

6.6 FlexRay

6.6.1 Einleitung

Mit der zunehmenden Menge von Datenkommunikation zwischen den elektronischen Steuereinheit (ECUs) des Fahrzeugs, ist es wichtig eine hohe Datenrate zu erzielen. FlexRay lässt eine Datenrate von ungefähr 10Mbit/sec zu. das Design des Protokolls erlaubt jedoch, dass viel höhere Datenraten erzielt werden können.

FlexRay, als skalierbares Kommunikationssystem, erlaubt die synchrone und asynchrone Übertragung von Daten. Abhängig von den Anforderungen der jeweiligen Applikation, kann die Kommunikationsschleife synchron, asynchron oder eine Mischung von beiden sein. Die synchrone Datenübertragung ermöglicht die zeitgesteuerten Kommunikation, die den Anforderungen von sicherheitsrelevanten Systemen entspricht. Die asynchrone Übertragung basiert auf den Grundlagen des Byteflight™ Protokolls, und lässt jeden Knotenpunkt die volle Bandweite für Event gesteuerte Kommunikationen verwenden.

FlexRays synchrone, deterministische Datenübertragung hat eine garantierte minimale Meldungslatenzzeit und Meldungschwankung. FlexRay unterstützt Redundanz und eine fehlertolerante verteilte Uhrensynchronisierung für eine globale Zeitbasis. Somit werden die Zeitpläne aller Netzknoten innerhalb eines vorbestimmten engen Präzisionsfenster gehalten.

6.6.2 Geschichte

FlexRay ist ein herstellerübergreifendes Bussystem für Highspeed-Anwendungen. Das FlexRay Protokoll ist eine Kombination von dem Byteflight™ (BMW1999) Protokoll und TTP/C. Byteflight™ wurde ursprünglich für passive Sicherheitssysteme wie Airbags entwickelt, in denen kurze Reaktionszeiten gefordert werden. Es ist jedoch nicht für aktive Kontrollsysteme geeignet da eine Fehlertoleranz nicht unterstützt wird. Für X-by-Wire Systeme ist die Fehlertoleranz eines Systems jedoch eine wichtige Voraussetzung.

Das FlexRay Konsortium zwischen BMW und DaimlerChrysler wurde gegründet um das Byteflight™ Protokoll von BMW dahingehend weiterzuentwickeln, um es für X-by-wire Systeme tauglich zu machen. FlexRay zielt mit seiner Datenrate mit bis zu 10Mbits/s auf Applikationen wie X-by-Wire oder Powertrain ab, die ein deterministisches und fehlertolerantes Kommunikationssystem benötigen und wird frei verfügbar sein.

6.6.3 Systemstruktur

Auf der Hardwareseite lassen sich bis 64 Knoten verbinden. Jeder Knoten besitzt einen Host mit einem Kommunikations-Controller. Ein oder zwei Bustreiber und Buswächters können an diese angeschlossen werden und alle Komponenten sind über eine Spannungsversorgung (Fahrzeuggestaltung) gekoppelt. Der Kommunikations-Controller ist das Herzstück von FlexRay. Er regelt den Datenfluss zwischen Host und Bus nach dem FlexRay Protokoll und erzeugt die globale Uhrzeit.

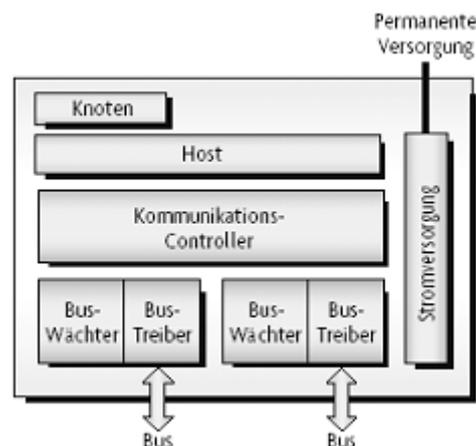


Abb. 6-25: Abbildung eines Knotens

Der Buswächter überwacht die Zeitslots und andere sicherheitsrelevante Funktionen. Er wird vom Controller gesteuert während der Bustreiber die Spannungsversorgung kontrolliert. Der Buswächter ist demnach nicht unabhängig vom Controller. Aus veröffentlichten Diagrammen ist jedoch ersichtlich, dass alle Komponenten eigene Uhrenoszillatoren besitzen, so dass der Controller und die Buswächter als eigenständige FCUs (Fault Containment Units) angesehen werden können. FlexRay unterstützt passiv Bus- und active Star- oder Multi-Star-Topologien, welche für sicherheitsrelevante Konfigurationen empfohlen werden. In beiden Fällen ist die Verdoppelung der Verbindungen optional. Diese aktiven Sterne enthalten jedoch keinen Buswächter.

