

HANSER

Mathias Rausch

FlexRay

Grundlagen, Funktionsweise, Anwendung

ISBN-10: 3-446-41249-2

ISBN-13: 978-3-446-41249-1

Vorwort

Weitere Informationen oder Bestellungen unter
<http://www.hanser.de/978-3-446-41249-1>
sowie im Buchhandel

Vorwort

Nach sieben Jahren intensiver Entwicklungsarbeit, über einem Dutzend gehaltener Seminare und Schulungen, sowie hunderten von Fragen von Kunden und Kollegen wurde es einfach Zeit, das Wissen über und die Erfahrungen mit FlexRay in einem Buch zusammenzufassen.

Dieses Buch soll kein Ersatz für die Protokollspezifikation sein noch eine kommentierte Ausgabe davon. Vielmehr soll das Buch die Lücke zwischen abstrakten Beschreibungen sowie der Darstellung von Teilaspekten in einer Reihe von Veröffentlichungen und der detaillierten Protokollbeschreibung in der Spezifikation schließen. So wurde bewusst auf die Darstellung und Erläuterung der SDL-Diagramme der Spezifikation verzichtet, um sich auf das Zusammenspiel der einzelnen Teile konzentrieren zu können. An einigen Stellen geht dieses Buch aber auch über die Spezifikation hinaus, wenn z. B. die Regeln für die Konfiguration eines Netzwerkes detailliert erklärt und die Zusammenhänge dargestellt werden.

Die Leserin, der Leser, sollen mit diesem Buch befähigt werden, das Protokoll zu verstehen, sodass sie selbst ein Gefühl dafür bekommen, an welchen Stellen bei ihren Anwendungen Optimierungspotenzial vorhanden ist und wann aus Gründen von Sicherheit und Verfügbarkeit auf ein paar Prozent Bandbreite verzichtet werden sollte. Das Buch wird sicher nicht alle Fragen beantworten; dazu ist das Thema zu vielschichtig und die Thematik zu komplex.

Die Leserin möge es mir nachsehen, dass ich ausschließlich die männliche Anrede verwende, wenn ich den Anwender anspreche, aber ich wollte die Sätze durch die Verwendung jeweils beider Formen in einem Satz nicht unnötig verkomplizieren.

Das Protokoll selbst wurde in einem internationalen Konsortium entwickelt, sodass wie allgemein üblich sowohl die Kommunikations- als auch die Dokumentationsprache Englisch ist. Somit sind die englischsprachigen Begriffe eingeführt und nicht immer existiert auch eine deutsche Entsprechung.

Danksagung

An dieser Stelle möchte ich mich bei denen bedanken, die direkt oder indirekt zu diesem Buch beigetragen haben. Dies sind die Vertreter im FlexRay-Konsortium. Besonders eng habe ich mit Gregor Pokorny, Bernd Müller, Jörn Ungermann, Matthias Kühlewein, Tom Forest, Arnie Millsap, Bernd Elend und Thomas Wagner zusammengearbeitet. Gemeinsam haben wir Protokollmechanismen entwickelt und viele Details untersucht und diskutiert.

In meiner Firma gab es eine enge Zusammenarbeit bei der Entwicklung des Protokolls, dessen Analyse und dem Aufspüren und Beseitigen von möglichen Schwachstellen. Insbesondere möchte ich Florian Bogenberger, Thomas Würz, Christopher Temple, Dirk Möller, Christian Steffen, Thomas Lüdeke, Markus Brenner, Hans-Christian von der Wense und Leonhard Link für die gute Zusammenarbeit danken. Doch nicht nur den Kollegen möchte ich danken,

sondern auch den Studenten Matthias Fink, Markus Regner und Christian Steur von der Fachhochschule in Kempten, für ihre wertvollen Arbeiten zum Thema FlexRay.

Ein besonderer Dank gilt den Reviewern Matthias Kühlewein, Bernd Elend und Jürgen Jagst für zahlreiche Anregungen und das Finden einer Vielzahl von Fehlern im Manuskript sowie Herrn Herzberg vom Hanser-Verlag für die gute Zusammenarbeit.

