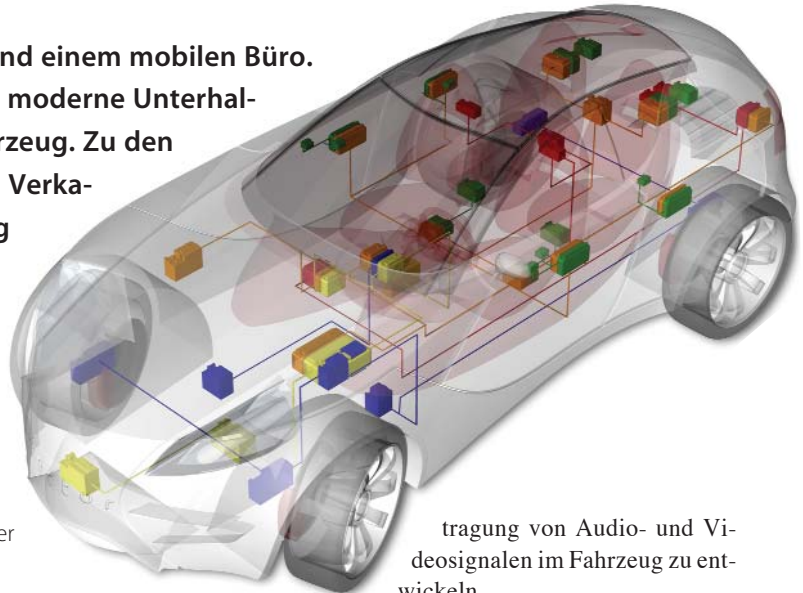


Serielle Bussysteme im Auto

Teil 5: MOST für die Übertragung von Multimediadaten

Ein Auto der Premiumklasse ähnelt zunehmend einem mobilen Büro. Auf Wunsch des Kunden drängen immer mehr moderne Unterhaltungs- und Informationsmedien in das Kraftfahrzeug. Zu den wichtigsten Herausforderungen gehört dabei, den Verkabelungsaufwand gering zu halten und gleichzeitig den gestiegenen Anforderungen an den Funktionsumfang eines Infotainmentsystems im Fahrzeug gerecht zu werden. In circa 50 Modellreihen kommt daher bereits das Bussystem MOST (Media Oriented System Transport) zur Übertragung von Audio- und Videosignalen zum Einsatz.

Von Eugen Mayer



Übertragung von Audio- und Videosignalen im Fahrzeug zu entwickeln.

War früher das Autoradio einziges Infotainment-Gerät, kamen im Laufe der Zeit CD- und MP3-Player, Navigationsgeräte und schließlich auch Bildschirme für die Wiedergabe von Video- und DVD-Filmen hinzu. Darüber hinaus lassen Bluetooth-Freisprecheinrichtungen mit integriertem Mikrofon und iPod-Steuerung das Cockpit allmählich zum Multimedia-Center werden, über das sich alle Playlisten und Verzeichnisse eines digitalen MP3-Players direkt auf dem Fahrzeugdisplay anzeigen und auch starten lassen.

Der ohnehin bereits umfangreiche Verkabelungsaufwand nimmt durch die weiter ansteigende Vernetzung der immer leistungsfähigeren Infotainmentgeräte kaum mehr handhabbare Ausmaße an. Glücklicherweise erkannten einige Kfz-Hersteller schon früh, welche Vorteile die Busvernetzung auch für diesen Bereich zu bieten hat. Mitte der 1990er Jahre begannen BMW und Daimler auf der Basis des von Matsushita und Philips entwickelten D2B-Bus (Digital Data Bus) eine einheitliche Kommunikationstechnologie zur seriellen Über-

Die MOST Cooperation

1998 gründeten BMW, Daimler, Harman/Becker und SMSC (vormals OASIS Silicon Systems) die MOST Cooperation [1]. Inzwischen hat sich MOST als De-facto-Standard für die Übertragung von Multimediadaten im Fahrzeug etabliert; die MOST Cooperation umfasst 15 internationale Fahrzeug- und mehr als 70 Gerätehersteller. Die MOST-Spezifikation liegt seit Oktober 2006 in der Version 2.5 vor. Sie gliedert sich in die Bereiche Application, Network und Hardware.

