

Bussysteme im Automobil CAN, FlexRay und MOST

Thomas Dohmke
thomas@dohmke.de

Technische Universität Berlin
Fakultät Elektrotechnik und Informatik
Fachgebiet Softwaretechnik

In Zusammenarbeit mit der DaimlerChrysler AG.

Berlin, im März 2002

Zusammenfassung

Die Entwicklung verteilter eingebetteter Systeme war das Thema eines Seminars, das am 25. Februar 2002 vom Fachgebiet Softwaretechnik der TU-Berlin in Zusammenarbeit mit der DaimlerChrysler AG veranstaltet wurde. Neben dem Entwurf und der Analyse von verteilten eingebetteten Systemen war das Anwendungsgebiet Automotive Schwerpunkt des Seminars.

Der Anteil der Elektronik im Automobil hat in den letzten Jahren stetig zugenommen. Damit verbunden ist eine immer komplexere Vernetzung von Sensoren, Aktoren oder Steuergeräten sowie von Unterhaltungs- und Navigationssystemen. Dabei kommen häufig verschiedene Bussysteme, die sich hinsichtlich Anforderungen, Leistungsfähigkeit, Kosten und Echtzeitfähigkeit unterscheiden, zum Einsatz. Mit X-By-Wire- und Telematik-Systemen wird zudem eine neue Generation an Netzwerken notwendig werden.

Die vorliegende Arbeit gibt einen Überblick über die verbreiteten und zukünftigen Bussysteme im Automobil, wobei insbesondere die Netzwerke CAN, FlexRay und MOST im Vordergrund stehen.

Inhaltsverzeichnis

1	Einleitung	2
2	CAN und TTCAN	3
2.1	Merkmale	3
2.2	Kommunikation	4
2.3	Fehlererkennung	5
2.4	TTCAN	6
3	FlexRay	7
3.1	Merkmale	7
3.2	Kommunikation	8
4	MOST	11
4.1	Merkmale	11
4.2	Architektur	12
4.3	Kommunikation	13
4.4	Fehlererkennung	14
5	Weitere Bussysteme	15
5.1	LIN	15
5.2	TTP	15
5.3	Firewire / USB	15
5.4	Power Line Communication (DC-Bus)	15
6	Vergleich	17
7	Vernetzung von Netzen	18
7.1	Supergateway	18
7.2	Backbone-Architektur	18
8	Software	19
8.1	Entwurf mit xDesigner	19
8.2	Analyse mit CANalyzer	20

1 Einleitung

Heutige Automobile vom Kleinwagen bis zur Oberklasse enthalten eine Vielzahl von elektronischen Geräten. Die Systeme lassen sich dabei grob in zwei Klassen einteilen:

- Steuergeräte, z.B. ABS, ESP oder Motor-/Getriebesteuerungen
- Multimediakomponenten, z.B. Navigationssystem oder CD-Player

Es ist absehbar, dass die Menge der Elektronik in den kommenden Jahren deutlich zunehmen wird. Sowohl der Gesetzgeber, der an einer Verbesserung des Abgasverhaltens und des Treibstoffverbrauches interessiert ist, als auch der Kunde, der immer neue Verbesserungen in Fahrkomfort und Fahrsicherheit wünscht, fördern diese Entwicklung. Schließlich verbessern auch die Autohersteller kontinuierlich ihre Modelle, sei es um Kosten zu sparen oder als innovatives Element, das Systeme hervorbringt, die sich ein Kunde heute noch gar nicht vorstellen kann.

Eine der zukünftigen Entwicklungen sind sogenannte X-By-Wire-Systeme. Unter X-By-Wire werden Systeme zusammengefasst, die herkömmliche mechanische Systeme ersetzen sollen. Schon im Einsatz ist beispielsweise Brake-By-Wire als Ersatz des hydraulischen Brems-Kreislaufes durch eine elektronische Bremse, bei dem Bremsimpulse über Draht (Wire) ausgelöst werden.¹ Noch Zukunftsmusik ist hingegen die elektronische Lenkung Steer-By-Wire.

Neben den X-By-Wire-Systemen gewinnt die Telematik zunehmend an Bedeutung. Unter Telematik versteht man die Verbindung von Telekommunikation und Informatik. Im Kraftfahrzeug wird diese beispielsweise als Kombination von Navigationssystem und Verkehrsfunk zur Verkehrslenkung eingesetzt werden.

Darüberhinaus werden Systeme, die sich die Vorteile von X-By-Wire und Telematik zunutze machen, wie z.B. das virtuelle Verbinden von LKWs, das automatische Einparken oder gar das automatische Fahren verfügbar sein. Mit der Komplexität der dabei realisierten Funktionen steigt auch der Aufwand der Vernetzung. Statt einzelne Geräte über dedizierte Signalleitungen zu verbinden, gewinnen Bussysteme daher immer mehr an Bedeutung. Bussysteme haben hierbei folgende Vorteile:

- Jedes Gerät muss nur einmal an den Bus angeschlossen werden. Dadurch sinken Gewicht, Raumbedarf und Preis der Verkabelung.
- Ein an den Bus angeschlossenes Gerät kann mit allen anderen Geräten am Bus kommunizieren.
- Es sind Diagnose-Komponenten, die den Bus überwachen, durch Broad/Multicasts möglich.
- Über redundante Leitungen kann die Ausfallsicherheit eines Systems erhöht werden.
- Standards für den Bus, seine Schnittstellen und die Kommunikation fördern die Modularität der Geräte, so dass einerseits Geräte mit gleichartigen Funktionalitäten wiederverwendet werden können und andererseits der Variantenvielfalt vieler Automobile Rechnung getragen wird.

Bei der Verwendung von Bussystemen im Automobil stehen andere Anforderungen im Vordergrund als bei Computer-Netzwerken. Neben dem Preis sind vor allem sicherheitsrelevante Eigenschaften wichtig. In den folgenden Kapiteln werden daher nicht ATM, Ethernet oder Token-Ring, sondern der heute im Fahrzeug stark verbreitete CAN-Bus, der für X-By-Wire-Systeme konzipierte FlexRay-Bus und der Multimedia-Bus MOST betrachtet.

¹Der Mercedes SL und die neue Mercedes E-Klasse haben das elektronisch gesteuerte Bremssystem SBC (Sensotronic Brake Control) serienmäßig. Allerdings besitzt SBC eine hydraulische Rückfallebene, so dass die Bremse im Notfall auch ohne Elektronik funktioniert.

2 CAN und TTCAN

Das Controller Area Network - kurz CAN - wurde 1981 von Bosch und Intel entwickelt und findet seit Anfang der 90er Jahre verstärkten Einsatz in Kraftfahrzeugen. Weitere wichtige Anwendungsgebiete sind Haushaltsgeräte, Textilmaschinen oder Aufzüge. Durch die hohe Verbreitung von CAN sind die Komponenten sehr preisgünstig und in hohen Stückzahlen erhältlich.

CAN ist ein Standard der International Organization for Standardization (ISO 11898), der die untersten beiden Layer des OSI-Modelles, den Physical Layer (Bitübertragungsschicht) und den Data Link Layer (Datensicherungsschicht), spezifiziert.