Meiner Familie möchte ich danken und dabei besonders meiner Frau Eva für die Hilfe und das Verständnis.

München, September 2007

Mathias Rausch

HANSER

Mathias Rausch

FlexRay

Grundlagen, Funktionsweise, Anwendung

ISBN-10: 3-446-41249-2

ISBN-13: 978-3-446-41249-1

Leseprobe

Weitere Informationen oder Bestellungen unter
<http://www.hanser.de/978-3-446-41249-1>
sowie im Buchhandel

3 Prinzipielle Funktionsweise des Protokolls

In diesem Kapitel werden schrittweise alle Funktionen des Protokolls vorgestellt, sodass der Leser einen ersten Überblick über die Funktionen und deren Zusammenspiel bekommt. Im Kapitel 4 werden die Funktionen und Mechanismen detaillierter beschrieben.

3.1 Aufbau eines Kommunikationsknotens

Ein Kommunikationsnetzwerk setzt sich aus mehreren Kommunikationsteilnehmern und einem oder mehreren Kommunikationskanälen (engl.: channels) zusammen. Die Kommunikationsteilnehmer werden auch als Kommunikationsknoten oder kurz nur Knoten (engl.: nodes) bezeichnet.

Über Kommunikationskanäle werden die Informationen elektrisch oder optisch (z. B. über Lichtwellenleiter) übertragen. FlexRay kann sowohl mit elektrischen als auch optischen Medien arbeiten, wobei bisher aber nur der elektrische Physical Layer spezifiziert wurde [EPL06].

Ein FlexRay-Knoten besteht aus einem Mikrocontroller, kurz Host genannt, dem eigentlichen FlexRay-Controller, auch als Kommunikationscontroller (engl.: communication controller (CC)) bezeichnet, und einem oder zwei Bustreibern (engl.: bus driver). Die Bustreiber stellen die physikalische Verbindung zum Kommunikationskanal her. Der FlexRay-Controller realisiert das (logische) FlexRay-Protokoll. Auf dem Host läuft das eigentliche Anwendungsprogramm, das Informationen von anderen Knoten empfängt und an diese sendet. Bevor Kommunikation stattfinden kann, muss der Host den FlexRay-Controller konfigurieren.

Es gibt zwei Ausprägungen von FlexRay-Knoten. Beim stand-alone FlexRay-Controller (Bild 3.1) ist der FlexRay-Controller als eigenständiger Schaltkreis ausgeführt und über einen Adress- und Datenbus oder eine serielle Schnittstelle mit dem Host verbunden. Beim integrierten Controller (Bild 3.2) befindet sich der FlexRay-Controller als Peripheriemodul auf dem Hostcontroller-Schaltkreis im gleichen Gehäuse. Dies vereinfacht den Datenaustausch, da beide Controller Adress- und Datenbus miteinander teilen. Je nach Ausführung können Host und FlexRay-Controller je einen eigenen Speicher besitzen oder beide teilen sich den gemeinsamen Hauptspeicher. Der stand-alone FlexRay-Controller lässt sich flexibler einsetzen, wogegen der integrierte FlexRay-Controller Kostenvorteile mit sich bringt und einen schnelleren Austausch der Daten erlaubt. Ein Mikrocontroller kann auch mehr als einen FlexRay-Controller integriert haben.