Der Bereich Application beschreibt ein logisches Gerätemodell zur transparenten Modellierung und Steuerung von verteilten Infotainment-Systemen und basiert in erster Linie auf Methoden der Objektorientierung. Weiterhin definiert er ein hierarchisches Kommunikationsmodell sowie Dienste zur Verwaltung eines Infotainment-Systems.

Die Network Section beschreibt den „MOST Network Interface Controller“ und seine Dienste, das Netzwerkmanagement sowie die Handhabung des Datentransports in einem MOST-System. Die Hardware Section setzt sich mit der Hardware-Struktur eines MOST-Gerätes auseinander.

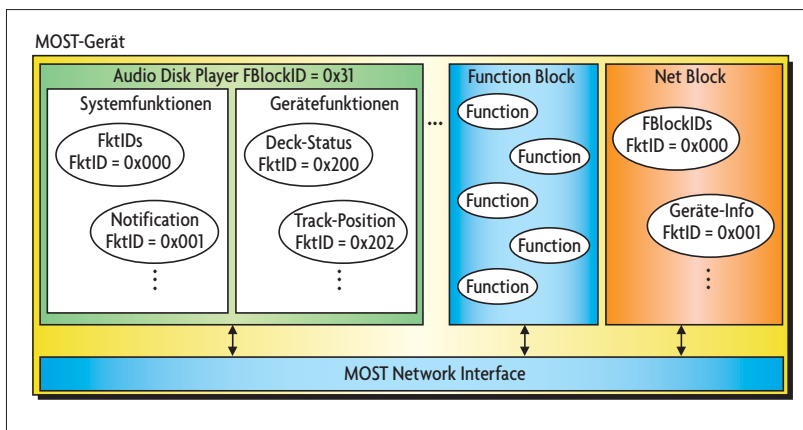


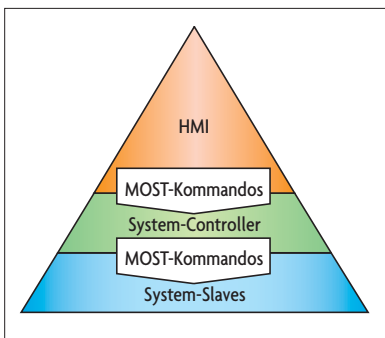
Bild 1. Struktur eines MOST-Gerätes, das unter anderem den Function Block „Audio Disk Player“ beherbergt. Zur Systemverwaltung sind für jedes MOST-Gerät der Net Block, für jeden Function Block Systemfunktionen obligatorisch.

■ Modellierung von Funktionen

Ein MOST-Gerät gliedert sich in eine Funktions- und in eine Netzwerkebene (MOST Network Interface). Auf der Funktionsebene sind die Infotainment-Funktionen als Funktionsblöcke (Function Blocks) untergebracht. Jeder Funktionsblock wie etwa der „Audio Disk Player“ stellt dem MOST-Netzwerk einen dedizierten Satz von Funktionen (z.B. Track Position) zur Verfügung, auf die mittels Operation Types (z.B. Set zum Setzen eines Tracks, oder SetGet zum Setzen und Lesen eines Tracks) zugegriffen werden können (Bild 1).

Sowohl den Funktionsblöcken als auch den von einem Funktionsblock bereitgestellten Funktionen sind Adressen (FBlockID, FktID) zugeordnet. Man kann sie, ebenso wie die Kennungen der Operation Types, dem Function Catalog entnehmen. So hat der FBlock „Audio Disk Player“ den FBlockID=0x31 – die Funktion „Track Position“ den FktID=0x202.