2.1 Merkmale

- CAN verfügt über eine Datenübertragungsgeschwindigkeit von 10 kBit/s bis 1 Mbit/s bei einer Buslänge bis 1km (50 kBit/s). Die theoretisch möglichen 1 Mbit/s reduzieren sich in der Realität jedoch auf effektive 500 kBit/s.
- Die maximale Anzahl der Knoten im Bus beträgt 32.²

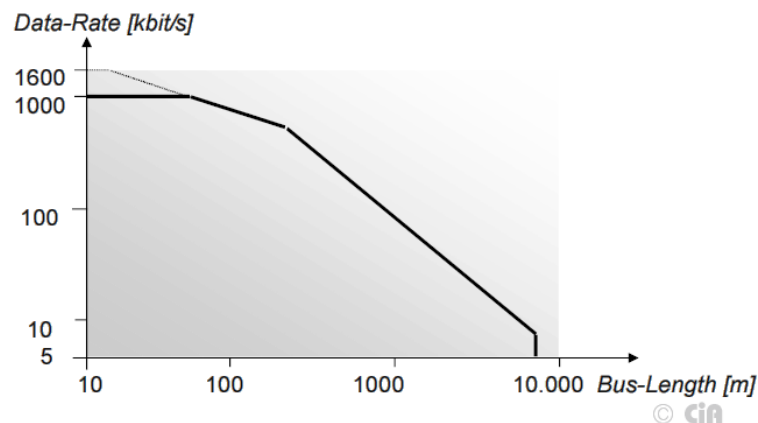


Abbildung 1: Bandbreite über Buslänge beim CAN-Bus (Quelle: CiA, <http://www.can-cia.de>)

- Als physikalisches Medium kommen Twisted-Pair Leitungen zum Einsatz, im Störfall kann auch auf eine Eindrahtleitung zurückgegriffen werden kann, wenn die entsprechenden Schaltvorrichtungen vorgesehen sind.
- CAN ist ein sogenannter Multi-Master-Bus, bei dem jeder Teilnehmer Master werden kann und dann seine Nachrichten versendet.
- Die Kommunikation erfolgt ereignisgesteuert. Will ein Rechner eine Nachricht versenden, so gibt er seinem zugehörigen CAN-Kontroller den Auftrag dazu. Dieser versendet die Nachricht, sobald entweder der Bus frei ist oder die zu versendende Nachricht die höchste Priorität hat.
- Es gibt keine Empfängeradressen im Bus, so dass jeder Teilnehmer alle Nachrichten hört (Broadcast/Multicast). Ebenso ist das Hinzufügen reiner Empfangsstationen während des Betriebs möglich, ohne Änderungen an Hard- oder Software durchführen zu müssen.

²Durch eine Aufteilung in Teilnetze ist eine wesentlich höhere Gesamtzahl möglich.

2.2 Kommunikation

Die CAN-Spezifikation benutzt für Higher- und Lower-Bits die beiden Bezeichner „rezessiv“ und „dominant“. Die Bezeichnung wurde so gewählt, da das dominante Bit immer das rezessive überschreibt.

Es werden vier verschiedene Frame-Formate definiert: Data-, Remote-, Error- und Overload-Frame. Die beiden letzteren kennzeichnen Fehlersituationen, wobei der Error-Frame einen Datenfehler und der Overload-Frame einen Protokollfehler, z.B. bei einem zu geringen Abstand zwischen zwei Frames, markieren.

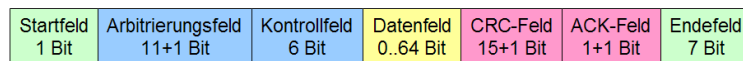


Abbildung 2: Aufbau eines CAN Data Frame's

Der Data-Frame dient zum eigentlichen Versenden von Daten und hat folgende Struktur:

Bits	Bezeichnung	Beschreibung
1	Start of Frame	Markiert den Beginn eines Frames durch ein dominantes Bit.
12	Arbitration Field	Das Arbitrierungsfeld setzt sich zusammen aus dem 11-bittigen Identifier und dem RTR-Bit, welches zur Unterscheidung zwischen Data- und Remote-Frame dient. Beim Data-Frame ist es immer ein dominantes Bit.
6	Control Field	Das Kontrollfeld beinhaltet 4 Bits, die die Länge des folgenden Datenfeldes beschreiben und 2 reservierte Bits für zukünftige Erweiterungen.
0..64	Data Field	Das Datenfeld kann 0 bis 8 Bytes lang sein und besteht aus den eigentlichen Daten. Im Remote-Frame ist das Datenfeld immer 0 Bytes lang.
16	CRC Field	Das CRC-Feld besteht aus 15 Bit, die die Prüfsumme darstellen und einem rezessiven Bit, das CRC Delimiter genannt wird.
2	ACK Field	Das ACK-Feld wird für die Empfangsbestätigung durch die Empfänger benutzt, die auf das rezessive 1. Bit ein dominantes Bit im 2. Bit, dem sogenannten ACK-Slot, schreiben.
7	End of Frame	Der Frame wird abgeschlossen durch 7 rezessive Bits.

Im Gegensatz zum Data-Frame dient der Remote-Frame zum Anfordern von Daten. Er gleicht von der Struktur dem Data-Frame, lediglich das RTR-Bit ist rezessiv und das Datenfeld 0 Bytes lang.

Wie schon im letzten Abschnitt erwähnt, werden bei der Datenübertragung keine Stationen adressiert, sondern der Inhalt einer Nachricht über einen Identifier gekennzeichnet. Dieser bestimmt gleichzeitig die Priorität einer Nachricht, die dann benötigt wird, wenn zwei Knoten gleichzeitig auf den Bus senden wollen.

Ein Knoten im CAN-Netzwerk kann entweder im Sende- oder im Empfangsmodus sein. Während ein Knoten sendet, lesen alle anderen Knoten vom Bus. Die Empfänger entscheiden anhand des Identifiers einer Nachricht, ob diese für sie relevant ist oder nicht. Nach dem Ende der Nachricht können alle Knoten auf den Bus schreiben. Zur Kollisionsvermeidung wird die bitweise Arbitrierung verwendet.

Wollen beispielsweise drei Knoten gleichzeitig eine Nachricht auf den Bus schreiben, so bestimmt die Priorität der Nachricht, welcher Knoten zuerst senden darf. Zunächst schreiben alle drei Knoten das Startfeld des Frames auf den Bus und überprüfen diesen, indem sie vom Bus lesen. Da das Startfeld immer ein dominantes Bit ist, sind alle drei Schreibvorgänge vorerst erfolgreich. Nun beginnen die Knoten bitweise das Arbitrierungsfeld - und damit den Identifier - auf den Bus zu schreiben und jeweils die geschriebenen Bits zu überprüfen. Liest ein Knoten vom Bus ein anderes Bit, als er geschrieben hat, bricht er die Arbitrierungsphase ab und geht in den Empfangsmodus über. Es gewinnt der Knoten die Arbitrierung, der den niedrigsten Identifier und damit die höchste Priorität hat. Dieser kann dann seine vollständige Nachricht senden, während die anderen Knoten auf die nächste Arbitrierung warten müssen.

Mit der bitweisen Arbitrierung wird gewährleistet, dass auch bei dem gleichzeitigen Sendewunsch meh-

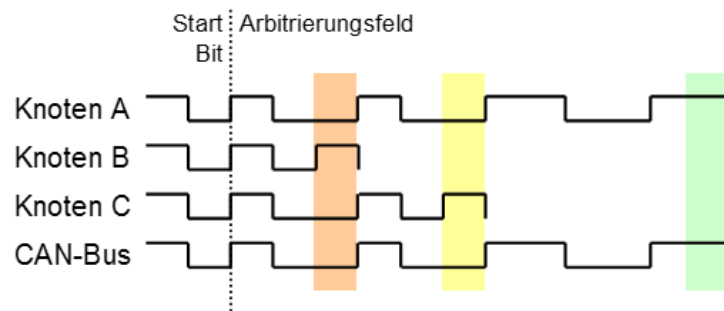


Abbildung 3: Busarbitrierung

erer Stationen keine Buskapazität verloren geht, wie das beispielweise beim Ethernet der Fall ist. Zudem wird eine Priorisierung der Nachrichten berücksichtigt. Ein Kollaps des Kommunikationssystems ist ausgeschlossen, nachteilig ist aber, dass Nachrichten mit niedriger Priorität in Volllastsituationen sehr lange warten müssen.

2.3 Fehlererkennung

Das CAN-Protokoll signalisiert auftretende Fehler über verschiedene Mechanismen:

- Cyclic Redundancy Check (CRC)

Alle versendeten Data- und Remote-Frames werden sendeseitig mit einer Prüfsumme versehen. Empfangsseitig wird die Prüfsumme erneut berechnet und mit der gesendeten Prüfsumme verglichen. Tritt ein Unterschied auf, so muss der Frame erneut gesendet werden.

- Frame-Check

Beim Frame-Check werden die Länge und die Struktur des Frames mit der Spezifikation verglichen. Ein auftretender Fehler, beispielweise ein zu langes Datenfeld, wird als Formatfehler bezeichnet und führt zum erneuten Senden des Frames.