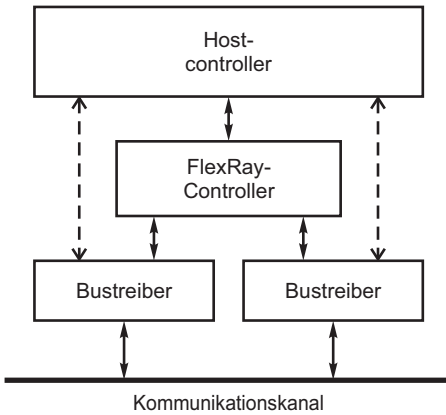


Bild 3.1: Prinzipieller Aufbau eines stand-alone FlexRay-Knotens

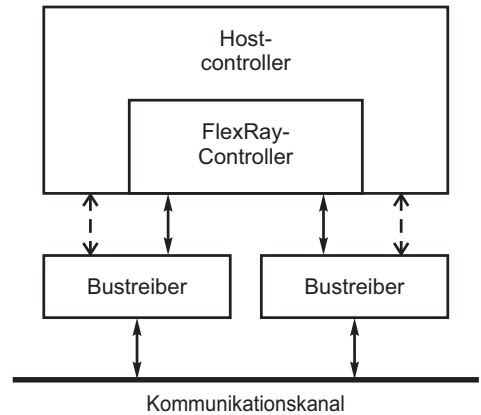


Bild 3.2: Prinzipieller Aufbau eines integrierten FlexRay-Knotens

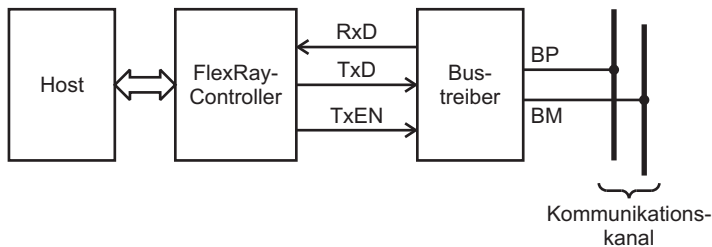


Bild 3.3: Schnittstellen eines FlexRay-Controllers

Der FlexRay-Controller besitzt Schnittstellen zum Host und zu den Bustreibern (Bild 3.3). Die Schnittstelle zwischen Host und FlexRay-Controller ist implementationsabhängig. Die Schnittstelle zum Bustreiber ist standardisiert und besteht aus drei Leitungen. Über die RxD-Leitung (Receive Data) überträgt der Bustreiber die empfangenen Signale an den FlexRay-Controller. Über die TxD-Leitung (Transmit Data) überträgt der FlexRay-Controller die Datensequenz, die der Bustreiber auf den Kommunikationskanal sendet. Mittels der Low-aktiven TxEN-Leitung (Transmit Data Enable Not) signalisiert der FlexRay-Controller, wann die auf der TxD-Leitung anliegenden Daten gesendet werden sollen.

Intern besteht ein FlexRay-Controller aus einem Controller Host Interface (CHI) und einer Protocol Engine (PE) (siehe Bild 3.4). Das Controller Host Interface bildet die Schnittstelle zum Host und verwaltet die Puffer für die Daten und die Konfiguration. Die Protocol Engine ist für den Empfang und das Senden der Daten verantwortlich. Transmitter und Receiver als Teil der Protocol Engine codieren bzw. decodieren die Signale, die der Bustreiber empfängt bzw. sendet. Eine Zeitsteuerung (Time Control Unit (TCU)) realisiert die Zeitbasis, die Grundlage des zeitgesteuerten Verhaltens des FlexRay-Protokolls ist.

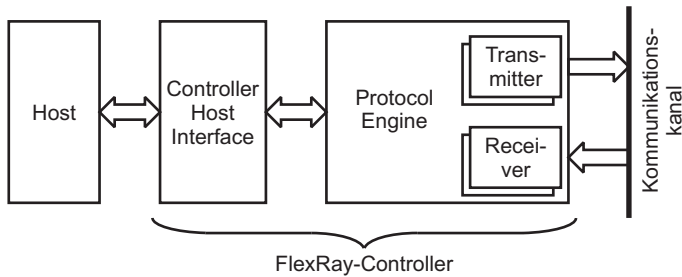


Bild 3.4: Aufbau eines FlexRay-Controllers

3.2 Topologien

Die physikalische Anordnung der Kommunikationsknoten wird als Topologie bezeichnet. Daneben gibt es noch eine logische Anordnung, die nicht notwendigerweise mit der physikalischen Anordnung übereinstimmen muss.

