

Dr. Markus Jochim

Zeitig steuern

Sichere Datenübertragung im Automobil

Unter der Haube der neuen Familiensänfte werkeln mehrere Dutzend Mikrocontroller, die im vernetzten Verbund Komfort und Fahrsicherheit steigern. Ihre Kommunikation muss fehlertolerant, deterministisch und schnell passieren, damit die Gurte noch vor dem Crash gestrafft werden und so die Folgen des Aufpralls mildern, statt den Insassen im Nachhinein die Luft zu nehmen. Wie ausgefeilte Netz- und Protokolltechniken dafür zusammenspielen, demonstriert der Newcomer FlexRay.

Ob nun siebzig oder auch nur sieben mit Mikrocontrollern bestückte Steuergeräte als unsichtbare Maschinisten mitfahren, hängt natürlich von Baujahr und Modell ab. Doch selbst in manchem Youngtimer aus den frühen 1990er Jahren steckt schon eine Handvoll Steuergeräte, die über Datenbusse miteinander sprechen. Seitdem hat die Präsenz von Software im Automobil rasant zugelegt: In einem modernen Oberklassefahrzeug können sich durchaus 80 MByte Software auf über 70 Steuergeräte verteilen, die über Sensoren Daten erfassen, Regelalgorithmen ausführen, über Aktoren auf das Fahrzeugverhalten wirken und untereinander Daten austauschen.

Sie bilden Fahrerassistenzsysteme, die vor dem Verlassen der Fahrspur warnen (Lane Departure Warning), auf der Autobahn den Abstand zum Vordermann regeln (Adaptive Cruise Control) oder beim Einparken assistieren. Pre-Crash-Systeme straffen schon vor einem als unvermeidlich eingestuften Zusammenstoß die

Gurte. Ein Bremsassistent lässt auch einen Fahrer, der kein Sicherheitstraining absolviert hat, eine Crash vermeidende Vollbremsung hinlegen, wenn er den Paniktritt aufs Bremspedal bemerkt – ESP verhindert dabei das Ausbrechen des Fahrzeugs. Telematikanwendungen (Navigations- und Verkehrsinformationssysteme, automatischer Notruf) helfen, die Aufmerksamkeit des Fahrers beim Verkehrsgeschehen zu halten, indem sie ihn akustisch mit Informationen versorgen.

Neben solchen mehr oder weniger direkt wahrnehmbaren Anwendungen leisten im Hinter-

grund elektronische Steuerungen für Getriebe, Motor und Fahrwerk oder auch das Energiemanagement essenzielle Dienste zugunsten von Komfort und möglichst niedrigem Kraftstoffverbrauch. Eine Reihe kleinerer Helferlein schaltet bei Dämmerung das Licht ein, aktiviert bei Regen den Scheibenwischer und kümmert sich um die Zentralverriegelung.

All diese Anwendungen resultieren in dramatisch gestiegenen Anforderungen an den Datenaustausch, der bei Edelfahrzeugen ohne weiteres über vier bis sechs verschiedene, über Gateways und Hubs (im Auto auch: Sternkoppler) organisierte Bussysteme läuft. Zu den bekanntesten Vertretern gehören CAN, LIN, MOST und seit kurzem FlexRay – nachfolgend zugunsten holperfreien Lesens mit kleinem R.

Die Bus-Vielfalt stammt zum einen aus der Spezialisierung einzelner Systeme, etwa MOST für Telematik und Multimedia-Anwendungen oder LIN für den Low-Cost-Bereich wie die Sitz- oder Spiegelverstellung. Eine Übersicht technischer Merkmale und Anwendungsbereiche liefert die Tabelle auf Seite 192. Andererseits genügt die Übertragungsrate etablierter Busse – beispielsweise bei CAN maximal 1 MBit/s – heutigen Anforderungen nicht mehr,

weshalb Kfz-Hersteller gelegentlich mehrere Bus-Stränge parallel betreiben.

Warum Flexray?

Dank seiner höheren Datenrate von 2×10 MBit/s kann Flexray nicht nur mehrere Stränge eines älteren Busses ersetzen, auch der Einsatz als Backbone für andere Systeme wird diskutiert. Die Autoelektronik hat in den letzten Jahren in puncto Zuverlässigkeit gelegentlich durch negative Schlagzeilen auf sich aufmerksam gemacht und manchem Autobauer eine Beule am Image beschert. Wenn Flexray die Komplexität und die damit verbundene Fehleranfälligkeit des Bordnetzes reduzieren kann, kommt das aber nicht nur der Autoindustrie, sondern auch dem Kunden zugute.

Nicht nur die Datenrate und die verschiedenen möglichen Topologien – später mehr zu Bus und Stern – waren Anlass zur Flexray-Entwicklung. Als mindestens ebenso wichtig galt die Zunahme sicherheitskritischer Applikationen im Fahrzeug: Vor einigen Jahren sorgten Ideen von Autos ohne Lenkstange (Steer-by-Wire), mit elektrischen Bremsen (Brake-by-Wire) und elektri-



scher Schaltung (Shift-by-Wire) für Enthusiasmus in der Industrie. Inzwischen ist die Erkenntnis gereift, dass zumindest die technisch anspruchsvollsten Applikationen – allen voran Steer-by-Wire – wohl frühestens im nächsten Jahrzehnt in Serie gehen. Noch bestehen zu starke Bedenken bezüglich der Zuverlässigkeit. In der Zwischenzeit beginnen die Fahrzeughersteller, andere sicherheitsrelevante Funktionen elektronisch zu realisieren.