Durch die Trennung von Funktion und Netzwerk und mit Hilfe der Funktionsmodellierung lässt sich ein Kom-



! Bild 2. Hierarchie eines MOST-Systems.

munikationsmodell realisieren, welches völlig unabhängig von physikalischen MOST-Geräten ist. Folglich spielt es keine Rolle, auf welchen MOST-Geräten man einen Funktionsblock unterbringt.

■ Hierarchisches Kommunikationsmodell

MOST-Systemen liegt eine dreistufige, hierarchische Steuerungsphilosophie nach dem Master-Slave-Prinzip zugrunde (Bild 2). Auf der obersten

Hierarchieebene ist die HMI (Human Machine Interface) angesiedelt, ein exponierter Controller, der dem Anwender den gesamten Funktionsumfang zur Verfügung stellt. Auf der mittleren Hierarchieebene befinden sich die üblichen Controller. Sie decken einen Teil der Systemfunktionen ab und stellen ihr partielles Systemwissen der HMI als System-Master zur Verfügung.

Die unterste Hierarchieebene bilden die System-Slaves, deren Funktionen jeweils von einem oder mehreren Controllern nutzbar sind. Sie sind mit keinerlei Systemwissen ausgestattet, was die Flexibilität hinsichtlich der Konfiguration wesentlich erhöht. System-Slaves lassen sich einem MOST-System einfach hinzufügen oder entfernen. Zur Steuerkommunikation dienen MOST-Kommandos. Sie umfassen im Kern den FBlockID, den FktID, den Operation Type und bis zu 65 535 Nutzbyte.

■ Systemverwaltung

Die Application Section definiert auch übergeordnete Funktionsblöcke und Funktionen zur Systemverwaltung. Zu den Systemfunktionen zählt unter anderem die Funktion FktIDs (FktID=0x000), mit der man die von einem Funktionsblock unterstützten Funktionen abfragt. Die Systemfunktion Notification (FktID=0x001) erlaubt das Erstellen der Notification-Matrix für einen Funktionsblock. Aus der Notification-Matrix geht hervor, welches MOST-Gerät zu benachrichtigen ist, wenn sich eine bestimmte Eigenschaft eines Funktionsblocks verändert hat. Dieser Mechanismus verhindert ein unnötiges Ansteigen der Buslast im MOST-System.

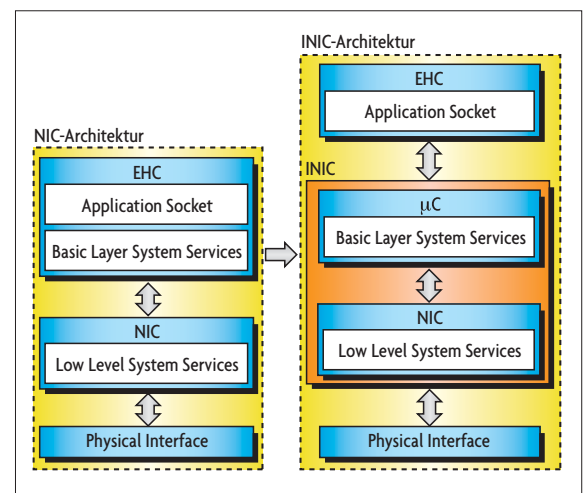
Zur Abfrage seiner Funktionsblöcke und Adressen hat jedes MOST-Gerät den (System-)Funktionsblock Net Block mit der FBlockID=0x01. Mittels der Funktion FBlockIDs (FktID=0x000) können die auf einem MOST-Gerät implementierten Funktionsblöcke in Erfahrung gebracht werden. Über die FktIDs 0x002, 0x003 und 0x004 lassen sich die physikalische und die logische Adresse sowie die Gruppenadresse eines MOST-Gerätes ermitteln.

Eine wichtige Aufgabe bei der Verwaltung eines MOST-Systems hat der Network Master. Er ist für den Systemstart und die Verwaltung der Central Registry verantwortlich. Diese umfasst die logischen Adressen der in einem MOST-System implementierten MOST-Geräte und die Adressen der darauf untergebrachten Funktionsblöcke.