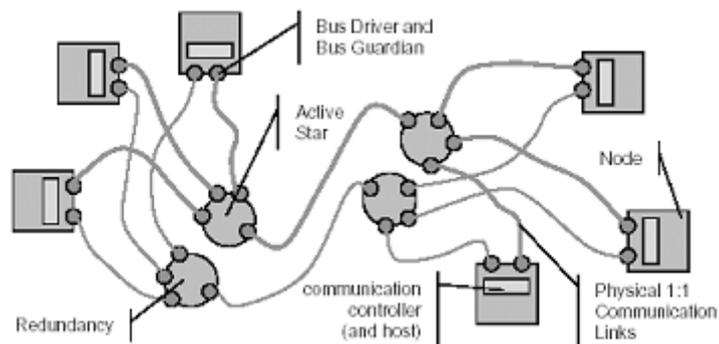


Abb. 6-26 : Netzkonfiguration mit aktiven Stern

Der Kommunikations-Controller kann entweder über zwei redundante Kanäle Daten senden und empfangen oder nur mit einem physikalischen Kanal verbunden sein (Abb. 6-20). Die Kommunikationsschnittstelle zwischen dem Kommunikations-Controller und dem Host Computer ist das Kommunikations- Network Interface (CNI). Jedes zeitgesteuerte Protokoll, wie FlexRay, benötigt dynamische Konfigurationsdaten, die vor Inbetriebnahme, in den Kommunikationscontroller geschrieben werden müssen. FlexRay minimiert diese Daten, indem viele der Parameter vorab im Protokoll festgesetzt sind. Es gibt drei Schnittstellentypen: Steuerparameter, Online-Steuerung und Online-Daten.

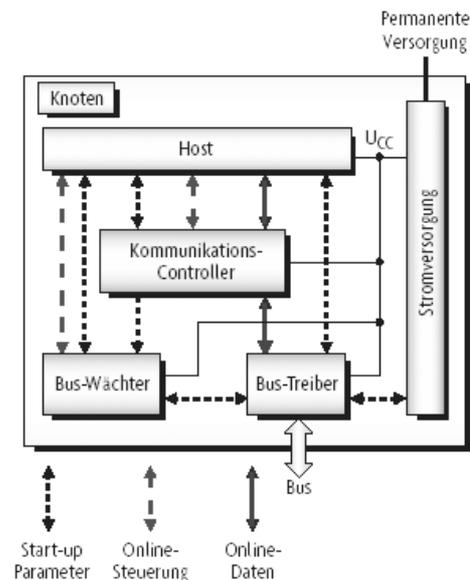


Abb. 6-27 : Schnittstellen zwischen den einzelnen Bausteinen [5]

6.6.4 Kommunikationsverfahren

Die Kommunikation läuft im Rahmen eines Kommunikationszyklus. FlexRay teilt die Zeit in zwei parallele wiederkehrende Intervalle, einen synchronen statischen Kanal für die zeitgesteuerten Nachrichten und einen asynchronen dynamischen Kanal für die Event-gesteuerten Nachrichten. Die Anteile dieser können flexibel festgelegt werden, wobei auch ein Teil leer sein kann. Somit gibt es drei mögliche Zykluskonfigurationen (statisch, gemischt (min. 2 statische Slots) und dynamisch).

Der Kommunikationszyklus beginnt mit einem SYNC-Symbol. Die Zeit-Slots werden durch die ID-Nummern identifiziert, die sowohl im statischen wie auch im dynamischen Teil benutzt werden.

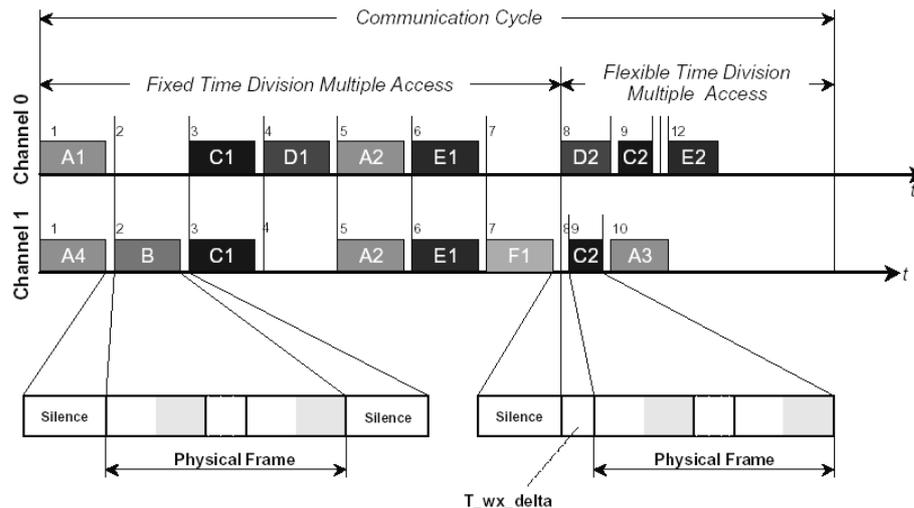


Abb. 6-28 : Kommunikationszyklus [10]