Im Bild 3.5 sind Knoten dargestellt, die über ein Netzwerk miteinander verbunden sind. Dieses Netzwerk kann recht unterschiedlich gestaltet sein. In diesem Buch wird als Netzwerk, genauer als Kommunikationsnetzwerk, die Gesamtheit der Kommunikationsstruktur in einem Fahrzeug bezeichnet. Ein Netzwerk kann dabei aus mehreren Teilnetzwerken, auch als Cluster bezeichnet, aufgebaut sein. Ein Cluster ist ein Teilnetzwerk, das eine Einheit bildet und unabhängig von anderen Clustern arbeiten kann.

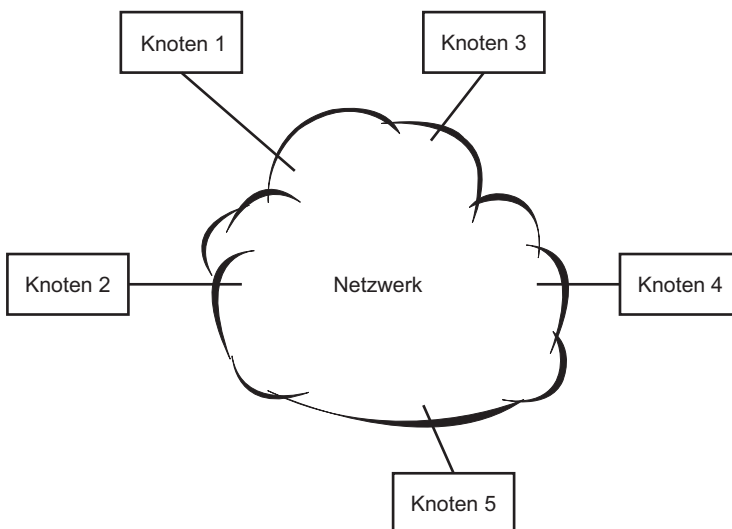


Bild 3.5: Verbindung von Knoten zu einem Netzwerk

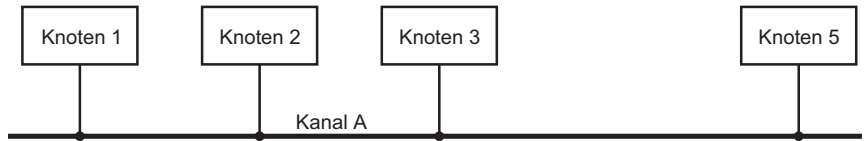


Bild 3.6: Beispiel für eine einkanalige Bustopologie

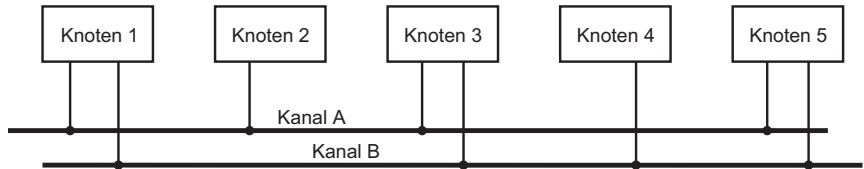


Bild 3.7: Beispiel für eine zweikanalige Bustopologie

Die Kommunikation innerhalb heutiger Autos erfolgt beispielsweise über mehrere CAN-Busse, mehrere LIN-Busse, einen Multimediabus (z. B. MOST) und eventuell über ein oder mehrere FlexRay-Systeme. Der Begriff Bus wird in diesem Zusammenhang meist als Synonym für Cluster verwendet und weniger als Topologieeigenschaft. Jeder CAN-, LIN- und MOST-Bus stellt einen Cluster dar. Die einzelnen Cluster sind über Gateways miteinander verbunden und bilden das Kommunikationsnetzwerk des Autos. Das FlexRay-System würde natürlich auch einen Cluster bilden.²

Es gibt verschiedene Grundtopologien, um ein Cluster zu realisieren: Bus-, Stern-, Ring- und Baumtopologie sowie andere. FlexRay unterstützt Bus- und Sterntopologien.