- ACK-Fehler

Im ACK-Slot von Data- und Remote-Frame, dem jeweils 2. Bit des ACK-Feldes, wird von den Empfängern ein dominantes Bit auf den Bus geschrieben, das den Empfang quittiert.

- Bitstuffing

Bitstuffing ist eine Fehlerüberprüfung auf Bitebene. Dabei wird nach 5 aufeinanderfolgenden gleichwertigen Bits ein sogenanntes Stuffbit mit dem komplementären Wert eingefügt. D.h. nach 5 High-Bits folgt immer ein Low-Bit. Die Stuff-Bits werden beim Sender eingefügt und beim Empfänger wieder entfernt und beeinträchtigen nicht die Struktur des Frames. Bitstuffing wird nur bei Data- und Remote-Frames benutzt.

- Monitoring

Alle am CAN-Bus angeschlossenen Knoten überwachen ständig den Buspegel. So können Differenzen zwischen gesendeten und empfangenen Bits erkannt werden.

2.4 TTCAN

Die im CAN-Bus verwendete ereignisgesteuerte Kommunikation verhält sich im Worst-Case-Fall nicht deterministisch. Das heißt, dass über die Sendezeit einer Nachricht keine Aussage getroffen werden kann. In sicherheitsrelevanten Systemen verwendet man daher den zeitgesteuerten Ansatz. Dabei wird jede Nachricht in einen eigenen Zeitfenster versandt.

Einen solchen zeitgesteuerten Ansatz stellt Time Triggered CAN dar. TTCAN basiert im wesentlichen auf der CAN-Spezifikation. Zur Festlegung der Zeitfenster wird eine Systemmatrix verwendet. Eine Beispielmatrix wird in Abbildung 4 gezeigt.

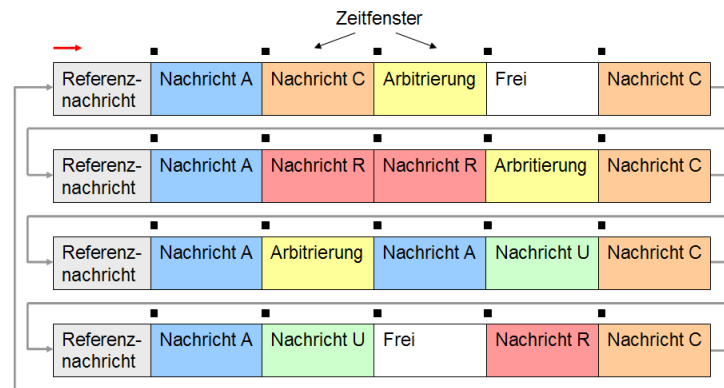


Abbildung 4: TTCAN Systemmatrix

Um Nachrichten zu einem bestimmten Zeitpunkt versenden zu können, muss sichergestellt sein, dass alle Knoten dieselbe Zeit haben. Daher werden in jeder Zeile der Systemmatrix Referenznachrichten versandt, die die Zeit der Knoten auf 0 setzt. Das Versenden der Referenznachrichten übernimmt ein Zeitmaster. Da bei dessen Ausfall die gesamte Kommunikation gestört ist, werden häufig mehrere Zeitmaster verwendet.

In den jeweiligen Knoten werden nur die für sie relevanten Daten der Systemmatrix, also die Sendezeitpunkte gespeichert. Zudem können freie Zeitfenster spezifiziert werden, wodurch eine flexible Erweiterung des Systemes in gewissen Rahmen ermöglicht wird. Ebenso kann in den zeitgesteuerten Ansatz der ereignisgesteuerte Ansatz integriert werden. In diesen Zeitfenstern findet dann wieder die bitweise Arbitrierung statt.

3 FlexRay

FlexRay ist ein noch junges Bussystem, dessen Entwicklung erst seit 1999 von den Initiatoren BMW und DaimlerChrysler vorangetrieben wird. Mittlerweile sind weitere Konzerne - u.a. General Motors, Bosch und Motorola - an der Entwicklung beteiligt. Im Gegensatz zu CAN ist FlexRay bisher lediglich eine Spezifikation, erste Hardware-Implementierungen werden 2002 erwartet, in die Massenproduktion soll es 2003 gehen.

FlexRay soll das Kommunikationssystem der Zukunft werden, wenn es um X-By-Wire-Systeme geht. Dementsprechend standen bei den Anforderungen vor allem eine höhere Datenübertragungsrate, eine deterministische Kommunikation, eine hohe Fehlertoleranz und Flexibilität im Vordergrund.

FlexRay ist bisher nicht ISO/OSI standardisiert, eine Standardisierung wird aber angestrebt. Daneben wurden folgende Ziele spezifiziert (siehe [15]):

- FlexRay ist ein herstellerübergreifendes Bussystem für High-Speed Anwendungen.
- Ein breiter Einsatz des Datenbussystems im Sinne der Standardisierung ist angestrebt.
- BMW und DaimlerChrysler haben keine kommerziellen Interessen bei der Vermarktung des Bussystems.
- FlexRay ist frei verfügbar.

Hinter diesen Zielen steht in erster Linie der Wunsch der höchstmöglichen Verbreitung von FlexRay.

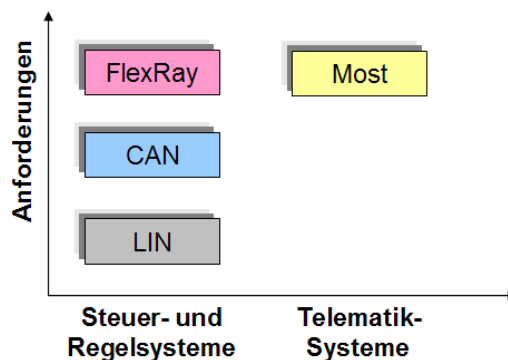


Abbildung 5: Anforderungen von FlexRay, CAN, LIN und MOST

Wie Abbildung 5 zeigt, wird FlexRay das High-End Datenbussystem darstellen. CAN bleibt vor allem für die ereignisgesteuerte Kommunikation zuständig, während LIN (siehe Kapitel 5.1) als kostengünstiger Bus für die Vernetzung von beispielsweise Regen- und Temperatursensoren zum Einsatz kommen wird. Neben CAN, FlexRay und LIN ist MOST der wichtigste Telematik-Bus und wird in Kapitel 4 beschrieben.

3.1 Merkmale

- FlexRay ermöglicht eine konfigurierbare synchrone und asynchrone Übertragung mit hoher Bandbreite bis zu 10 Mbit/s.
- Die Übertragung ist arbitrierungsfrei, um eine Unabhängigkeit von der Buslast zu erreichen.
- FlexRay spezifiziert keinen konkreten Physical Layer, es sollen sowohl elektrische als auch optische Übertragungsmedien unterstützt werden.

- FlexRay ermöglicht verschiedene Bustopologien, die sich alle durch folgende Merkmale auszeichnen:
 - optionale redundante Kommunikation durch mehrere Kanäle
 - alle Knoten sind mit der Fahrzeugbatterie verbunden
 - alle Knoten verfügen über Power-Management
 - alle Knoten lassen sich über den Bus wecken
- Die Knoten besitzen eine globale Zeit. Diese ist zur synchronen Übertragung notwendig, so dass die Nachrichten zu deterministischen Zeitpunkten übertragen werden können.
- Nachrichten haben keine Empfänger- oder Absender-Adressen.
- Die Spezifikation nennt außerdem Flexibilität, Erweiterbarkeit und Projektierbarkeit.

3.2 Kommunikation

Um sowohl eine synchrone als auch eine asynchrone Nachrichtenübertragung zu ermöglichen, wird der Kommunikationszyklus in einem statischen und einen dynamischen Teil aufgeteilt. Die Länge der Zyklen ist flexibel, wobei auch einer der beiden leer sein kann. Vergleichbar mit TTCAN beginnt ein Zyklus mit einer Referenznachricht - bei FlexRay SYNC genannt - die die Synchronisation der Knoten bewirkt.