Damit der Fahrer sich auf sein ESP oder den Fahrspurassistenten verlassen kann, musste ein Bus her, der besser für harte Echtzeitanforderungen geeignet ist als zum Beispiel CAN, der aber gleichzeitig wie CAN nicht sicherheitskritische Meldungen spontan senden kann (ereignisgesteuerter Nachrichtenversand).

Offener Standard

Nachdem eine Zeit lang das mit Flexray technisch verwandte und unter anderem im Airbus A380 eingesetzte zeitgesteuerte Bussystem TTP (Time-Triggered Protocol) im Gespräch war, entschloss sich ein großer Teil der Autoindustrie, die Gestaltung des künftigen Busses selbst in die Hand zu nehmen. Weil den Käufer aber kaum interessiert, welcher Bus in seinem neuen Auto den Datenaustausch erledigt, ist die Frage „Flexray oder CAN“

nicht direkt in Kaufanreize übersetzbar. Einen greifbaren Vorteil bringt ein Standard folglich nur über Einsparungen in der Elektrik, doch dazu müssen sich Zulieferer und Hersteller auf eine gemeinsame Grundlage einigen.

Den Antrieb dafür gaben unter anderem BMW, Bosch, DaimlerChrysler, Freescale Semiconductor, General Motors, Philips und Volkswagen. Sie bilden den Kern der zurzeit etwa 125 Mitglieder des Flexray-Konsortiums, das sich um die Ausarbeitung der Details kümmert. Die offengelegten Spezifikationen für Protokoll (Data Link Layer) und Übertragungstechnik (Physical Layer) sind auf den Webseiten des Flexray-Konsortiums frei verfügbar (www.flexray.com).

Derzeit sammeln die Autobauer mit dem noch jungen Bussystem praktische Erfahrungen. Die Vorreiterrolle hat BMW mit der Ausstattungsoption Adaptive Drive im 2007er-Modell des X5 übernommen, das in Deutschland im März auf den Markt kommt. Fünf Flexray-Knoten – vier an den Stoßdämpfern und ein zentrales Steuergerät – passen die Dämpfercharakteristik über elektrisch gesteuerte Bypassventile dynamisch an die Streckenbeschaffenheit an. Zusammen mit einer aktiven Wankstabilisierung sorgt das System für eine verbesserte Fahrdynamik und weniger Seitenneigung in schnell gefahrenen Kurven.

Zyklische Kommunikation

Bussysteme für das Automobil sind stark mit den Feldbussen der Automatisierungstechnik wie Profibus oder Interbus verwandt: Die Hauptaufgabe beider Sorten ist, Steuergeräte zu verbinden, die zyklisch Sensoren auslesen, Regelalgorithmen ausführen und Stellglieder aktivieren. So entsteht – trotz aller Unterschiede im Detail – ebenfalls eine zyklisch organisierte Kommunikation.

Das Neue an Flexray ist, dass es als erster Auto-Bus einen fest eingeteilten (statischen) Zyklusabschnitt mit einem optionalen dynamischen Segment kombiniert. Flexray arbeitet im statischen Teil rein zeitgesteuert: Er wird in ab 1 durchnummerierte Slots fixer Dauer unterteilt. Zugleich legt der Entwickler für das ganze System – in der Flexray-Terminologie

ein Cluster – fest, wie viele Slots das statische Segment umfassen soll. Jeder Knoten darf nur in ihm zugeordneten Slots senden.

Das statische Schema scheint unflexibel, bietet aber den Vorteil der garantierten Latenzzeiten (harte Echtzeit): Als Beispiel geht der Sensorwert, der den Abstand zum vorausfahrenden Fahrzeug anzeigt, garantiert alle 5 ms an das Steuergerät, das notfalls eine Bremsung auslöst. Bei der Zeitsteuerung kann es im Unterschied zur Ereignissteuerung eben gerade nicht passieren, dass eine wichtige Nachricht durch eine Kollision verzögert wird, nur weil ein anderes Steuergerät ebenfalls gerade senden möchte. Denn bei der Automobilsteuerung könnte eine Kollision deutlich teurere oder schmerzhaftere Folgen haben als im Computernetzwerk.

Mal eben schnell

In der Praxis gibt es nun nicht nur sicherheitskritische Informationen, sondern auch solche, auf die man ohne Probleme mal etwas länger warten kann: Ob der angezeigte Benzinverbrauch um 200 statt 20 ms veraltet ist oder zwischen Meldung des Regensensors und Auslösen der Scheibenwischer dreißig statt fünf Millisekunden verstreichen, fällt dem Fahrer nicht auf. Solche Daten transportiert Flexray – obwohl vorrangig als zeitgesteuertes System ausgelegt – mit ereignisgesteuertem Nachrichtenversand im optionalen dynamischen Segment.