FlexRay wurde als zweikanaliges System entwickelt, sodass sowohl ein- als auch zweikanalige Systeme realisiert werden können. Bild 3.6 zeigt eine einkanalige Bustopologie, wogegen Bild 3.7 eine zweikanalige Bustopologie zeigt.

Sterntopologien können mittels aktiver oder mittels passiver Sternkoppler realisiert werden. Ein aktiver Sternkoppler ist ein elektronisches Gerät, das die elektrischen Signale verstärkt und verteilt. In Computernetzwerken wird ein Gerät mit dieser Funktionalität als Hub bezeichnet. Im Gegensatz dazu besitzt ein passiver Stern keine aktiven Elemente, sodass die elektrischen Signale auch nicht verstärkt werden. Ein passiver Stern ist im Wesentlichen nur eine elektrische Verbindung ohne weitere Bauelemente.

Bild 3.8 zeigt ein Beispiel für eine einkanalige Sterntopologie, wogegen Bild 3.9 ein zweikanaliges FlexRay-System in Sterntopologie zeigt. Die beiden Beispiele für die einkanalige Bustopologie (Bild 3.6) und die einkanalige Sterntopologie (Bild 3.8) haben eine unterschiedliche

² Der Begriff *FlexRay-Bus* wurde hier bewusst vermieden. FlexRay wurde als zweikanaliges System entwickelt, sodass ein zweikanaliger FlexRay-Cluster aus zwei Bussen besteht. Die herkömmliche Zählweise, dass die Anzahl der Busse gleich der Anzahl der Cluster ist, gilt hier nicht mehr.

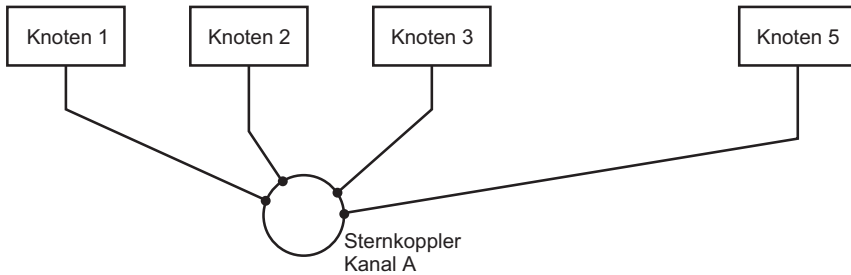


Bild 3.8: Beispiel für eine einkanalige Sterntopologie

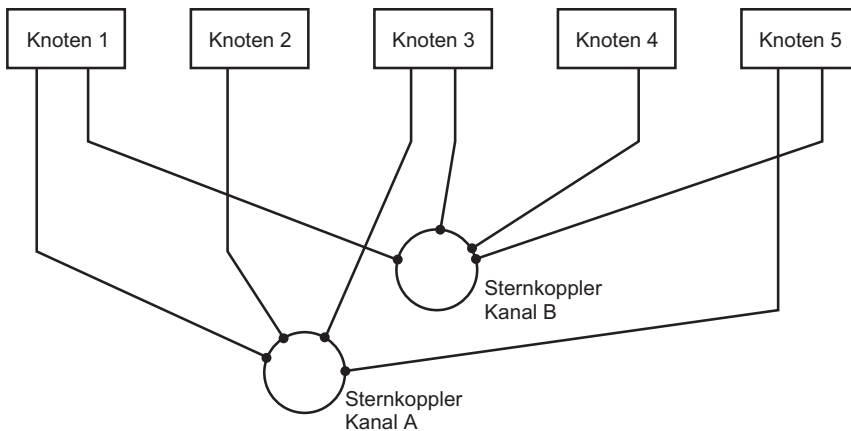


Bild 3.9: Beispiel für eine zweikanalige Sterntopologie

physikalische Topologie, aber die gleiche logische Topologie. Dies gilt auch für die beiden zweikanaligen Beispiele für Bustopologie (Bild 3.7) und Sterntopologie (Bild 3.9), die ebenfalls die gleiche logische Topologie haben.