ID 10 Bit	MUX 1 Bit	SYNC 1 Bit	LEN 4 Bit	CYCLE 8 Bit	DATA 0..88 Bit	CRC 16 Bit
--------------	--------------	---------------	--------------	----------------	-------------------	---------------

Abbildung 6: Aufbau einer Nachricht in Flexray

Die Spezifikation kennt nur ein einheitliches Nachrichtenformat, das wie folgt aufgebaut ist:

Bits	Bezeichnung	Beschreibung
10	ID	Der Identifier einer Nachricht, der im statischen Teil die Slotposition kennzeichnet, im dynamischen Teil dagegen die Priorität. Wenn ID = 0 ist, stellt die Nachricht das SYNC-Symbol dar.
1	MUX	Dieses Bit ermöglicht es, bei gleicher ID unterschiedliche Nachrichten zu verschicken.
1	SYNC	Ist dieses Bit gesetzt, so enthält die Nachricht den sogenannten Frame-Counter, der zur Uhrensynchronisation benutzt wird.
4	LEN	LEN gibt die Länge des Datenfeldes an. Wenn LEN größer als 12 ist, wird der Wert als 12 interpretiert, bei gesetztem SYNC-Feld ist 11 der Maximalwert.
8	CYCLE	Der Frame-Counter zur Uhrensynchronisation. Dieser ist nur gesetzt, wenn SYNC = 1.
0..88(96)	DATA	Das Datenfeld kann 0 bis 12 Bytes bzw. 11 Bytes bei gesetztem SYNC-Feld lang sein.
16	CRC	Die Prüfsumme für den Cyclic Redundancy Check (15 Bit) und ein 0-Bit.

Abbildung 7 stellt eine mögliche Topologie dar, bei der die Knoten vergleichbar mit derzeitigen CAN-Systemen über einen passiven Bus verbunden sind. Damit lassen sich Bandbreiten von 1 - 3 Mbit/s erreichen. Eine alternative Architektur zeigt Abbildung 8. Die als aktiver Stern bezeichnete Topologie erlaubt Bandbreiten bis zu 10 Mbit/s. Der aktive Stern arbeitet als bidirektionaler Repeater. Beide Konfigurationen besitzen einen zweiten Kanal, der sowohl zur Redundanz als auch zur Erhöhung der Bandbreite genutzt werden kann.

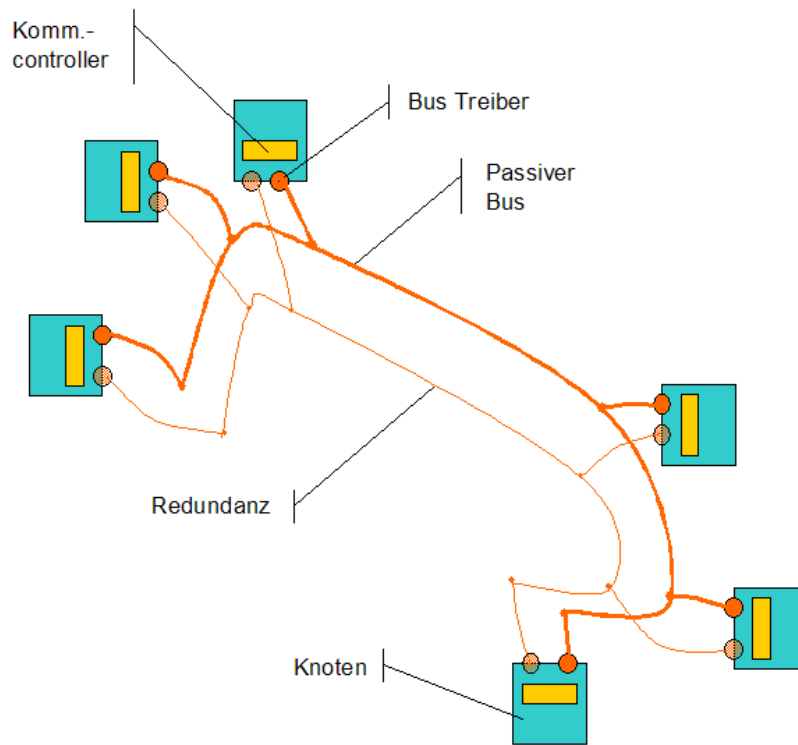


Abbildung 7: Konfiguration mit passivem Bus

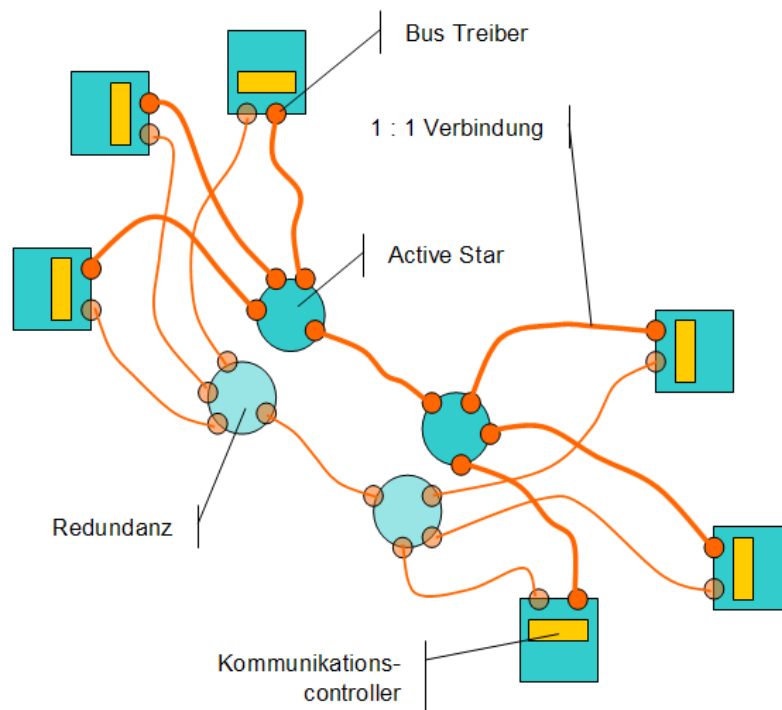


Abbildung 8: Konfiguration mit aktivem Stern

Abbildung 9 erläutert den Ablauf der Kommunikation. Das System besteht aus 5 Knoten, die als aktiver Stern miteinander verbunden sind. Die Knoten A, B und C verfügen über einen zweiten Kanal. Der statische Teil der Kommunikation besitzt 5 Zeitslots, der dynamische 6.