Die Slot-ID-Zählung geht im dynamischen Segment nahtlos weiter. Durch kombinierte Zeitmessung und Beobachtung des Busses entwickeln die Knoten ein gemeinsames Verständnis von der jeweils gültigen Slot-ID. Stellt ein Knoten fest, dass die aktuelle Slot-ID mit der ihm zugewiesenen übereinstimmt, darf er eine Nachricht variabler Länge versenden. Anschließend stellen alle Knoten durch passives Lauschen fest, dass die Slot-ID zu erhöhen ist und das Spiel beginnt von vorn.

Zwar können die dynamischen Slots unterschiedlich lang sein, doch muss der Entwickler die Länge des dynamischen Segments trotzdem fest vorgeben, weil nur eine feste Zykluslänge eine feste Sendefrequenz im statischen Segment ermöglicht.

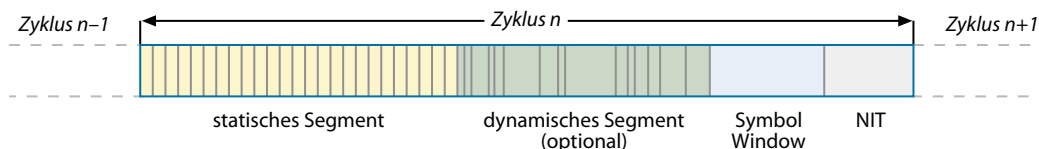
Potenter Knoten

Selbst in einem simplen Flexray-Teilnehmer, kurz Knoten, für Sensoren und Aktoren steckt Rechenleistung, die einen PC vor 20 Jahren gut ausgesehen lassen hätte. Ein rund 4 cm² kleines IC bildet den Mittelpunkt. Darin kümmert sich ein mit vier bis 40 MHz getakteter 16-Bit-Prozessorkern um A/D-Wandler, die analoge Messwerte digitalisieren, mehrere verschiedene Timer, serielle SPI-Schnittstellen zu externen I/O-Bausteinen, Ein- und Ausgänge für binäre Signale und nicht zuletzt die Flexray-Busanschaltung. Flash-Speicher für die Applikation und RAM sind gleich in den Chip integriert, bei der Freescale-Bausteinfamilie MC9S12X beispielsweise bis zu 512 KByte Flash und 32 KByte RAM.

Deshalb kann man nicht sicherstellen, dass die Sendewünsche **aller** dynamischen Teilnehmer in **jedem** Zyklus berücksichtigt werden. Als Ausweg sieht Flexray Cycle-Multiplexing im dynamischen Segment vor: Es weist verschiedenen Knoten in unterschiedlichen Zyklen eine unterschiedliche Slot-ID zu, und damit auch eine geänderte Priorität.

In einem Slot steckt übrigens immer auch genau ein Frame, in dem Flexray maximal 254 Byte transportiert. Das genügt für die Anwendungen im Automobil locker, selbst wenn ein Knoten analoge Messwerte gleich mehrerer Sensoren sendet. Die konkrete Größe gibt der Systemdesigner dabei für jeden Knoten fest vor. Eine individuelle Adressierung, mit der ein Knoten Nachrichten an einen ganz bestimmten anderen schickt, gibt es bei Flexray nicht. Alles geht als Broadcast auf den Bus, und jeder Knoten fischt sich anhand der Slot-ID nur die Informationen heraus, die für ihn wichtig sind.

Den dritten Zyklusabschnitt bildet das Symbol Window. Es dient zur Übermittlung von Flexray-internen Steuerungsinformationen. Beim Hochfahren des Systems, quasi dem Booten, wecken die Knoten einander durch Senden bestimmter Bitfolgen (Symbole) auf.



Ein Flexray-Zyklus läuft mit fester Frequenz, beispielsweise 400 Hertz. In den Slots des statischen Teils senden Knoten, die harte Echtzeitfunktionen erledigen und ihre Daten garantiert übermitteln müssen.

Während der den Zyklus beschließenden Network Idle Time (NIT) können die Knoten schließlich die individuelle Gangabweichung ihrer Taktgeber ausgleichen und sich auf eine gemeinsame Systemzeit einigen (mehr dazu unten).

Ein typischer Flexray-Zyklus dauert vielleicht 2,5 Millisekunden, sodass ein Knoten seine Daten 400-mal pro Sekunde auf die Reise schicken kann. Ist das für einzelne Funktionen noch zu wenig, kann der Systementwickler dem jeweiligen Knoten mehrere geschickt auf den Zyklus verteilte Slots zuzuordnen, um zum Beispiel alle 1,25 ms eine Nachricht abzusetzen.

