



Media Oriented Systems Transport

Die MOST-Systembus Architektur

Christian Greth
s7277485@mail.inf.tu-dresden.de

Dresden, 17. Juni 2009



Gliederung

1. Einleitung und Motivation für MOST
2. Überblick über das MOST-Netzwerk
3. Funktionsweise eines MOST-Netzwerks
4. Organisation der Devices im MOST-Netzwerk
5. MOST-Protokoll
6. Ausblick und Zusammenfassung



Einführung

- Übertragung von Multimediadaten
- Kooperation der Firmen Audi, BMW, Daimler/Chrysler, Harmann/Becker und SMSC
- Einsatz in Fahrzeugen und Gebäuden

Motivation

- alte Bussysteme erfüllen Anforderungen an die hohen Datenraten nicht mehr
- pufferlose Übertragung (Streaming) von Multimediadaten nötig

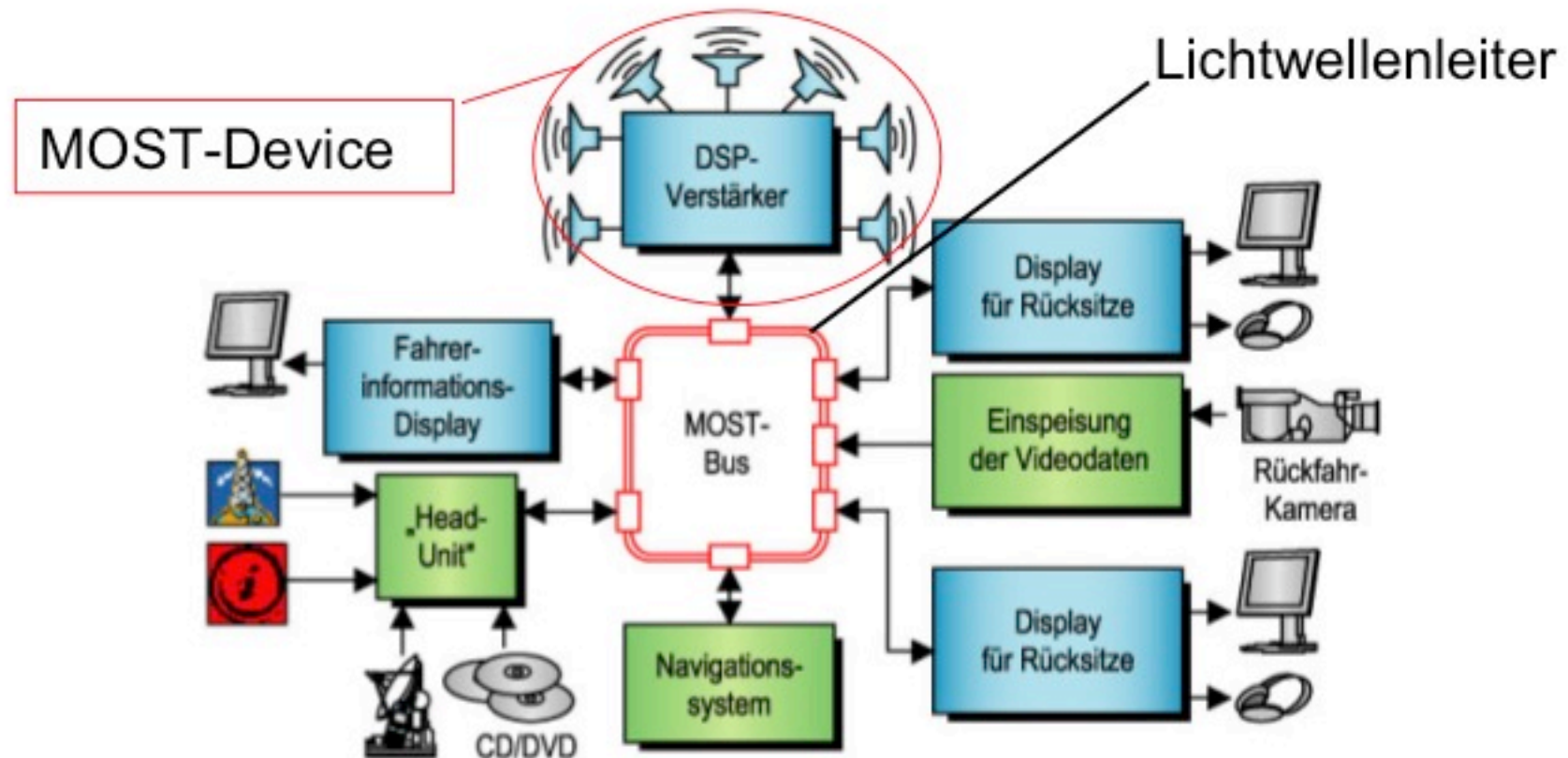


Merkmale

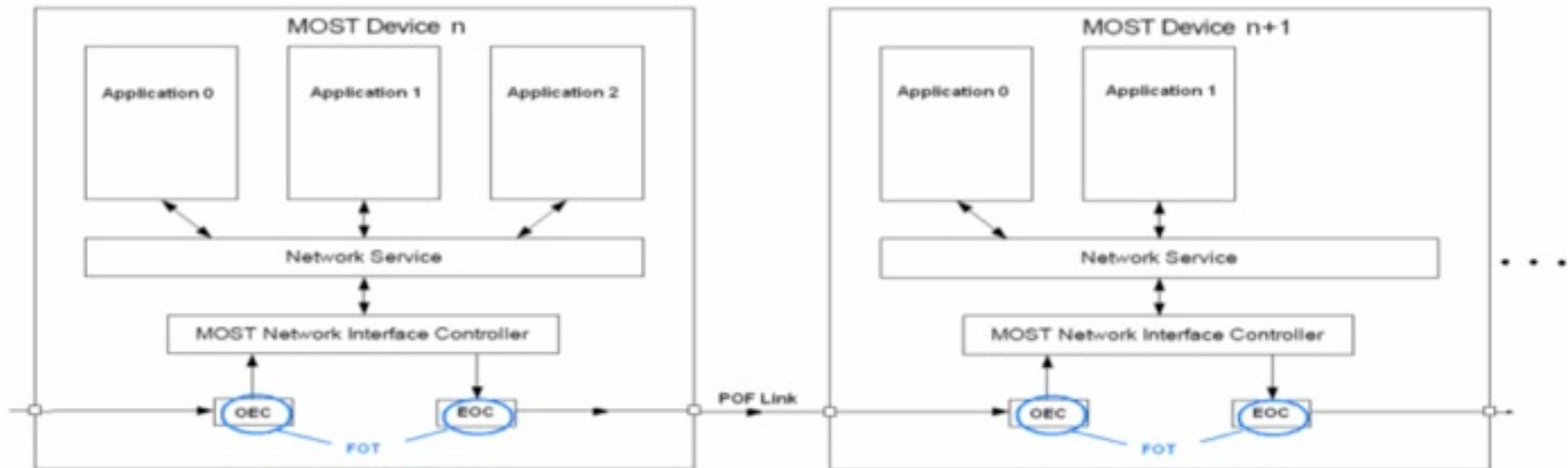
- Übertragungsraten: 24MBit/s Streaming-Daten für Multimedia und 14,4MBit/s Paket-Daten
- physikalische Schicht bildet ein optisches Medium (Plastic Optical Fibre)
- Ring-Topologie
- Hot-Plug-fähig
- Zugriff auf Medium mittels Zeitmultiplex (Time Division Multiplex - TDM)

- pufferloses Streaming von MPEG2-Video und CD-Audio möglich
- für administrative Zwecke existiert ein Steuerkanal mit bis zu 700kBit/s
- weiterhin Unterstützung für Standards aus der PC-Industrie (z.B. TCP-IP)