■ „MOST Network Interface“

Das „MOST Network Interface“ (Bild 3) sorgt dafür, dass die auf unterschiedlichen MOST-Geräten untergebrachten Funktionsblöcke in der Lage sind, miteinander zu kommunizieren. Dabei erbringen die „MOST System Services“ („Low Level System“ und „MOST Network Services“) die Kommunikationsfunktionen, die für den Transport aller multimedienrelevanten Daten (zeitkontinuierliche Bitströme, Paketdaten und Steuerdaten) notwendig sind. Die „Low Level System Services“ (Schicht-2-Dienste) sind in Hardware (Network Interface Controller; NIC) implementiert und setzen auf dem Physical Layer auf.

Die „MOST Network Services“, die den Transport Layer in Form von „Basic Layer System Services“ und das höhere Management in Form eines Application Sockets umfassen, sind auf einem externen Host Controller (EHC) untergebracht und steuern den NIC. Dabei muss sichergestellt sein, dass der EHC auch die zeitkritischsten Teile des Network Interface bedienen



! Bild 3. Unterschied zwischen der NIC- und der INIC-Architektur eines MOST-Gerätes.

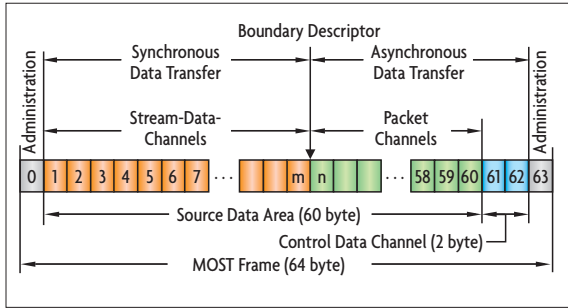


Bild 4. Aufbau des MOST-Frames: Im Administrations-Byte 0 werden Synchronisationsinformationen und der Boundary Descriptor, im Administrations-Byte 63 werden Status-Bits und ein Paritäts-Bit zur Sicherung des MOST-Frames übertragen.

kann. Durch die Weiterentwicklungen, von MOST25 über MOST50 zu MOST150, stößt diese Architektur inzwischen an ihre Grenzen.

In Neuentwicklungen löst der INIC (Intelligent Network Interface Controller) den NIC ab. Während der INIC die Abwicklung der zeitkritischen Teile des Netzwerktreibers vom EHC übernimmt, läuft auf dem EHC nur noch ein relativ kleiner Teil des Netzwerktreibers, der im Wesentlichen einen Sockel für die Applikation darstellt. Die INIC-Architektur entlastet somit den EHC. Zur Steuerung stellt der INIC dem EHC bzw. der MOST API („MOST Network Services“) mit der INIC API eine

Schnittstelle zur Verfügung. Die Funktionen des INIC sind durch einen Funktionsblock (FBlock INIC) gekapselt.

■ MOST Networking

MOST ermöglicht die Übertragung von kontinuierlichen Bitströmen (Bitstreaming) ohne Pufferung und unnötigen Overhead. Dazu speist ein ausgewiesenes MOST-Gerät (Timing Master) den MOST-Frame (Bild 4) mit einer festen Frequenz (44,1 kHz oder 48 kHz) in das üblicherweise optische Übertragungsmedium ein.

In einem MOST25-System stellt der MOST Frame 60 Streaming Channels zu je 8 bit (bzw. 15 Quadlets zu je 4 byte) zur Übertragung von kontinuierlichen Bitströmen zur Verfügung (Source Data Area). Die Übertragungsrate eines Streaming Channels ergibt sich entweder zu 352,8 kbit/s (44,1 kHz) oder zu 384 kbit/s (48 kHz).