Zeit versus Ereignis

In einem rein Event-gesteuerten Bussystem sendet jeder Teilnehmer, wann immer seiner Anwen-

dung danach ist. Kollisionen gehören da zum Alltag. Gegenüber einem zeitgesteuerten Zugriff bietet das Verfahren verschiedene Vorteile:

Der Planungsaufwand sinkt, denn nicht jede von einer beliebigen Applikation auch nur sporadisch versendete Nachricht muss im Vorhinein in einem akribisch berechneten statischen Schedule berücksichtigt werden. Den gerade bei zeitgesteuerten Systemen entstehenden Aufwand zum Feineinstellen des Datenfahrplans sollte man auch bei gegebener Tool-Unterstützung nicht unterschätzen. Sowohl die im Verlauf der Entwicklung unvermeidbaren Änderungsforderungen als auch die nötige Abstimmung mit Zulieferern verschärfen dieses Problem.

Das System kann die Bandbreite flexibler nutzen: Will ein Knoten in einem bestimmten Zyklus des zeitgesteuerten Busses keine Daten senden, verstreicht sein Slot ungenutzt. Bei einem rein ereignisgesteuerten System teilen sich alle Knoten die zur Verfügung stehende Bandbreite. Solange sendewillige Knoten im System sind, wird die zur Verfügung stehende Bitrate genutzt.

Das lässt sich mit der Belegung einer Festplattenpartition vergleichen: Viele Betriebssysteme teilen den Platz in stets gleich großen Clustern zu. Dadurch entsteht bei zahlreichen Dateien, die deutlich kleiner als ein Cluster sind, viel „Verschnitt“ – die Platte ist schneller voll. Könnte das Betriebssystem die Clustergröße dem Bedarf anpassen, ließe sich die Festplatte wesentlich effizienter nutzen.

Ein ereignisgesteuerter Bus kann Nachrichten unterschiedlicher Länge versenden. Im rein zeitgesteuerten System haben dagegen typischerweise alle Sendeslots dieselbe Dauer und damit eine fixe Nachrichtenlänge. Sehr kurze Nachrichten lassen Teile eines Slots ungenutzt und für sehr lange Nachrichten müssen möglicherweise gleich mehrere Slots reserviert werden.

Diesen Vorzügen steht bei zeitgesteuerten Systemen im Wesentlichen der garantierte Nachrichtenversand gegenüber: Der von einem Sensor gemessene Abstand zum vorausfahrenden Pkw muss mit einer garantierten Frequenz an das Steuergerät fließen. Um eine Entscheidung treffen zu können, muss es ebenso

die eigene, aktuelle Geschwindigkeit kennen, die ein anderes Steuergerät auf den Bus gibt.

Mit Zeitgefühl

Striktes Timing setzt voraus, dass die Knoten eines Flexray-Strangs eine gemeinsame „Wahrnehmung“ der Zeit haben: Nur wenn alle Knoten die jeweils gültige Slotnummer wissen und Start und Ende des Slots hinreichend genau kennen, funktioniert die Zeitsteuerung kollisionsfrei. Nun arbeitet ein Flexray-Cluster aber als Multi-Master-System. Es gibt keinen einzelnen Knoten, der als „Uhrmacher“ für alle übrigen die Systemzeit festlegt. Jeder Knoten kann daher zunächst nur von seiner lokalen Uhr (Oszillator) ausgehen und hat lediglich eine lokale Sicht. Die globale Sicht müssen die Knoten durch Austausch und Abgleich von Zeitinformationen schaffen.

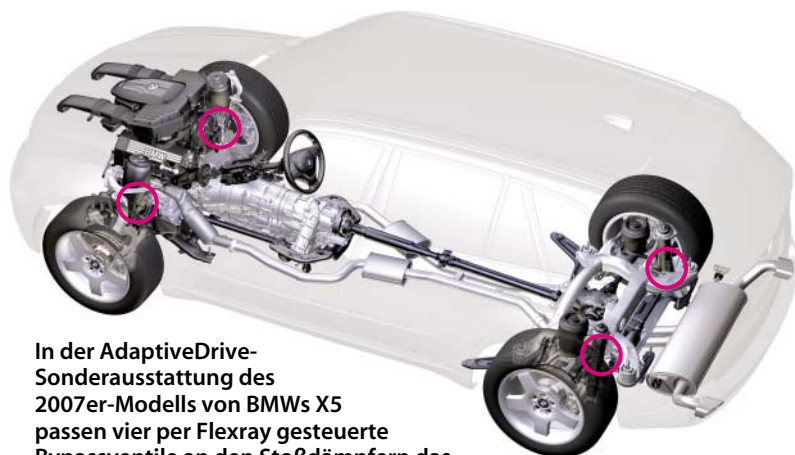
Weil die lokalen Oszillatoren aber schon fertigungsbedingt individuell von der Nominalfrequenz abweichen, und beim Auto im Betrieb zwischen Kälte und Hitze auch noch deutlich temperaturabhängig wegdriften, besteht das globale Zeitempfinden nur temporär. Regelmäßiges Nachstellen wird nötig.

