereits veröffentlichten Themen auf der Internetseite der VectorAcademy [4] ergänzende und vertiefende Informationen. sj

Literatur und Links

- [1] www.can.bosch.com
- [2] Unruh, J.; Mathony, H. J.; Kaiser, K.H.: Error Detection Analysis of Automotive Communication Protocols. SAE International Congress 1990.
- [3] www.vector-informatik.de
- [4] www.vector-academy.de
- [5] Mayer, E. Datenkommunikation im Automobil. Teil 1: Architektur, Aufgaben und Vorteile serieller Bussysteme. *Elektronik Automotive* 2006, H. 7, S. 70 bis 73.



Dipl.-Ing., Dipl.-Techpaed. Eugen Mayer

hat nach der Berufsausbildung zum Kommunikationselektroniker an der FH Ravensburg/Weingarten Elektronik und an der Universität Stuttgart Elektrotechnik und Berufspädagogik studiert. Er arbeitet seit 1999 bei der Vector Informatik und ist dort als Senior Trainer tätig. eugen.mayer@vector-informatik.de