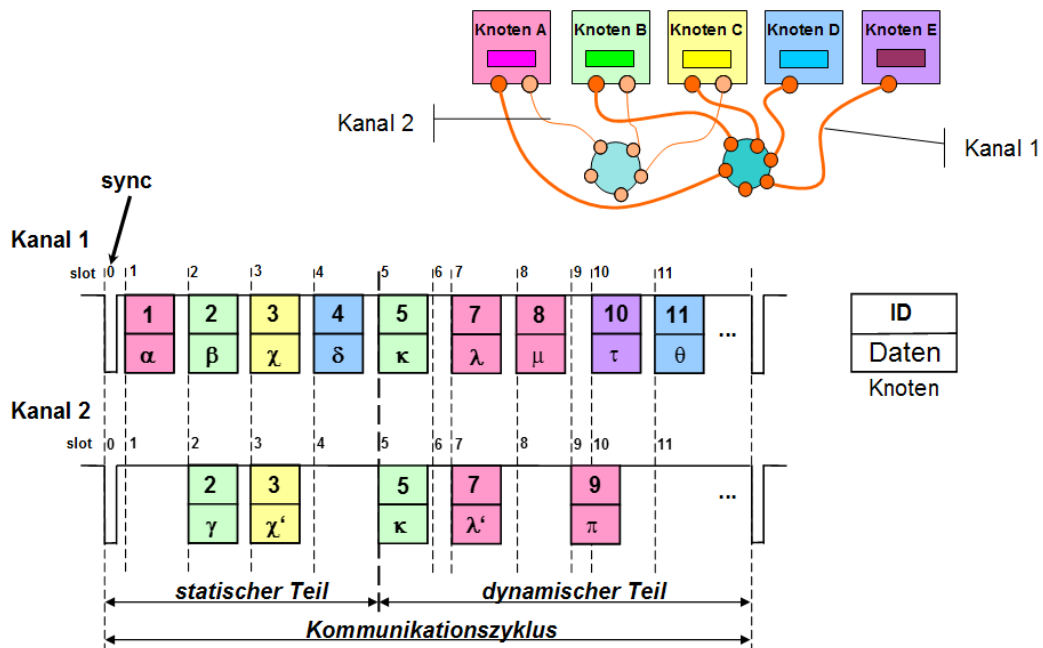


Abbildung 9: Buszugriff im statischen und dynamischen Teil

Die Kommunikation beginnt mit dem SYNC-Signal. Im folgenden statischen Teil sind die Nachrichten an den jeweiligen Slot gebunden, die ID der Nachricht bestimmt den Slot. Der Zugriff erfolgt nach dem TDMA³ Verfahren. Knoten, die an beide Kanäle angeschlossen sind, können den zweiten Kanal zur Redundanz - wie beispielsweise im Slot 3 - oder für verschiedene Nachrichten - wie beispielsweise im Slot 2 - benutzen. Wenn eine Nachricht nicht gesendet wird, verstreicht die Zeit ungenutzt.

Im dynamischen Teil wird der Slotzähler in allen Knoten synchron hochgezählt und der Zugriff erfolgt gemäß FTDMA⁴. Erfolgt für einen Slot keine Sendeanforderung, wird der Slotzähler schon nach kurzer Zeit (timeout) erhöht. Liegt hingegen eine Sendeanforderung vor, so kann die Nachricht gesendet werden und die Knoten stoppen für die Dauer der Übertragung das Hochzählen.

³Time Division Multiple Access. Siehe 6

⁴Flexible Time Division Multiple Access. Siehe 6.

4 MOST

Der Media Oriented Systems Transport (MOST) Bus wird seit 1998 von der MOST Cooperation, in der unter anderem BMW und DaimlerChrysler mitarbeiten, entwickelt. Das Anwendungsgebiet umfasst alle Multimedia-Anwendungen, also Audio, Video, Navigation und Telekommunikation (Internet). MOST ist ISO/OSI standardisiert. Der Bus wird heute bereits im neuen 7er BMW eingesetzt.

Der MOST-Bus zeichnet sich durch eine hohe Bandbreite bei gleichzeitig niedrigen Kosten aus. Die heutige Verwendung von Multimedia-Anwendungen zeigt Abbildung 10.

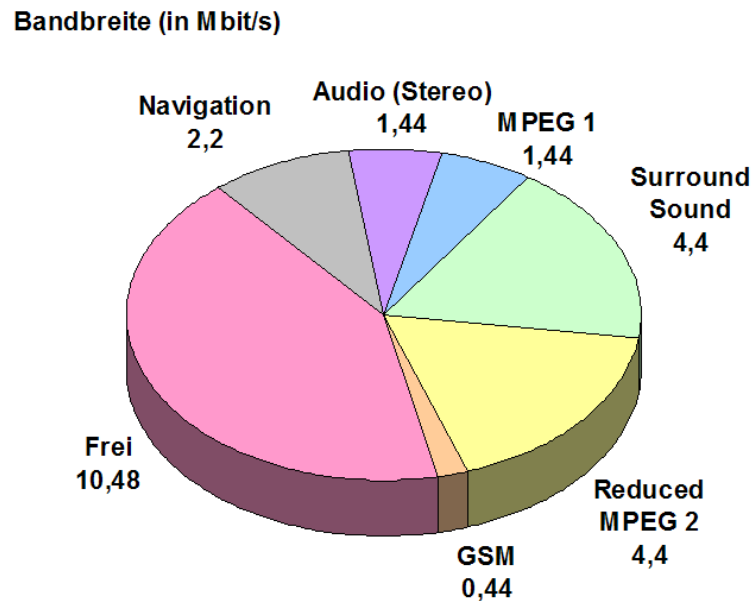


Abbildung 10: Multimedia-Anwendungen im MOST-Bus

4.1 Merkmale

- Hohe Datenraten bis zu 24,8 Mbit/s.
- Es wird die synchrone (bis zu 24 Mbit/s) und asynchrone (bis zu 14,4 Mbit/s) Übertragung von Nachrichten unterstützt. Zusätzlich existiert ein asynchroner Steuerkanal mit bis zu 700 kBit/s.
- MOST ordnet jedem Knoten eine Adresse zu, die zur Kommunikation als Absender- oder Empfänger-Adresse verwendet wird.
- Als Physical Layer kommt ein optisches Medium (Plastic Optic Fibre - POF) zum Einsatz.
- Es werden Ring-, Stern- und Ketten-Topologien unterstützt.
- MOST erlaubt Plug & Play mit bis zu 64 Knoten, die über ein Powermanagement verfügen.
- Echtzeit-Audio-Fähig und Echtzeit-Video-Fähig.
- Weitere Merkmale sind Flexibilität, niedrige Kosten und Kompatibilität mit der PC-Industrie (Netzwerk-Protokolle wie TCP, Streaming-Formate, Plug & Play Standards).

4.2 Architektur

MOST-Systeme werden durch folgende Bereiche beschrieben:

- MOST Interconnect

MOST Interconnect regelt die Kommunikation während der Initialisierung des Systems. Alle Geräte erhalten eine eindeutige ID, die eine physikalische Adresse relativ der Position zum Zeitmaster darstellt. Anschließend werden die Knoten über einen kontinuierlichen Bitstream synchronisiert.

- MOST Devices

Unter diesem Punkt werden alle Geräte, sei es ein Plasma-Bildschirm, ein Lautsprecher oder komplexe Systeme wie das Navigationssystem, zusammengefasst.

- MOST System Services

Die MOST System Services beschreiben alle übrigen Funktionalitäten eines MOST-Systems, die wie folgt unterteilt werden:

- Application Socket
- Basic Layer System Services
- Low Level System Services

Letztere sind für die Kanalzuordnung, Kanal freigabe und Systemmanagement-Funktion verantwortlich, während die ersten beiden für das eigentliche Senden und Empfangen von Daten und weitere Netzwerkmanagement-Funktionen zuständig sind. Abbildung 11 zeigt die genaue Aufteilung.

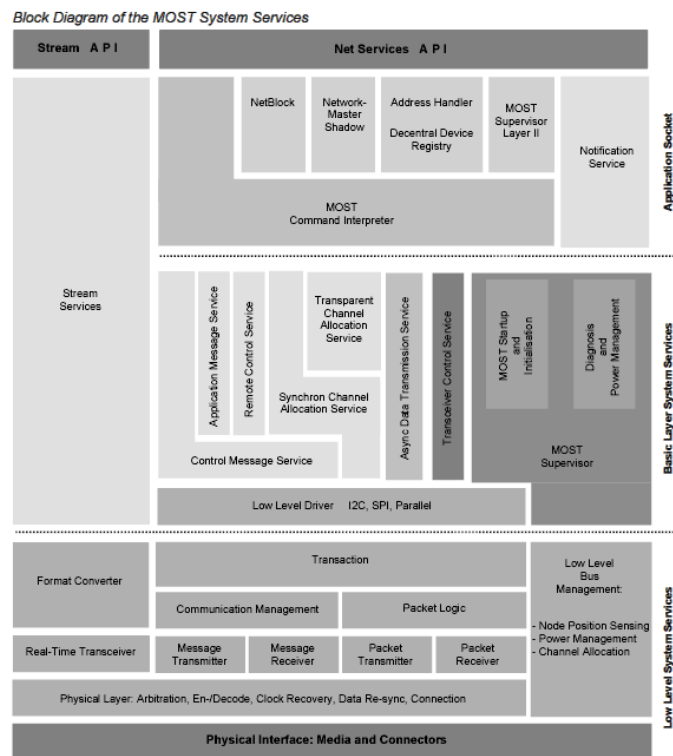


Abbildung 11: MOST System Services (aus [21])

4.3 Kommunikation

Die MOST-Spezifikation verwendet den Begriff „Frame“ nicht wie üblich als Beschreibung einer Nachricht, sondern für einen Kommunikationszyklus. Dieser wird wie folgt beschrieben:

Bits	Bezeichnung	Beschreibung
4	Preamble	Die Preamble dient zur Synchronisierung von Master und Slave.
4	Boundary Descriptor	Bestimmt die Länge der nachfolgenden Bereiche für synchrone bzw. asynchrone Daten. Beide Bereich sind insgesamt 480 Bit breit.
0..480	Synchronous Area	Synchrone Daten werden in diesem Bereich übertragen. Die Zuteilung der Kanäle erfolgt per Time Division Multiplexing, die Bandbreite eines Kanales kann flexibel angefordert werden.
0..480	Asynchronous Area	Im asynchronen Bereich können asynchrone Nachrichten übertragen werden. Die Zuteilung erfolgt per Arbitrierung.
16	Control Frame	Dieser Kanal ist vergleichbar mit der Asynchronous Area, die Bandbreite steht jedoch immer zur Verfügung und ist wesentlich geringer. Er ist vor allem für Diagnose- und Status-Nachrichten gedacht. Die Zuteilung wird ebenfalls über Arbitrierung geregelt.
7	Frame Control	Dient zur Fehlererkennung.
1	Parity	Dient zur Fehlererkennung.

Ein solcher Frame ist immer 512 Bit lang. 16 Frames werden als Block bezeichnet.

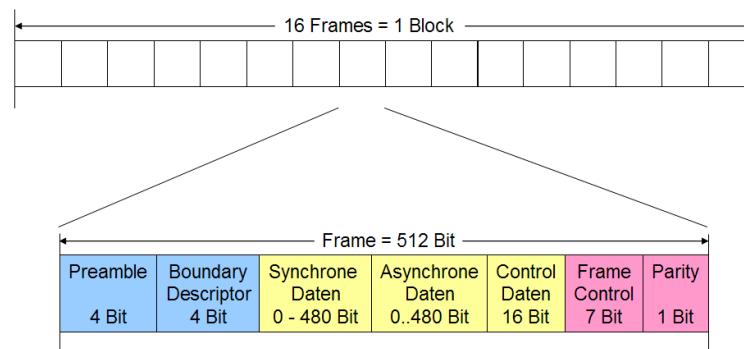


Abbildung 12: MOST Kommunikation

In der Synchronous Area werden vor allem Echtzeit-Daten wie Audio/Video oder Sensor-Daten übertragen werden. Der Sender alloziert dazu einen Kanal, der maximal 60 Bytes/Frame breit sein kann. Für die Übertragung wird jedoch kein Nachrichtenformat spezifiziert. Stattdessen werden mit der sogenannten Routing Engine (RE) die Daten vom Sender zum Empfänger übertragen.

In der Asynchronous Area wird folgendes Format für Nachrichten definiert:

Bits	Bezeichnung
8	Arbitration
16	Target Address
8	Length
16	Source Address
0..384	Data Area
32	CRC

Da der asynchrone Bereich eines Frames eine beliebige Länge haben kann, kann eine Nachricht über mehrere Frames verteilt sein. Mit einem „alternative data link layer“ sind zudem Nachrichten mit einer Datenlänge bis 1014 statt der obigen 48 Bytes möglich.

Letztlich ist auch in der Control Area ein Format für Nachrichten spezifiziert:

Bits	Bezeichnung
24	Arbitration
16	Target Address
16	Source Address
0..144	Data Area + 1 Byte Message Type
16	CRC
16	Transmission Status
16	Reserved

In der Asynchronous Area und in der Control Area werden also im Unterschied zur Synchronous Area Nachrichten übertragen. Zur Kollisionsvermeidung wird wie bei CAN die bitweise Arbitrierung verwendet, im Unterschied zu CAN werden aber Absender- und Empfänger-Adressen spezifiziert.

4.4 Fehlererkennung

Die Fehlererkennung wird von den MOST System Services durchgeführt. Diese bieten Function Blocks für Fehlermanagement auf Applikationsebene. Tritt ein Fehler auf, so kann über einen Remote Access geprüft werden, ob der betroffene Knoten noch richtig funktioniert. Im Fehlerfall kann der Knoten aus dem Netz ausgeschlossen werden, so dass dieser das Netzwerk nicht weiter beeinflusst.

Frames werden durch Parity-Bits, Status-Flags, CRC-Prüfsummen und ACK-Flags überprüft.

5 Weitere Bussysteme

5.1 LIN

LIN - das Local Interconnect Network - ermöglicht eine kostengünstige Kommunikation für Sensoren und Aktoren dort, wo die Bandbreite und Vielseitigkeit von CAN nicht erforderlich ist. Die Kommunikation basiert auf einem seriellen Datenformat. Es gibt einen einzigen Master und mehrere Slaves im Netzwerk. Der Physical Layer besteht aus einem 12V Eindrahtbus. Die Bandbreite beträgt maximal 19,6 kBit/s. Die LIN-Spezifikation ist offengelegt und wird durch ein Konsortium der Automobilindustrie entwickelt.

LIN soll wie in Abbildung 5 gezeigt als „Billigbus“ unterhalb von CAN zur Vernetzung von Temperatur- und Feuchtigkeitssensoren, Aktuatoren und Beleuchtungselementen eingesetzt werden, also überall dort, wo ein CAN-Kontroller aus Platz- und Kostengründen nicht sinnvoll ist.

5.2 TTP

Das zeitgesteuerte Kommunikationsprotokoll TTP (Time-Triggered Protocol) wird seit über 15 Jahren an der Technischen Universität Wien entwickelt. TTP ist in seinen Anforderungen und Merkmalen vergleichbar mit FlexRay. Die Systeme unterscheiden sich im Wesentlichen durch die Fehlertoleranz. FlexRay legt hier den Schwerpunkt mehr auf die Flexibilität der Kommunikation, während bei TTP Fehlertoleranz und Sicherheit priorisiert wurden. Desweiteren existieren für TTP fertige Hardware-Implementierungen⁵. Ein Vergleich zwischen TTP/C und FlexRay ist in [14] zu finden, weitere Dokumente in [9], [10] und [11].

DaimlerChrysler war schon sehr früh an der Entwicklung von TTP beteiligt. Diese Kooperation führte schließlich zur Gründung der TTTech AG. Ende der 90er Jahre fand aber einer Abkehr des Konzerns von TTP statt, desse Gründe offenbar Patent- und Lizenzprobleme waren. Seitdem ist die Frage nach dem zukünftigen, zeitgetriggerten Bus offen. Die FlexRay-Group, vertreten durch BMW, DaimlerChrysler und GM, steht der TTA⁶ Group, die zur Zeit als wichtigsten Partner Volkswagen inklusive Audi hat, gegenüber.

5.3 Firewire / USB

Im Bereich der Telematik-Busse gibt es Bestrebungen, konventionelle Busse wie Firewire und USB in die Automobilelektronik zu übernehmen. Beide erlauben Datenübertragungsraten zwischen 200 und 400 Mbit/s und lassen sich mit Lichtwellenleitern realisieren. Neben den rauen Umgebungsbedingungen - von -40 bis +85 Grad Celsius müssen Übertragungssysteme im Auto aushalten - spricht vor allem das Kommunikationsprotokoll gegen eine Anwendung im Kraftfahrzeug. Firewire und USB benutzen die isochrone Übertragung, die in jedem Knoten schnelle Zwischenspeicher erforderlich macht. Dadurch wird der Einsatz auch zur Kostenfrage.

5.4 Power Line Communication (DC-Bus)

Mit der Power Line Communication wird im Vergleich zu den bisher betrachteten Bussystemen ein anderer Ansatz verfolgt, nämlich die Datenkommunikation über im Auto bestehende Gleichstromleitungen. Dies stellt vor allem wegen der zahlreichen Störeinflüsse eine Herausforderung dar. Warum die Entwicklung einer solchen Kommunikation dennoch in Frage kommt, verdeutlicht die S-Klasse von DaimlerChrysler. Bei rund 1900 Leitungen mit circa 3800 Steckverbindungen und einer Gesamtlänge von 3 km zählt der Kabelbaum zu den teuersten Komponenten im Fahrzeug.