Alle paar Wochen zur Tageschau-Zeit bemerken Sie, dass die von Oma geerbte Wanduhr mal wieder vorgeht. Die momen-

Fahrzeug-Bussysteme				
	CAN (Controller Area Network)	LIN (Local Interconnect Network)	MOST (Media Oriented Systems Transport)	Flexray
Charakterisierung	weiche Echtzeitanforderungen	Low-Cost-Bereich	Multimedia und Telematik	harte Echtzeitanforderungen, sicherheitskritische Systeme
typische Anwendungen	High Speed CAN: Getriebesteuerung, Motorsteuerung, Bremsensteuerung, Fahrwerksteuerung; Low Speed CAN: vergleichbar zu LIN	Regensensor, Lichtsensor, Sitzverstellung, Schiebedach, Zentralverriegelung, Scheibenwischer, Fensterheber	Echtzeit-Video- und Audio-Streams, GPS, Telefon, Rückfahrkamera, Digital-Radio, Internet	sicherheitskritische Anwendungen, Einsatz als Backbone für verschiedene Bussysteme geplant, Einsatz in heutigem Anwendungsgebiet von High-Speed-CAN geplant, langfristig: Brake-by-Wire, Steer-by-Wire
Topologie	Linienstruktur, Stichleitungen sowie Ringe und Sterne möglich	Linienstruktur	meist als Ring ausgelegt, auch Sterne oder Doppellinien möglich	Linienstruktur, Sterne, Hybrid (Linie und Stern) ¹
Physical Layer	Zweidrahtleitung oder Eindrahtleitungen je nach Datenrate	Eindrahtleitung	meist Lichtwellenleiter, teilweise auch elektrische Leiter (Most 50)	Zweidrahtleitung ²
maximale Datenrate	High Speed: 1 MBit/s, Low Speed: 125 kBit/s	20 kBit/s	23 MBit/s Bitstreaming, 768 KBit/s Steuerkanal	2 × 10 MBit/s ³
Buszugriff	Multi-Master-System; ereignisorientierter Nachrichtenversand; CSMA-CA ⁴	Single Master / Multiple Slave; Master definiert Reihenfolge des Sendezugriffs der Slaves durch Polling	Timing Master legt die zyklische Weitergabe von Most-Frames fest	Multi-Master-System: Teilmenge der Knoten (Sync Nodes) vereinbart die Systemzeit; zeitgesteuerter Buszugriff; optional: ereignisgesteuerter Versand
Nutzdatenmenge pro Frame	bis 8 Byte	bis 8 Byte	bis 60/117 Byte (MOST 25/50)	bis 254 Byte
relative Kosten ⁵	0,5 (Low-Speed) / 1,0 (High-Speed)	0,5	5,0	2,5
Besonderes	CAN-Nachrichten enthalten keine Zieladressen, sondern einen Identifier, der den Nachrichteninhalte für mögliche Empfänger kennzeichnet; keine garantierten Latenzzeiten; Eignung für weiche Echtzeitsysteme	Low-Cost-System für niedrige Anforderungen an Datenrate; garantierte Latenzzeiten; durch Versand von Synchronisations-Informationen im Header sind Knoten ohne bzw. mit minderwertigen Quarzen einsetzbar	Datenrate auf bis zu 60 Streaming-Kanäle aufteilbar; Plug&Play-Integration von Knoten möglich	Eignung für harte Echtzeitsysteme; zahlreiche Fehlertoleranzmechanismen; redundante Kanäle

¹ beide Flexray-Kanäle können eine voneinander weitgehend unabhängige Topologie aufbauen
² Spezifikation sieht optionalen Einsatz von Lichtwellenleitern vor, aber bisher nicht auf dem Markt
³ optionaler Einsatz von zwei Kanälen zu je 10 MBit/s, zweiter Kanal kann redundant arbeiten oder die verfügbare Datenrate steigern

⁴ Carrier Sense Multiple Access: CAN-Nachrichten sind über Identifier priorisiert, bei Kollisionen bricht der Sender einer niedrigeren Nachricht verlustfrei ab
⁵ Faktor bezogen auf die Kosten eines High-Speed-CAN-Knotens



In der AdaptiveDrive-Sonderausstattung des 2007er-Modells von BMWs X5 passen vier per Flexray gesteuerte Bypassventile an den Stoßdämpfern das Fahrwerksverhalten an die Strecke an.



Quelle: BMW Group

Beim Stellen der Pendeluhr dient das TV-Programm als Referenz, doch solch eine zentrale Uhr gibt es bei Flexray ja nicht. Statt dessen bekommt eine Teilmenge aller Knoten eine Nebenaufgabe: Jeder dieser Sync-Knoten verschickt in jedem Zyklus in einem ihm zugeordneten statischen Slot ein Sync-Frame: ein normales Daten-Frame, in dem das Sync-Bit gesetzt ist. Die anderen Teilnehmer – auch die Sync-Knoten selbst – vergleichen den Zeitpunkt jedes Sync-Empfangs mit dem dafür erwarteten Termin und bekommen so eine Liste von Differenzen, aus denen sie mit dem FTM-Algorithmus (Fault Tolerant Midpoint) ihren Offset-Korrekturwert errechnen [3].

tane Abweichung ist mit einem beherzten Dreh am Minutenzeiger schnell korrigiert, doch das Beheben des ursächlichen Gangunterschieds (Frequenzabweichung) per Rädchen am Pendel erfordert Fingerspitzengefühl und Geduld. Ein Flexray-Knoten korrigiert seine interne Uhr mit den gleichen Methoden: Die Offset Correction entspricht dem Einstellen der aktuellen Zeit, die Rate Correction dem Drehen am Rädchen der Pendeluhr.