6.6.5 Statischer Teil

Die statischen Slots sind für Nachrichten hoher Priorität reserviert. Der Buszugriff erfolgt nach dem TDMA Verfahren (Time Division Multiple Access) für die Übertragung von zeitgesteuerten Nachrichten. Der Übertragungszyklus besteht aus einer frei konfigurierbaren Anzahl von Sende-Slots identischer Länge. Netzknoten die mit beiden Kanälen verbunden sind senden ihre Nachrichten synchron auf beiden Kanälen. Wird eine Nachricht nicht gesendet, verstreicht die entsprechende Zeit ungenutzt. Die Länge und der Inhalt der einzelnen Nachrichten können sich unterscheiden.

Jeder Teilnehmer erhält dieselbe Priorität, woraus sich der Vorteil ergibt, dass berechenbar und vorhersagbar ist, wann eine Nachricht gesendet oder empfangen werden kann. Dabei weiß jeder Busteilnehmer a priori die Intervalle, die er nutzen darf, wodurch eine Kollision automatisch ausgeschlossen wird.

Das Protokoll enthält eine fehlertolerante Uhrensynchronisation und schützt die Kommunikationskanäle vor „Babbling Idiots“ über einen Buswächter. Die Nachsynchronisation der Uhren ist Watchdog-überwacht. FlexRay bietet seit neuestem einen skalierbaren, erweiterten statischen Datenframe mit bis zu 254 Bytes.

6.6.6 Dynamischer Teil

Weniger wichtige „asynchrone“ Nachrichten werden im dynamische Teil gemäß der Byteflight™ Spezifikation übertragen. Seine Slots sind variabel und es kann Unterschiede in den Zeit-Slots auf den zwei Kanälen geben.

Diese Flexibilität erhöht die Übertragungsrates von FlexRay beträchtlich. Die Slot Zähler werden synchron hochgezählt und der Buszugriff folgt über das FTDMA Verfahren (Flexible Time Division Multiple Access), bei der die Einzelkanäle innerhalb einer verfügbaren Bandbreite auf einen Teilbereich dieser Bandbreite zugreifen. Bei einer Datenübertragung werden für die Dauer der Übertragung die

Slotzähler auf dem aktuellen Wert gestoppt. Kollisionsfreie Zugriffsmechanismen steuern den Sendevorgang.

6.6.7 Nachrichtenformat

Das Nachrichtenformat ist im statischen wie im dynamischen Teil identisch.

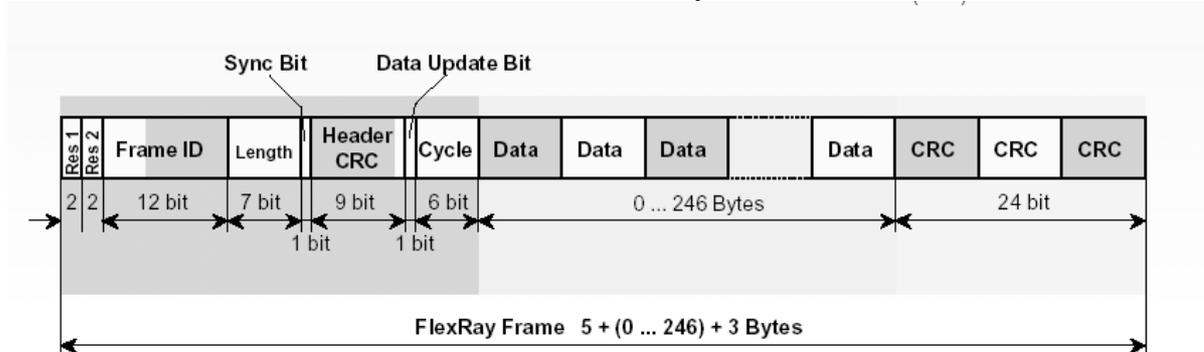


Abb. 6-29 : Frame Format mit Cycle Counter[10]