Die israelische Firma Yamar Electronics hat daher den DC-Bus⁷ entwickelt, der zur Zeit eine fehler-

⁵Für TTP/C, die fehlertolerante Variante von TTP, beispielsweise seit 1998.

⁶Time-Triggered Architecture.

⁷DC ist das Symbol für Gleichstrom in der Elektrotechnik!

freie Kommunikation mit bis zu 250 kBit/s ermöglicht. Zur Fehlerkorrektur wird dazu die „Forward Error Correction“ verwendet. Digitale Signale, die beispielsweise von einem CAN-Kontroller kommen, werden mit zusätzlichen Prüfbits (Error Protection Code) versehen. Anschließend wird das Signal in ein analoges Signal moduliert und über Stromleitungen übertragen. Auf der Empfänger-Seite wird das Signal zunächst demoduliert, mit Hilfe der Prüfbits wiederhergestellt und an einen CAN-Kontroller übertragen.

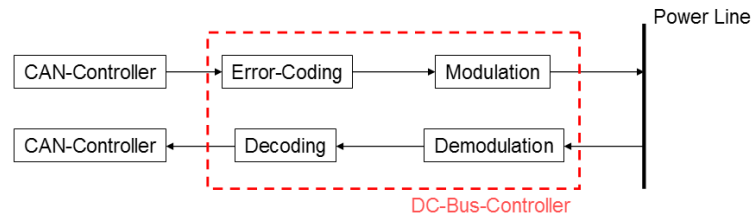


Abbildung 13: Funktionsweise des DC-Bus

In Zukunft will man höhere Datenraten erreichen - Prototypen mit 1,7 Mbit/s sind in Entwicklung. Damit wird das Bussystem auch für Multimedia-Komponenten interessant, so dass CD-Player oder Lautsprecher per Plug & Play an Stromleitungen angeschlossen werden könnten.

6 Vergleich

	LIN	CAN	FlexRay	TTP/C	MOST
Anwendung	Low-Level Kommunikation	Soft-Real-Time-Systeme	Hard-Real-Time-Systeme (X-By-Wire)	Hard-Real-Time-Systeme (X-By-Wire)	Multimedia Telematik
Steuerung	Single-Master	Multi-Master	Multi-Master	Multi-Master	Multi-Master
Nachrichten-Übertragung	synchron	asynchron	synchron asynchron	synchron asynchron	synchron asynchron
Buszugriff	Polling	CSMA/CA	TDMA FTDMA	TDMA	TDM CSMA/CA
Bandbreite	20 kBit/s	500 kBit/s	10 Mbit/s	25 Mbit/s	25 Mbit/s
Datenbyte pro Frame	2, 4 oder 8	0 - 8	0 - 12	0 - 236	0 - 60
Redundante Kanäle	nicht unterstützt	nicht unterstützt	2 Kanäle	2 Kanäle	nicht unterstützt
Fehlererkennung	Checksumme 8-Bit	CRC 15 Bit	CRC 15 Bit	CRC 3 Byte	CRC 4 Byte
Physical Layer	Elektrisch (Single Wire)	Elektrisch (Twisted Pair)	Optisch Elektrisch	Optisch Elektrisch	Optisch

Erläuterungen Buszugriffsverfahren

CSMA/CA steht für Carrier Sense Multiple Access with Collision Avoidance. Ein Client prüft zunächst, ob der Bus frei ist (Carrier Sense), bevor er mit dem Senden beginnt. Senden mehrere Stationen gleichzeitig, so wird durch die Verwendung von Prioritäten eine Kollision vermieden, indem die Sender mit niedrigeren Prioritäten das Senden abbrechen. (siehe Kapitel 2.2).

FTDMA Flexible Time Division Multiple Access wird im asynchronen Teil des Kommunikationszyklus von FlexRay verwendet. Wie auch bei TDMA ist die Kommunikation in Zeitschlitze eingeteilt, wobei die Priorität der Nachricht den jeweiligen Zeitschlitz bestimmt. Zudem werden unbenutzte Zeitschlitze schon nach kurzer Zeit übersprungen. (siehe Kapitel 3.2).

Polling Der LIN-Bus verwendet ein UART-Protokoll (Universal Asynchronous Receiver/Transmitter), bei dem der zentrale Master nacheinander jeden Slave abfragt und dieser daraufhin eine Nachricht versenden kann.

TDM Time Division Multiplexing ist ein Multiplexing-Verfahren, bei dem die Datenströme mehrerer Sender mit niedriger Bitrate zu einem Datenstrom hoher Bitrate durch Aufteilung der Datenströme auf einzelne Zeitschlitze zusammengefasst werden.

TDMA Time Division Multiple Access ist ein Bus-Zugriffsverfahren, bei dem mehrere Stationen auf einen Bus zugreifen. Jeder Station sind dabei ein oder mehrere Zeitschlitze zugeordnet, in denen sie Daten senden dürfen. Der wesentliche Unterschied zwischen TDM und TDMA ist, dass bei TDM der Zugriff durch **einen** Multiplexer, bei TDMA durch **mehrere** unabhängige Stationen erfolgt.

7 Vernetzung von Netzen

Die einzelnen Bussysteme LIN, CAN, FlexRay, TTP und MOST arbeiten mit unterschiedlichen Protokollen und Übertragungsmethoden. Für einen netzübergreifenden Datenaustausch sind daher Verfahren zur Protokollumsetzung notwendig. Dabei lassen sich zwei verschiedene Ansätze verfolgen:

7.1 Supergateway

Als Supergateway wird ein einzelnes, zentrales Gateway bezeichnet, das alle im Fahrzeug verwendeten Bussysteme miteinander verbindet. Dazu muss das Gateway neben den Hardwareanforderungen, wie z.B. unterschiedliche Steckerverbinder, auch die steigende Komplexität der Software zur Protokollumsetzung und damit eine höhere Rechenleistung bewältigen. Probleme ergeben sich zudem bei der Fehlerausbreitung und wie diese in einem solchen System kontrolliert werden kann. Letztlich ist auch die Variantenvielfalt der Fahrzeuge - vom gering ausgestatteten Kleinwagen bis zum vollausgestatteten Luxusmodell - nachteilig, da ein universelles Gateway im Hinblick auf Preis/Leistung, Funktionalität und Testbarkeit ungünstig ist.

7.2 Backbone-Architektur

Bei einer Backbone-Architektur übernehmen innerhalb der einzelnen Netzwerke lokale Gateways, die mit einem Backbone-Bus verbunden sind, die Protokollumsetzung. Dieser Backbone muss entsprechend den Anforderungen dimensioniert sein. Durch diesen dezentralen Ansatz wird eine modulare Vernetzung der Bussysteme ermöglicht, da ein Gateway hard- und softwareseitig nur mit dem Backbone verbunden sein muss. Die Funktionalität, die Kosten und die Skalierbarkeit und damit eine bessere Unterstützung der Variantenvielfalt sind die Vorteile einer solchen Architektur.

8 Software

Die für Kommunikationssysteme erhältliche Software läßt sich grob in die Kategorien

- Modellierung und Simulation
- Design- und Konfiguration
- Analyse
- (Echtzeit)Betriebssysteme

einteilen. Nachfolgend werden die Prozesse Entwurf und Analyse eines Kommunikationssystemes für FlexRay betrachtet.