Die Synchronisation setzt bei den sogenannten Microticks an. Flexray tastet seinen 10-MBit/s-

Bus nominell mit 80 MHz ab, ein Sample-Tick entspricht 0,0125 µs. Typischerweise zwei davon bilden einen Microtick. Bei einer Zykluszeit von zum Beispiel 5 ms treten so 200 000 Microticks pro Zyklus auf. Einen Offset gleicht der Knoten schlicht durch einmaliges Hinzufügen oder Weglassen einiger Ticks in der Network Idle Time am Zyklusende aus. So verschiebt er den eigenen Start des nächsten Zyklus und passt sich an die anderen Knoten an.

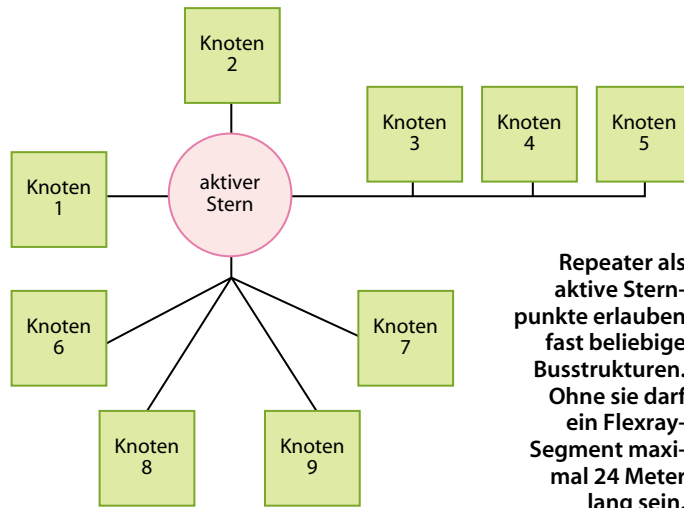
Während Sie nun in Ihrem Wohnzimmer dem Taktgeber (Pendel) durch das Drehen am

besagten Rädchen unmittelbar eine veränderte Frequenz aufzwingen, kann der Flexray-Knoten den Takt seines lokalen Oszillators nicht beeinflussen. Die Frequenzkorrektur muss deshalb indirekt erfolgen, indem der Knoten künftig einige Microticks mehr oder weniger gleichmäßig über den gesamten Zyklus verteilt und ihn so geringfügig verlängert oder verkürzt.

E pluribus unum

Nun fehlen noch die konkret anzuwendenden Korrekturwerte.

Sind mehr als zwei Sync-Knoten aktiv, verwirft FTM Extremwerte, damit stark fehlgehende Uhren das System nicht aus dem Tritt bringen. Bei drei bis sieben Sync-Werten fliegen Minimum und Maximum heraus, bei mehr als sieben auch die zweitgrößten beziehungsweise -kleinsten Messwerte. Das dann verbleibende Maximum und Minimum mittelt



Repeater als aktive Sternpunkte erlauben fast beliebige Busstrukturen. Ohne sie darf ein Flexray-Segment maximal 24 Meter lang sein.

der Knoten und sieht das Ergebnis als eigene Abweichung an.

Die erzielbare Konvergenz ist in der Literatur gut untersucht; ein Flexray-System synchronisiert sich nach dem Systemstart normalerweise innerhalb weniger Zyklen. Die Rate Correction läuft sehr ähnlich ab: Die Knoten messen die Zykluslänge anhand der Zeitstempel zweier aufeinanderfolgender Zyklen und vergleichen sie mit ihrer eigenen Zykluslänge. Dann kommt abermals der FTM-Algorithmus zum Zug.

Mit Offset und Rate Correction hält ein Flexray-Segment seinen Gleichschritt. Doch dabei kann seine Zeit immer noch allmählich von der anderer Systeme im Auto wegwandern. Dieses sogenannte „Clusterdriften“ dämpft Flexray durch Beschränken der Korrekturwerte.

Präzision versus Datenrate

Präzision heißt bei Flexray die maximale Zeitabweichung, die zwei beliebige Knoten trotz Synchronisation aufweisen können. Sie beeinflusst die im zeitgesteuerten System erreichbare Datenrate entscheidend, denn je höher die Differenz ist, desto kleiner wird das Fenster, in dem tatsächlich Daten fließen können (siehe Bild auf S. 195).

Zu der Differenz P zwischen den Uhren zweier Knoten kommt die Signallaufzeit D auf dem zwischenliegenden Kabel und in den Transceivern und Sternen hinzu. Auch im schlimmsten Fall – bei maximal nachteilendem Sender und maximaler Signalverzögerung – muss der Empfänger den Frame noch

vollständig im zugehörigen Slot wahrnehmen. Der Flexray-Entwickler konfiguriert also die Idle-Zeit so, dass sie mindestens P+D entspricht. Muss er wegen minderwertiger Quarze oder Signalverzögerungen durch zu lange Leitungen respektive langsame Bus-Transceiver jedes Frame mit längeren Idle-Zeiten rahmen, sinkt die erzielbare Datenrate weiter ab.