Aktive Sterne können auch kaskadiert werden. Dabei wird ein aktiver Stern mit einem anderen verbunden. Vorteil dieser Topologie ist die größere Anzahl von Knoten, die angeschlossen werden können. FlexRay unterstützt maximal zwei kaskadierte Sterne je Kanal. Das heißt, dass sich im Signalpfad von einem beliebigen Knoten zu jedem anderen beliebigen Knoten maximal zwei aktive Sternkoppler befinden dürfen (Bild 3.10). Dabei ist es nicht notwendig, dass beide Kanäle kaskadierte Sternkoppler besitzen. So könnte ein Kanal mittels kaskadierter Sternkoppler realisiert werden, wogegen der andere Kanal nur einen oder sogar keinen Sternkoppler besitzt (siehe auch Bild 3.12).

Passive Sterne können auch als kurzer Bus mit langen Stichleitungen gesehen werden. Vom Protokollstandpunkt aus gesehen sind Bustopologie und eine Topologie mit passivem Stern(en) identisch. Wesentliche Unterschiede ergeben sich aber bei den elektrischen Eigenschaften des Physical Layers. Nähere Ausführungen sind dazu im Kapitel 5 zu finden.

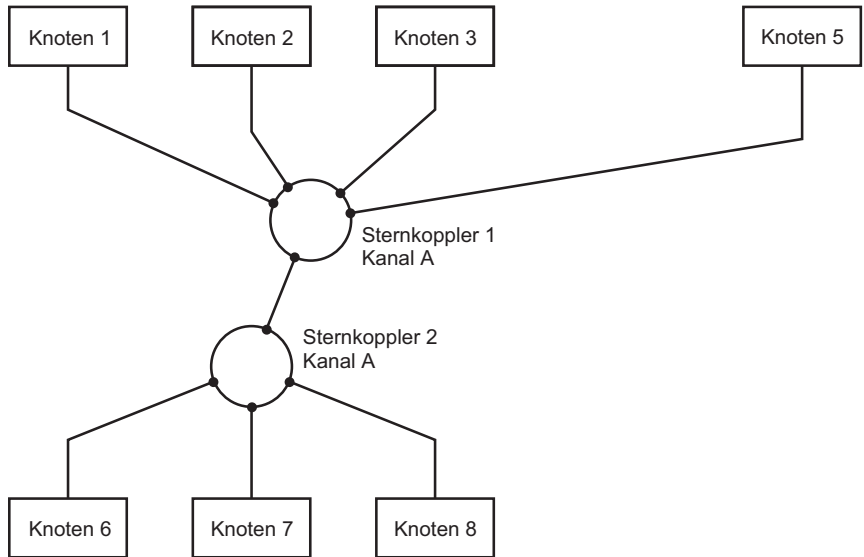


Bild 3.10: Beispiel für eine einkanalige Topologie mit kaskadierten Sternkopplern