8.1 Entwurf mit xDesigner

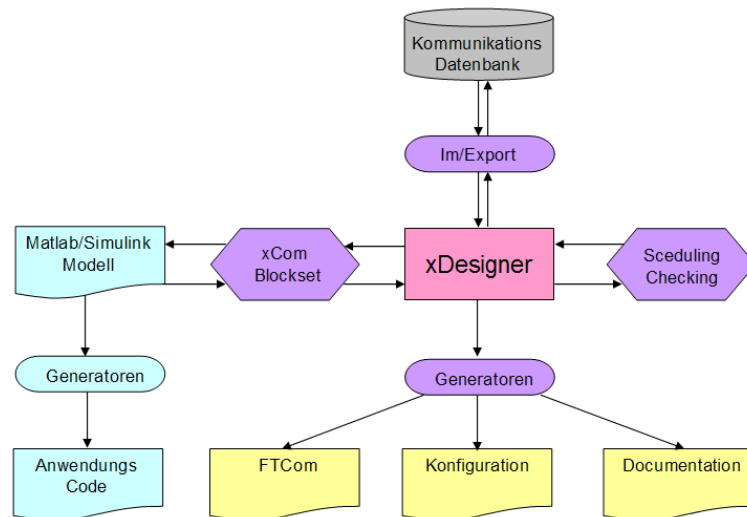


Abbildung 14: Systementwurf mit xDesigner

Der xDesigner der Firma DECOMSYS stellt das zentrale Design- und Konfigurationstool für FlexRay Systeme dar. Ein möglicher Entwicklungsprozess wird in Abbildung 14 gezeigt. Aus einer Datenbank werden die Anforderungen für die Kommunikation in den xDesigner importiert. Dort können nach einem Sceduling-Check Konfigurationsdaten für die Kommunikationskontroller und der FTCOM-Layer generiert werden. FTCOM ist die Kommunikationsschicht im Echtzeitbetriebssystem OSEKtime und stellt standardisierte Schnittstellen zu Diensten wie Datenübertragung, Reintegration und externer Uhrensynchronisation bereit. Die Konfigurationsdaten können aber auch über das xCOM Blockset nach Matlab/Simulink exportiert werden, wodurch dort eine Simulation des Regelalgorithmus inklusive der Eigenschaften der Signalübertragung ermöglicht wird. Es entsteht ein virtuelles Kommunikationssystem, das dieselben Eigenschaften wie das reale System im Zeit- und Wertebereich hat.

Der dargestellte Prozess wird als erweiterter Entwicklungsprozess bezeichnet, da er einen Re-Import des Modelles aus Matlab/Simulink über xCOM in den xDesigner ermöglicht und damit auch die Kommunikationsdatenbank verändert werden kann.

8.2 Analyse mit CANalyzer

Der CANalyzer der Firma Vector Informatik ist ein in der Automobilindustrie weit verbreitetes Analysewerkzeug. Neben der ursprünglichen Anwendung für CAN-Systeme bietet die Firma Erweiterungen für LIN, MOST, FlexRay und zahlreiche andere Bussysteme.

In der Grundausstattung wird mit der Software eine PC-Karte mitgeliefert, die als CAN-Kontroller fungiert. Da für FlexRay entsprechende Kontroller fehlen, gibt es stattdessen ein ServiceModul, das mittels eines Mikrokontrollers und eines FPGAs eine Anbindung an FlexRay realisiert.

Der CANalyzer bietet folgende Grundfunktionen:

- Auflisten des Busdatenverkehrs (Tracing)
- Anzeige von Dateninhalten bestimmter Botschaften
- Interaktives Aussenden vordefinierter Botschaften
- Aussenden aufgezeichneter Botschaften
- Statistik über Botschaften
- Statistik über Busauslastung und Busstörungen
- Aufzeichnen von Botschaften zur Wiedergabe oder zur Offline-Auswertung
- Generieren von Busstörungen

Die weiteren Werkzeuge derselben Firma bieten darüberhinaus noch mehr Funktionen, z.B.:

- Mit CANdb lassen sich vom Bus aufgezeichnete Nachrichten als Nutzinformationen darstellen, beispielsweise aus einem CAN-Frame die Drehzahl des Motors.
- CANgraph übernimmt die grafische Darstellung der Busdaten.
- CANscope dient als Oszilloskop zum Messen und Anzeigen des Buspegels.

Literatur

- [1] BOSCH's Controller Area Network, <http://www.can.bosch.com/>
- [2] CAN in Automation, <http://www.can-cia.de/>
- [3] CAN Beschreibung/TU-München, <http://www.rcs.ei.tum.de/courses/seminar/fieldbus/node59.html>
- [4] Warwick Control Technologies: Introduction to CAN, <http://www.warwickcontrol.com/TICAN.htm>
- [5] FlexRay Group, <http://www.flexray.de/>
- [6] MOST Cooperation, <http://www.mostnet.de/>
- [7] Byteflight, <http://www.byteflight.com/>
- [8] LIN, <http://www.lin-subbus.org/>
- [9] TTTech - Time Triggered Technology, <http://www.tttech.de/>
- [10] TTPforum, <http://www.ttpforum.org/>
- [11] Dokumente zu TTP/TU-Wien, http://www.vmars.tuwien.ac.at/php/pserver/extern/searchpaper.php?search_string=ttp
- [12] Vector Informatik CANalyzer, <http://www.vector-informatik.de/>
- [13] DECOMSYS xCom/xDesigner, <http://www.decomsys.com/>
- [14] Hermann Kopetz, *A Comparison of TTP/C and FlexRay*, Oct. 2001
- [15] R. Belschner, J. Berwanger u.a., *Anforderungen an ein zukünftiges Bussystem für fehlertolerante Anwendungen aus Sicht Kfz-Hersteller*, 9th International Conference on Electronic Systems for Vehicles, Baden-Baden, Oct. 5/6, 2000
- [16] Reinhold Beck, Claas Bracklo, Gerald Faulhaber, Volker Seefried, *Backbone-Architektur: Vom zentralen Gateway zur systemintegrierenden Kommunikationsplattform*, 9th International Conference on Electronic Systems for Vehicles, Baden-Baden, Oct. 5/6, 2000
- [17] *CAN Specification, Version 2.0*, Robert Bosch GmbH, 1991
- [18] R. Belschner, J. Berwanger u.a., *FlexRay Requirements Specification (Draft)*, 07/09/2001
- [19] Hans-Christian von der Wense, *Introduction to LIN*, March 2000
- [20] Dr.-Ing. J. Will Specks, Antal Rajnk, *LIN - Protocol, Development Tools, and Software Interfaces for Local Interconnect Networks in Vehicles*, 9th International Conference on Electronic Systems for Vehicles, Baden-Baden, Oct. 5/6, 2000
- [21] *MOST Specification Framework, Rev. 1.1*, The MOST Cooperation, 1999
- [22] *MOST Specification, Rev. 2.1*, The MOST Cooperation, 2001
- [23] Fuchs, Galla u.a., *Tools und Simulation für FlexRay*, 9th International Conference on Electronic Systems for Vehicles, Baden-Baden, Oct. 5/6, 2000
- [24] Dr.-Ing. Christian Thiel, Dr.-Ing. Rainer König, *MOST Application Framework: Hierarchisches Design von Multimedia Systemen im Fahrzeug*, 9th International Conference on Electronic Systems for Vehicles, Baden-Baden, Oct. 5/6, 2000
- [25] Th. Führer, Dr. W. Dieterle u.a., *TTCAN: Zeitgesteuerter Nachrichtenverkehr im CAN-Netzwerk*, 9th International Conference on Electronic Systems for Vehicles, Baden-Baden, Oct. 5/6, 2000
- [26] Charles J. Murray, *Automotive groups divide on road to data bus*, EETimes, 27/09/2001

- [27] *BMW and Mercedes Choose FlexRay*, The Hansen Report on Automotive Electronics, 01/2001
- [28] Ralf Higgelke, *Der Manta ist zurück*, Design & Elektronik 02/2001
- [29] *Entwurf und Konfiguration von X-By-Wire Netzwerken*, Auto & Elektronik: "media", 01/05/2001
- [30] Franz Fink, *Fault-tolerant bus supports 'by-wire'*, EE Times, 22/06/2001
- [31] Roland Wengenmayer, *FlexRay - ein neuer Datenbus für Autos entsteht*, Technische Rundschau, 18/2001
- [32] Rudolf Skarics, *Guter Draht ist teuer*, Auto-Zeitung 03/2002
- [33] *LIN macht mobil*, Design & Elektronik, 02/2001
- [34] *Multimedia-Netze fürs Auto*, Funkschau 23/99
- [35] *Power Line Communication inside: Stromkabel im Auto als Daten-Highway*, Elektronik-Automotive, Mai 2001
- [36] *X-by-Wire Communications Protocol: Possible FlexRay and TTP Merger*, The Hansen Report on Automotive Electronics, 05/2001