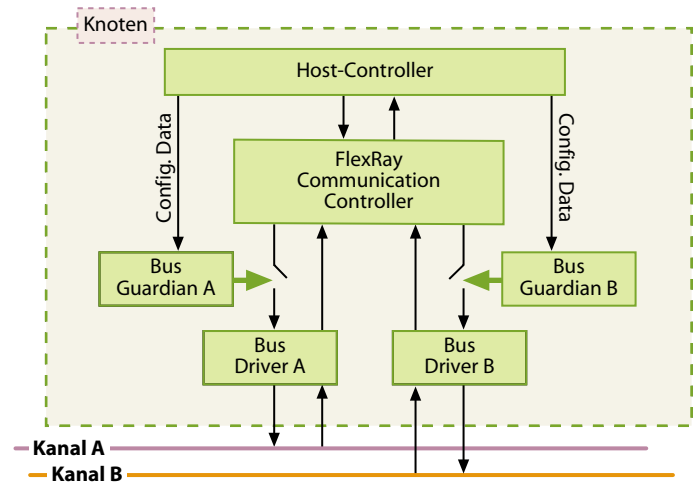
Als praktisches Beispiel darf die Signallaufzeit D bei zwei Knoten, die jeweils über 7,5 Kabelmeter an einem aktiven Stern hängen, laut Flexray-Spezifikation bis zu 1,05 µs erreichen. Ein typischer Wert für die ebenfalls von der Signallaufzeit abhängige Präzision P kann für ein solches System bei etwa 4,5 µs liegen, wodurch sich am Slot-Ende eine zu konfigurierende Idle-Zeit von immerhin 5,5 µs ergibt. Die reine Signallaufzeit ist folglich nicht vernachlässigbar klein.

Das zeigt sich, wenn man mit bestimmten Annahmen über die Anzahl der Sternpunkte, Kabellängen und Transceiver-Geschwindigkeiten spielt. Eine um 1 µs erhöhte Signalverzögerung kann die Nettodatenrate schon mal um etwa 0,4 bis 0,5 MBit/s senken. Dieser Einfluss macht sich angesichts der für einen 10-MBit-Kanal typischen Nettodatenraten von 3,5 bis 4 MBit/s deutlich bemerkbar. Allerdings streut der Einfluss stark mit der Architektur des konkreten Systems. Ein Flexray-Entwickler ist folglich gut beraten, in seinem Projekt Worst-Case-Simulationen mit variierender Idle-Zeit durchzuführen.

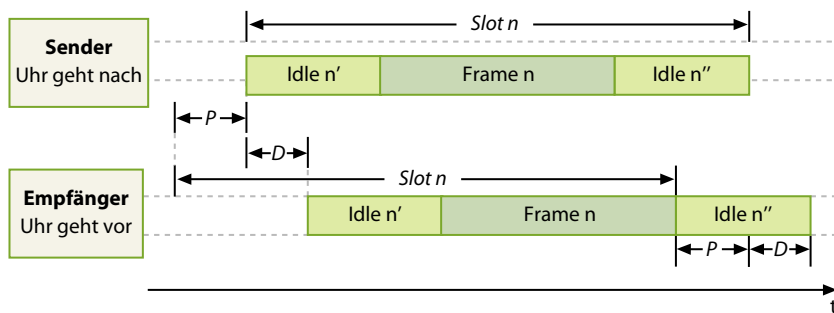
Die einzustellende Idle-Zeit ergibt sich übrigens genauso, wenn die Sender-Uhr gegenüber dem Empfänger vorgeht. Spielt man dieses Szenario im Detail durch, kommt man zu der zunächst absurd erscheinenden Erkenntnis, dass sich eine garantierte Mindestsignalverzögerung hier förderlich auswirkt, weil sie dazu beiträgt, die Idle-Zeit klein zu halten.

Flexibler Gestaltwandler

Die bei Flexray standardmäßig vorgesehenen zwei Übertragungskanäle erlauben ungewöhnliche Konstruktionen. Schon bei einem Kanal kann ein Entwickler klassische lineare Busse mit aktiven und passiven Sternen kombinieren. Die maximale Strecke ohne aktiven Hub darf 24 Meter zwischen zwei Knoten betragen; mit maximal zwei Hubs steigt die Distanz auf 72



In einem typischen Flexray-Chip steckt neben dem Communication Controller auch ein Mikrocontrollerkern, auf dem die Applikation unter einem Echtzeitbetriebssystem wie etwa OSEK-Time läuft.



Ausreichend dimensionierte Idle-Zeiten stellen sicher, dass ein Frame auch trotz Uhrenabweichung und Signalverzögerung im zugehörigen Slot des Empfängers landet.

Meter. Auf elektrischer Ebene arbeitet Flexray mit Differenzsignalen auf einer Doppelader, die wahlweise in geschirmten oder ungeschirmten Kabeln läuft.