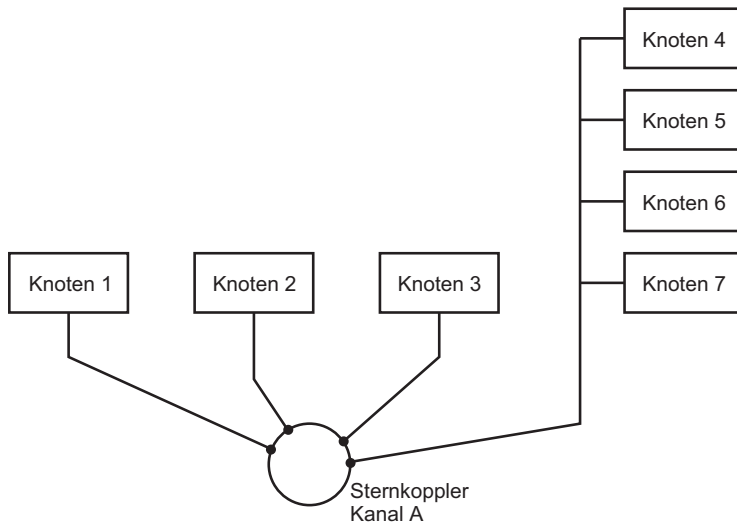


Bild 3.11: Beispiel für die gemischte Verwendung von Bus- und Sterntopologie

FlexRay ist nicht auf „reine“ Topologien beschränkt, sondern die bisher beschriebenen Topologien können auch miteinander gemischt werden. So kann bei einer Sterntopologie an einem oder mehreren Zweigen eines aktiven Sternkopplers ein Bus oder ein passiver Stern angeschlossen werden. Dies erhöht die Anzahl der Knoten, die an einen Kanal angeschlossen werden können (Bild 3.11).

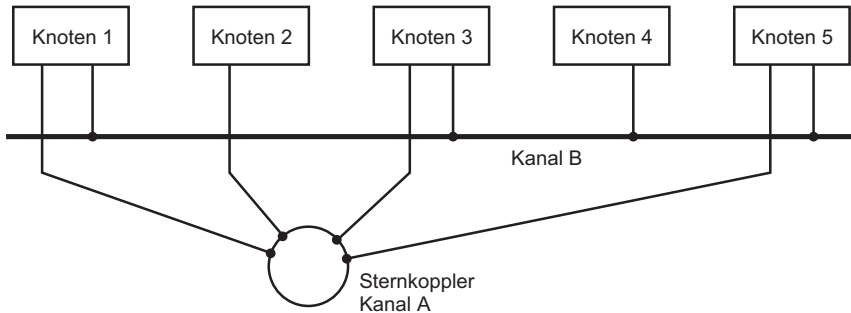


Bild 3.12: Beispiel für die Verwendung unterschiedlicher Topologien auf Kanal A und B

In einem FlexRay-Cluster ist es nicht notwendig, dass Kanal A und Kanal B in der gleichen Topologie ausgeführt sind. So ist es z. B. möglich, Kanal A als Sterntopologie mit aktivem Sternkoppler und Kanal B als Bustopologie auszuführen (Bild 3.12). Welche Topologie letztendlich zur Anwendung kommt, hängt von einer Reihe von Faktoren ab, die in Kapitel 5 näher erörtert werden.

3.3 Das Zugriffsverfahren

Bei einem Kommunikationssystem nutzen mehrere Teilnehmer zum Austausch von Nachrichten einen oder mehrere Kommunikationskanäle. Der Kommunikationskanal ist dabei eine Ressource, deren Verfügbarkeit Voraussetzung für die Kommunikation ist. Zu einem Zeitpunkt kann immer nur ein Teilnehmer diese Ressource nutzen. Das Zugriffsverfahren, auch als Zugriffssteuerung bezeichnet (engl.: media access control), beschreibt die Regeln zur Nutzung des bzw. der Kommunikationskanäle. Nach dem OSI-Referenzmodell ist die Zugriffsteuerung eine der wesentlichen Funktionen der Schicht 2 (Sicherheitsschicht).

3.3.1 Zugriffsverfahren im Überblick

Die Zugriffsverfahren können in Verfahren mit kontrolliertem (deterministischem) Zugriff und konkurrierendem (zufälligem) Zugriff unterteilt werden. Verfahren mit kontrolliertem Zugriff unterscheiden sich darin, ob der Zugriff zentral gesteuert oder ob der Zugriff dezentral verwaltet wird. Ein zentral gesteuerter Zugriff kann z. B. durch einen Master realisiert werden. Ein dezentral verwalteter Zugriff wird durch Absprache der Teilnehmer untereinander realisiert, z. B. durch die Weitergabe eines Token. Bei Verfahren mit konkurrierendem Zugriff werden kollisionsfreie und nicht kollisionsfreie Methoden unterschieden (Bild 3.13).

Das Zugriffsverfahren ist eine der Haupteigenschaften eines Kommunikationsprotokolls und dient oftmals zu dessen Klassifizierung. Nachfolgend werden die unterschiedlichen Zugriffsverfahren anhand konkreter Realisierungen charakterisiert.