Da die MOST-Geräte physikalisch zu einem Ring zusammenschlossen sind, muss jeder MOST-Frame jedes MOST-Gerät mit der vom Timing Master vorgegebenen Frequenz passieren. Sobald sich die entsprechenden Kommunikationspartner (Datenquelle und -senke) mit denselben Streaming Chan-

nels verbunden haben, beginnt das Bitstreaming (Bild 5).

Auf- und Abbau der Verbindung geschieht üblicherweise auf Anfrage durch den Funktionsblock Connection Master (CM; FblockID=0x03). Zu diesem Zweck stellt der CM die beiden Funktionen BuildSyncConnection und RemoveSyncConnection bereit.

Im Rahmen des Verbindungsaufbaus fordert der CM die entsprechende Datenquelle auf, zum Beispiel den TV-Tuner, sich die entsprechende Anzahl Streaming Channels beim Timing Master allokkieren zu lassen. Denn der Timing Master ist für die Verwaltung der „Channel Resource Allocation Table“ zuständig. Die Adressen der allokkierten Streaming Channels gibt der CM der Datensenke, zum Beispiel Display, weiter, damit sich diese mit den Streaming Channels verbinden kann. Zum Abschluss aktualisiert der CM die „Sync Connection Table“, über die er sämtliche synchronen Verbindungen verwaltet. Der Abbau funktioniert nach dem gleichen Schema.

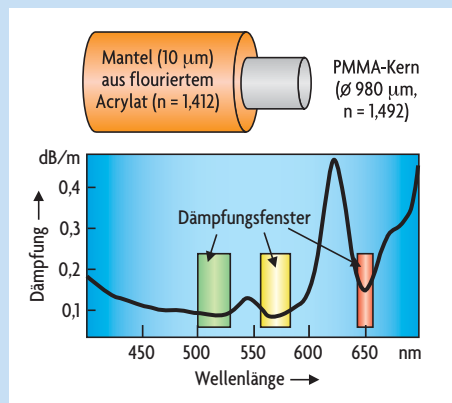
Um auch Datenpakete übertragen zu können, hat der Anwender die Möglichkeit, die Anzahl der Streaming Channels mittels Boundary Descriptor bis auf 24 (sechs Quadlets) zu verringern. Denn alle Streaming Channels, die nicht für das Bitstreaming reserviert sind, werden zum Packet Channel zusammengefasst. Während bei 44,1 kHz eine maximale Übertragungsrate von bis zu 12,7 Mbit/s möglich ist, erreicht man bei 48 kHz eine maximale Übertragungsrate von bis zu 13,8 Mbit/s. Verwaltet wird der Boundary Descriptor vom Funktionsblock Network Master (FBlockID= 0x02). Über die Funktion Boundary (FktId=0xA03) lässt sich dieser setzen.

Zur Übertragung von Datenpaketen kommt ein Schicht-2-Protokoll zum Einsatz. Der Frame setzt sich aus dem Arbitrierungsfeld, der Quell- und Zieladresse, dem Data Length Code, dem Datenfeld (48 oder 1014 byte) und der Datensicherung zusammen. Ein im Ring zirkulierender Token regelt den Buszugriff. Jenes MOST-Gerät, welches den Token vom Ring nimmt, darf auf den Packet Channel zugreifen.

Schließlich muss das MOST-System die zur Verwaltung und Steuerung notwendigen MOST-Kommandos übertra-

Signalübertragung mittels POF

Tritt ein Lichtstrahl von einem transparenten Medium in ein anderes über, so wird er gebrochen. Die Brechung ist um so stärker, je größer der Einfallswinkel ist. Das Medium, in dem der Lichtstrahl mit dem Lot den kleineren Winkel bildet, ist das optisch dichtere Medium. Beim Übergang vom optisch dichteren zum optisch dünneren Medium wird der Strahl vom Lot weg gebrochen. Der Brechungswinkel α kann berechnet werden, wenn die so genannten Brechzahlen n der beiden Medien bekannt sind (Snellius-Gesetz). Überschreitet der Lichtstrahl beim Übergang vom optisch dichteren zum optisch dünneren Medium den Einfallswinkel α_0 , so tritt Totalreflexion ein. Aufgrund dieser Eigenschaft kann man Licht in einem optischen Leiter transportieren. Im MOST-System setzt man üblicherweise Polymerfasern zur optischen Signalübertragung ein, deren Kern aus PMMA (Polymethylmethacrylat) von einem dünnen Mantel aus fluoriertem Acrylat umgeben ist. PMMA hat eine größere Brechzahl als fluoriertes Polymer.