Der zweite Kanal muss nicht zwangsweise die gleiche Struktur bekommen: Bei besonderen Sicherheitsanforderungen kann man ihn als redundanten, unabhängigen Übertragungsweg aufbauen, der nur die für eine besonders wichtige Funktion nötigen Knoten verbindet. Alternativ kann man ähnlich wie bei der Link Aggregation bei Ethernet-LANs die Kanäle zwecks Durchsatzverdoppelung parallel schalten. Ein im Flexray-Knoten autonom arbeitender Bus Guardian verhindert, dass ein vielleicht durch Glitches auf der Versorgungsspannung in die Irre geleckter Controller zur Unzeit Daten auf den Bus gibt und so die Zeitsteuerung im statischen Segment durcheinanderbringt.

Störschwelle

In der übertragungstechnisch gesehen „lauten“ Kfz-Elektrik sind Bitfehler durch kurzfristige Störungen (Glitches) an der Tagesordnung. Solche Ausrutscher

eliminiert Flexray mit achtfacher Überabtastung: Aus den jeweils letzten fünf Samples bildet es zunächst eine Mehrheitsentscheidung (3 aus 5), das Voted Signal. An dessen fallender Flanke synchronisiert sich der von eins bis acht laufende Sample Counter. Hat dieser die Stufe fünf erreicht, geht der Empfänger von einem stabilen Zustand aus und reicht das momentane Voted Signal als empfangenes Bit weiter. Die geringfügige Verzögerung (Voting Delay) stört nicht weiter.

Fehlertoleranz

Neben der Zeitsteuerung verfügt Flexray über eine Reihe weiterer Eigenschaften, die das System für die Übertragung sicherheitskritischer Informationen prädestinieren. So sorgt etwa der fehlertolerante Startup-Mechanismus dafür, dass beim Hochfahren des Clusters nur die vorab festgelegten Cold-Start-Nodes Sync-Frames an die anderen Knoten senden, und das wiederum kollisionsfrei.

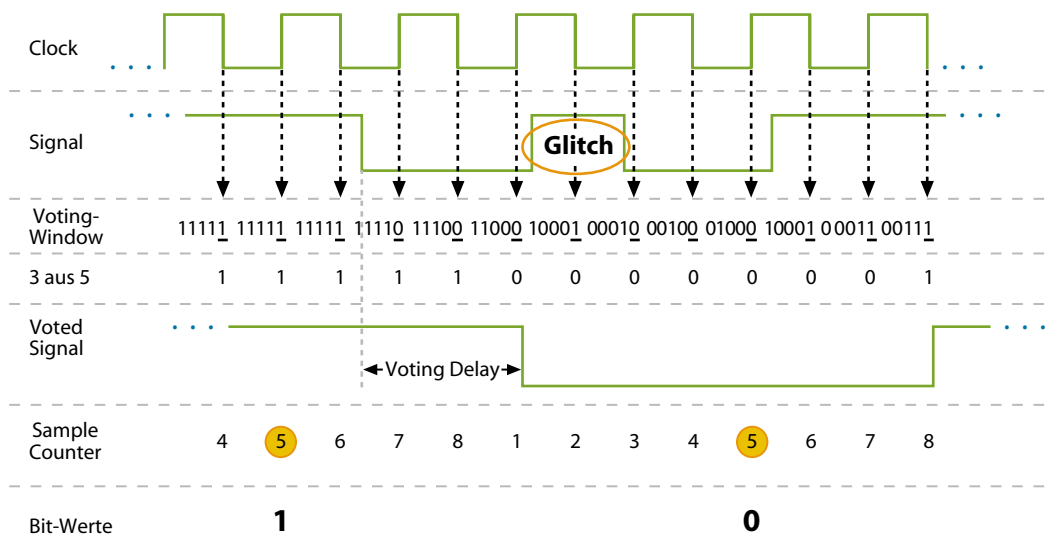
Die Fehlertoleranz ist so weit eingearbeitet, dass sogar Restsysteme dann noch ordnungsgemäß starten und kommunizieren,

wenn die Oszillatoren einzelner Knoten stark abweichen oder ganz ausfallen. All das bildet eine gute Grundlage für zuverlässige Kommunikation zwischen Steuergeräten, die letztlich dem Komfort und der Verkehrssicherheit zugute kommt.

Flexray hat die kritische Masse von Nutzern und Anbietern schon jetzt erreicht. Das bedeutet aber nicht, dass die Tage der etablierten Systeme wie CAN, LIN oder MOST gezählt sind. Denn unterschiedliche Kostenstrukturen und die Spezialisierung der einzelnen Systeme garantieren ihnen ihre Marktsegmente. (ea)

Literatur

- [1] FlexRay Communications System Protocol Specification, Version 2.1, Revision A, Flexray Consortium, 12/2005, www.flexray.com
- [2] FlexRay Communications System Electrical Physical Layer Specification, Version 2.1, Revision A, Flexray Consortium, 12/2005, www.flexray.com
- [3] Jennifer Lundelius-Welch, Nancy A. Lynch, A new fault-tolerant algorithm for clock synchronization, Information and Computation, Volume 77, 04/1988



Kurze Störungen eliminiert Flexray schon im Empfänger mittels gleitender Auswertefenster.