Frame ID:	Identifizier, 10 Bit, Wertebereich: (110 ... 102310), definiert die Slotposition im statischen Teil und die Priorität im dynamischen Teil. Ein kleinerer Identifizier bestimmt eine höhere Priorität. ID = 0 ist für das SYNC-Symbol reserviert. Ein Identifizier darf in einem Netzwerk nur einmal verwendet werden. Jeder Knoten kann einen oder mehrere Identifizier – sowohl im statischen als auch im dynamischen Teil – verwenden.
SYNC:	SYNC-Feld, 1 Bit. Dieses Bit zeigt an, ob die Nachricht zur Uhrensynchronisierung verwendet wird und ob das erste Datenbyte den Zykluszähler enthält. (SYNC = "1": Botschaft mit Frame-Counter und Uhrensynchronisation, SYNC = "0": Botschaft ohne Frame-Counter)
LENGTH:	Längen-Feld, 4 Bit, Anzahl der Datenbytes (010 ... 1210). Ein Wert größer 12 wird als LEN=12 interpretiert. Bei Nutzung des Zykluszählers (SYNC=1) wird bei einem Wert größer 11 LEN=11 gesetzt.
CYCLE:	Das CYCLE-Feld kann als Zykluszähler oder als erstes Datenbyte genutzt werden. Der Zykluszähler wird am Anfang eines jeden Kommunikationszyklus in allen Kommunikationscontrollern synchron hochgezählt.
D0 ... D246:	Daten Bytes, 0 – 246 Bytes
CRC:	24 Bit Cyclic Redundancy Check.

Tabelle 6-1 : Erläuterung zum Frame Format

6.6.8 Uhrensynchronisation

Um den richtigen Sendepunkt ermitteln zu können, ist eine verteilte Zeitbasis mit regelmäßiger Uhrensynchronisation durch die Kommunikations-Controller der einzelnen Knoten notwendig. Um die Uhrensynchronisation zu garantieren, werden hierfür jedoch nur Nachrichten von Netzknoten verwendet die an beiden Kanälen angeschlossen sind. Zusätzlich werden nur ausgesuchte Nachrichten aus dem statischen Teil verwendet. Die Uhrensynchronisation ist aktiv, sobald mindestens zwei Knoten aktiv sind.

6.6.9 Sicherheit und Fehlertoleranz

In den veröffentlichten FlexRay Dokumenten ist keine Fehler-Hypothese oder Never-give-up Strategy spezifiziert. Im Falle eines physikalischen Fehlers eines Knotens ist durch die fehlende Nachrichtenkonsistenz, die dafür Sorgen würde dass alle Knoten den Systemstatus kennen, kein Vorgehensweise spezifiziert worden. Der Buswächter befindet sich schließlich in dem fehlerhaften Knoten. FlexRay benutzt den Welch-Lyn Algorithmus, bei dem Zeitunterschiede wahrscheinlich in der Art und Weise von TTA detektiert werden. Da das FlexRay Netzwerk nicht über verteilte Membership-Informationen im Protokoll verfügt, erlangen fehlerhafte Knoten eine übermäßige Wichtigkeit in dem Algorithmus der Uhrensynchronisation. Es ist nicht bekannt wie das FlexRay-Protokoll sicherheitskritische Ereignisse behandelt. Eine mögliche Lösung dieses Problems, wäre die Einführung eines Membership Protokolls auf Applikationsebene, die jedoch individuell vom Anwender programmiert werden müsste und somit eine umfangreichere HostSoftware zur Folge hätte.

6.6.10 Zusammensetzbarkeit

Ein wichtiger Aspekt ist die Zusammensetzbarkeit verschiedener Knoten. Durch das Byteflight™ Protokoll ist es unmöglich die temporären Eigenschaften von Verbindungskomponenten unabhängig von dem globalen Inhalt der Applikation zu definieren. Zusammensetzbarkeit ist demnach nicht gegeben.

Literatur

- [1] FlexRay Overview, Advanced Automotive Communication System
- [2] Klasche, G: TTP und FlexRay: Wann kommt die Einigung? Elektronik Automotive September 2001
- [3] Kopetz, H.: A Comparison of TTP/C and FlexRay; TU Wien Research Report 2001/10
- [4] FlexRay-Homepage, www.flexray-group.com
- [5] FlexRay für verteilte Anwendungen im Fahrzeug, Elektronik Automotive, Mai 2001

- [6] Dr. Schedl, A., Lohrmann, P. :Anforderungen an ein zukünftiges Bussystem für fehlertolerante Anwendungen aus Sicht Kfz-Hersteller, VDI 17.10.2000
- [7] Wengenmayr, R. : FlexRay – ein neuer Datenbus für Autos entsteht, Technische Rundschau Nr.18 2001
- [8] Weiter verbessert, Design&Elektronik 03/2002
- [9] Fuchs, E. : FlexRay™ Requirements Specification, Version 1.9.7, 7.Sept. 2001
- [10] Bogenberger, F. : FlexRay, and where it stands today (02AE-41), SAE World Congress März 2002