Wenn der eingehende Lichtstrahl über dem Grenzwinkel liegt, wird aufgrund der Totalreflexion das Licht im Kern geführt. Die kleinsten Dämpfungen für das Übertragen von Licht in einer Step-Index-PMMA-Fasern erhält man bei 520 nm (grün), 560 nm (gelb) und 650 nm (rot). In erster Linie werden rote LEDs verwendet (Dämpfung: 0,14 dB/m), da sie sehr kostengünstig sind

Entwicklung, Test und Analyse von MOST-Systemen

Die Vector Informatik GmbH ist seit 1999 assoziiertes Mitglied der MOST Cooperation und bietet Unterstützung bei Entwicklung und Analyse von Infotainment-Lösungen im Automobil. Für Anwendungen wie High-End-Audiosysteme, Multimedia-Streaming, Telefonie und Navigation gibt es eine umfassende Palette an Analyse-, Entwicklungs- und Test-Werkzeugen, Hardware-Schnittstellen, Datenlogger sowie Schulungen und Dienstleistungen [2]. Das notwendige Basiswissen rund um die Steuergerätekommunikation im Automobil vermittelt die Vector Academy [3] im Rahmen von Seminaren zu CAN, LIN, FlexRay und MOST. *sj*

Literatur

- [1] www.mostcooperation.com
- [2] www.vector-group.net/most/de
- [3] www.vector-academy.de
- [4] Mayer, E.: Serielle Bussysteme im Automobil. Teil 1: Architektur, Aufgaben und Vorteile. *Elektronik automotive* 2007, H. 7, S. 70 – 73.
- [5] Mayer, E.: Datenkommunikation im Automobil. Teil 2: Sicherer Datenaustausch mit CAN. *Elektronik automotive* 2007, H. 8, S. 34 – 37.
- [6] Mayer, E.: Serielle Bussysteme im Automobil. Teil 3: Einfacher und kostengünstiger Datenaustausch mit LIN. *Elektronik automotive* 2008, H. 1, S. 40 – 43.
- [7] Mayer, E.: Serielle Bussysteme im Automobil. Teil 4: FlexRay für den Datenaustausch in sicherheitskritischen Anwendungen. *Elektronik automotive* 2008, H. 2, S. 42 – 45.

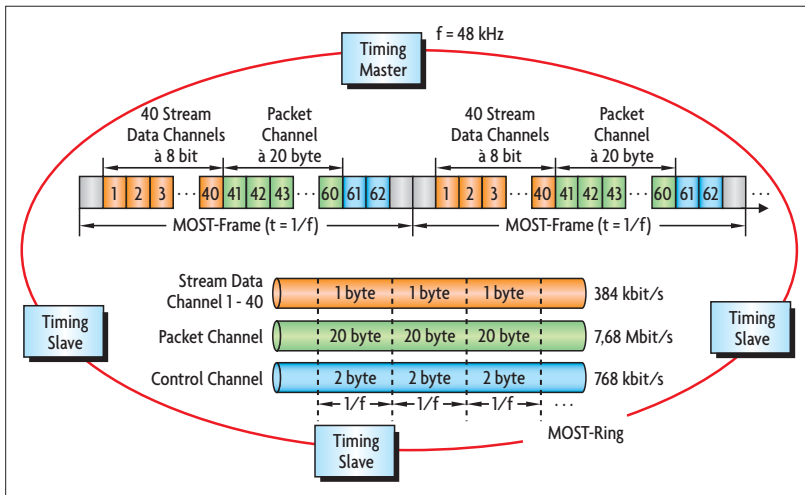


Bild 5. Prinzip des Bitstreamings: Der Timing Master überträgt mit der Frequenz von 48 kHz MOST-Frames. Es stehen 40 Streaming Channels (10 Quadlets) mit je 384 kbit/s zur Allokierung bereit (Boundary Descriptor = 0xA). Der Packet Channel (20 byte) stellt für die Übertragung von Datenpaketen eine Bandbreite von 7,68 Mbit/s zur Verfügung.