Überblick



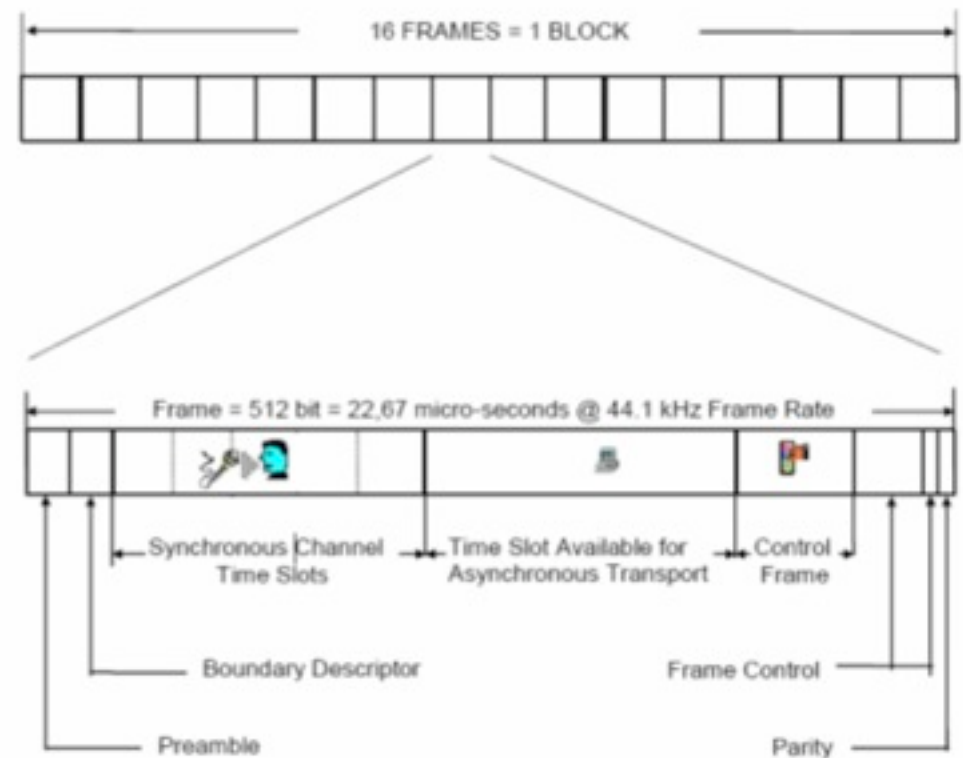
MOST-Device



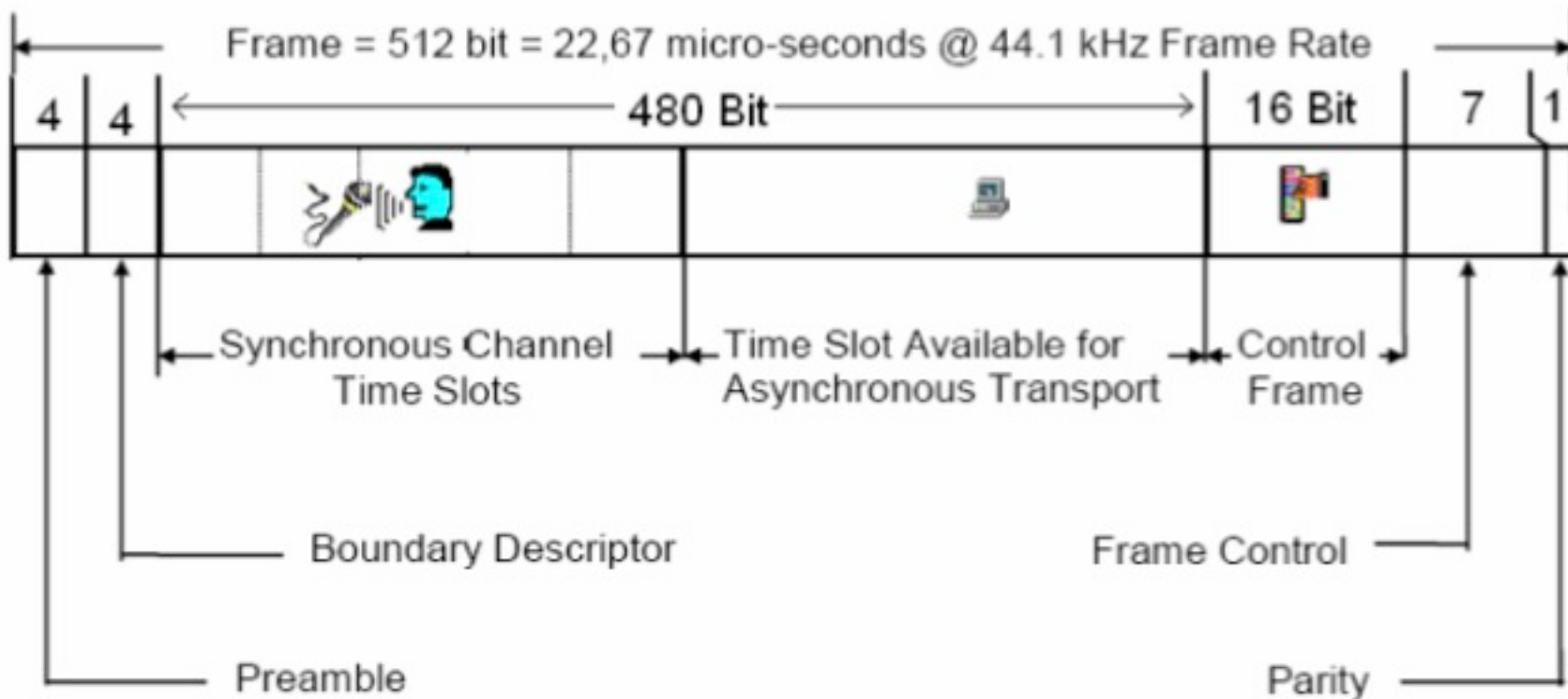
- Punkt-zu-Punkt-Verbindung zwischen Devices über POF-Link
- im FOT (Fibre Optical Transceiver) erfolgt Umwandlung:
 - optisch nach elektrisch (Optical-Electrical-Converter - OEC)
 - elektrisch nach optisch (Electrical-Optical-Converter - EOC)

Datenstruktur

- zu übertragende Daten werden in Blocks unterteilt
- jeder Block ist in 16 Frames geteilt mit jeweils 512 bit:
 - 4 bit für Preamble
 - 4 bit für Boundary Descriptor
 - 480 bit für synchrone und asynchrone Daten
 - 16 bit für Control-Data
 - 7 bit für Frame-Control
 - 1 bit für Parity-Check



Aufbau eines Frames





Medienzugriff

- Network-Interface-Controller verwaltet den Zugriff auf das Medium
- Zugriff mittels Time-Division-Multiplex (TDM)
- Medium über 3 unterschiedliche Kanäle nutzbar
 - Control-Channel
 - Synchronous-Channel
 - Asynchronous-Channel
- max. Datenrate: $480 \text{ Bit/Frame} * 44100 \text{ Frames/s} = 21,16 \text{ MBit/s}$

Datenübertragung

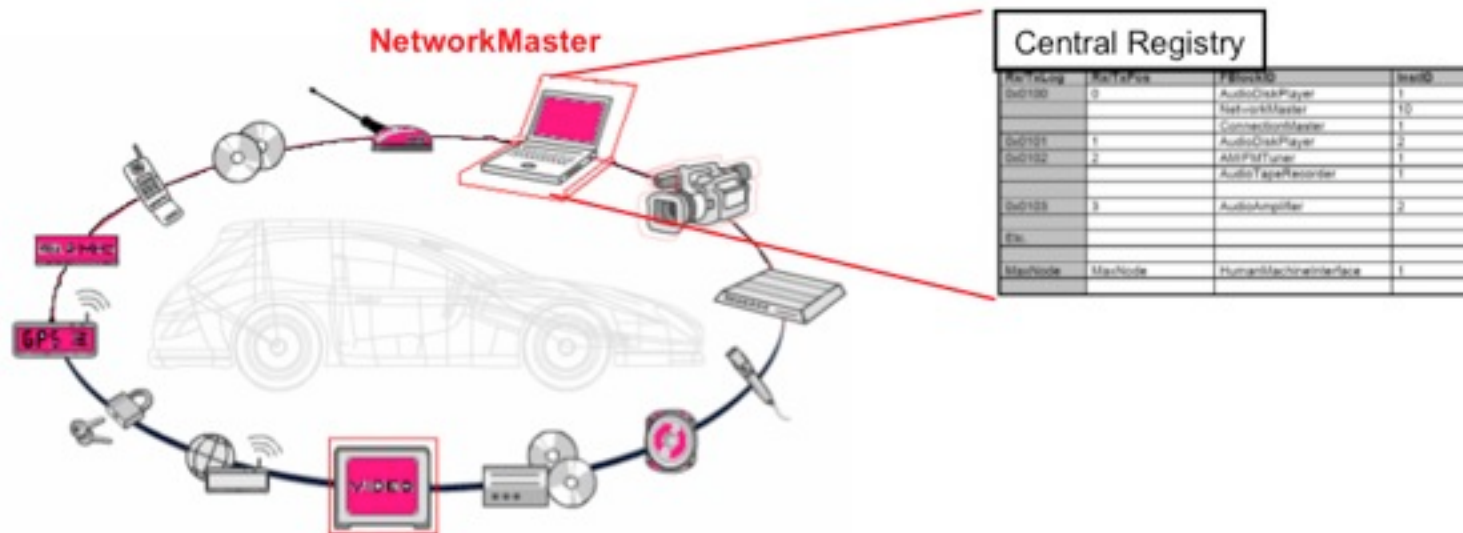
- Aufteilung der Daten in den Kanälen erfolgt über Telegramme
- mehrere Telegramme bilden eine Nachricht
- Aufbau eines Telegramms:

48 Bit	4 Bit	12 Bit	8 Bit	8 Bit	...	8 Bit
Adresse	Tel ID	Tel Len	Data 0	Data 1	...	Data 41

- Telegramm ist maximal 50 Byte groß
- TelID gibt die Identifikation des Telegramms an (z.B. um die Telegramme in einer Nachricht unterscheiden zu können)
- TelLen gibt die Anzahl der folgenden Bytes an

Network-Master und -Slaves

- es gibt nur ein Network-Master-Device
- Network-Master-Device kontrolliert und organisiert den Zugriff der Network-Slaves auf das Medium
- alle Network-Slaves durch die Central-Registry verwaltet



Central-Registry

Rx/TxLog	Rx/TxPos	FBlockID	InstID
0x0100	0x0400	Timing Master	1
		AudioDiskPlayer	3
0x0104	0x0404	NetworkMaster	1
0x0101	0x0401	Receiver	2
0x0102	0x0402	AM/FMTuner	2
		ConnectionMaster	1