gen. Dazu kommen Control Messages (Bild 6) zum Einsatz, die im Control Channel übertragen werden. Zur Übertragung einer Control Message sind 16 MOST-Frames (MOST-Block) erforderlich. Die Übertragungsrate beträgt bei 44,1 kHz 705,6 kbit/s und bei 48 kHz 768 kbit/s. Auch der Übertragung der Control Messages liegt ein Schicht-2-Protokoll zugrunde. Der Buszugriff erfolgt mittels CSMA-Verfahren (Carrier Sense Multiple Access).

elektrischer Übertragungsmedien weit überlegen. Zu erwähnen ist vor allem die hervorragende elektromagnetische Störfestigkeit und die vergleichsweise hohe Signalübertragungsgeschwindigkeit von bis zu 500 Mbit/s. Außerdem stellt die Kombination aus POF, roter Leuchtdiode als Lichtquelle und einer Silizium-PIN-Fotodiode als Empfänger eine sehr günstige und vergleichsweise einfach handhabbare Form der optischen Signalübertragung dar.

Physical Layer

Zur Übertragung der Audio- und Videosignale im MOST-System sind heute Lichtwellenleiter aus Polymerfasern (POF; polymeroptische Faser) Stand der Technik (Kasten). In der Summe sind die technischen Eigenschaften der Polymerfasern denen

Mit MOST150 geht nach MOST50 aktuell ein MOST-System an den Start, dem diese Sender- und Empfänger-Technik zugrunde liegt und dem Anwender eine Übertragungsgeschwindigkeit von 150 Mbit/s zur Verfügung stellt. Die im Auto vergleichsweise kurzen Strecken von bis zu 20 m sind damit problemlos zu bewältigen.

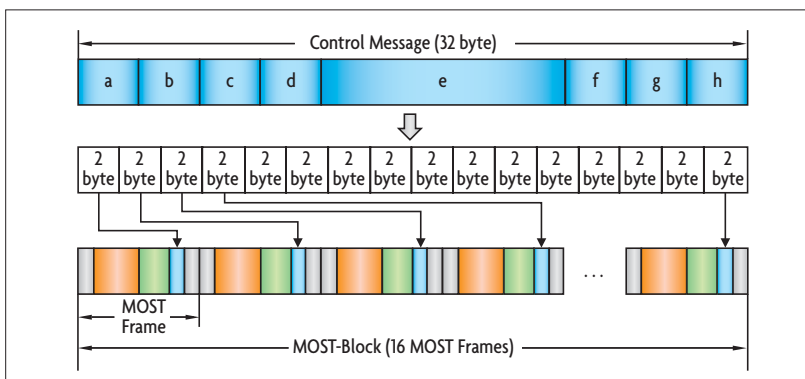


Bild 6. Zur Übertragung einer Control Message ist ein MOST-Block erforderlich. Die Control Message setzt sich zusammen aus: Arbitrierung (a), Zieladresse (b), Quelladresse (c), Message Typ (d), Datenfeld (e), Datensicherung (f), Bestätigung (g) und Reservierung (h).

Dipl.-Ing., Dipl.-Techpaed. Eugen Mayer
 hat an der FH Ravensburg/Weingarten Elektronik und an der Universität Stuttgart Elektrotechnik und Berufspädagogik studiert. Er arbeitet seit 1999 bei der Vector Informatik und ist dort als Senior Trainer tätig.
eugen.mayer@vector-informatik.de