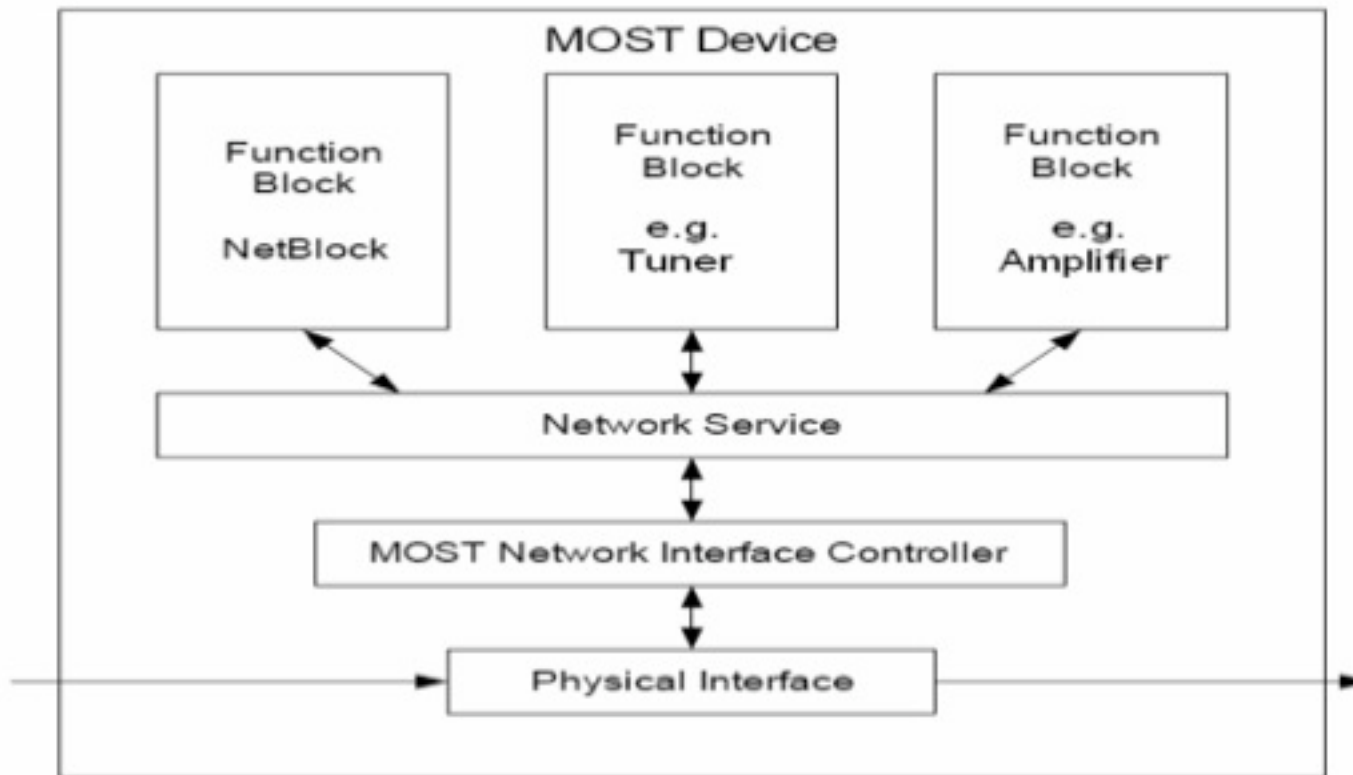
- Rx/TxLog gibt die logische Position im Netzwerk an (Startzustand)
- Rx/TxPos gibt die wirkliche Position im Netzwerk an (wird nach der Initialisierung durch den Network-Master zugewiesen)
- FBlockID gibt den Funktionsblock in einem Device an
- InstID wird zur Unterscheidung mehrerer gleicher Funktionsblöcke genutzt



Zustände eines Devices

- Devices können 3 unterschiedliche Zustände annehmen
 1. NetinterfacePowerOff (Knoten für Netzwerk nicht existent)
 2. NetinterfaceInit (Device synchronisiert und wartet auf Adresse)
 3. NetinterfaceNormalOperation (Device ist bereit)
- Zustandswechsel müssen dem Network-Master übermittelt werden
 - Network-Master muss dann Registry neu aufbauen
 1. Registry-Invalid-Signal an alle Devices
 2. Devices beenden Datentransfer
 - neue Registry muss den Devices mitgeteilt werden

MOST-Protokoll (1)





MOST-Protokoll (2)

- Struktur: DeviceID.FBlockID.InstID.FuncID.OptParam()
 16Bit 8Bit 8Bit 12Bit 4Bit = 48Bit
- 3 unterschiedliche Funktionen: Property, Event und Method
- Property zum Setzen und Lesen von Attributen eines Funktionsblocks
- Events signalisieren Änderungen innerhalb eines Funktionsblocks
- Method löst in einem Funktionsblock eine Funktion aus

Bsp: Autoradio.Tuner.1.Frequency.Set(92,1 MHz)



Zusammenfassung

- gleichzeitige Übertragung von Streaming-Daten und Paket-Daten
- pufferloses Streaming möglich
- optisches Medium gilt als robust und flexibel

Ausblick

- höhere Datenraten durch größere Frames (MOST50 und MOST150)
- Nachbildung bereits vorhandener Techniken (z.B. Ethernet)