

# Universell einsetzbar

## Der Einsatz von MOST als universelles Kommunikationsprotokoll



Eine Herausforderung der heutigen Head-Unit-Entwicklung ist, einen schnellen und flexiblen Datenaustausch zu ermöglichen, um dem Endnutzer eine ergiebige und komfortable Multimedia- und Infotainment-Umgebung zu gewährleisten, z.B. für die drahtlose Anbindung von mobilen Endgeräten. Die Antwort zur Realisierung dieser Aufgaben ist das Infotainment-System MOST.

Von Tobias Follner

**M**OST ist mehr als ein leistungsfähiges, physikalisches Netzwerk, da die MOST-Spezifikation, wie MS-DCOM oder CORBA, alle notwendigen Datentypen sowie Mechanismen zum einfachen Datenaustausch und zur Realisierung eines Schnittstellenverfahrens in einer verteilten Rechenumgebung enthält. Dieser Beitrag vermittelt ein umfangreiches Verständnis über die Anforderungen zur Realisierung solcher Szenarien, wie das Protokoll selbst, sowie einen Überblick über die Architektur.

Um modernste Infotainment-Systeme bereitzustellen, müssen verschiedene Architekturen mit zwei oder sogar mehr Embedded-CPU's sowie Multi-Core-CPU-Szenarien realisiert werden – in den meisten Fällen mit unterschiedlichen Betriebssystemen betrieben und über unterschiedliche Zwischenglieder verbunden. Das erfordert eine skalierbare Architektur mit direkter oder drahtloser Anbindung von 1, 2, ... n CPU's hinter einem physikalischen MOST-Knoten für jede Geschwindigkeitsklasse (MOST25,

MOST50, MOST150). Mögliche Einsatzszenarien zeigt **Bild 1**.

### ■ Unterstützung von proprietären und Standard-Protokollen gefordert

Für alle in ein System integrierten Anwendungen werden inhomogene Betriebssystem-Szenarien unterstützt, wobei z.B. ein Linux-Prozess OSEK-Aufgaben auf Augenhöhe realisiert oder eine Windows-Multithread-Anwendung zum Datenaustausch und zur Triggerung von Fernverarbeitungsaufufen angebunden werden kann. Im Weiteren verlangen zukünftige Anwendungsfälle die Fähigkeit, für eine serielle „Inter Processor Connection“ (IPC) unterschiedliche Lösungen zu unterstützen, die auf verschiedenen physikalischen Schichten realisiert sind, welche wiederum auf Standard- (z.B. TCP/IP) oder proprietären Protokollen basieren. Ein künftiges MOST-System sollte auch Modi umfassen, bei denen jede CPU als ein eigenständiges, unabhängiges MOST-(Sub-)System agiert, welches reine MOST-protokollbasierende Konnektivität für alle lokal angebundenen MOST-Objekte ermöglicht (Funktionsblöcke und Shadows) und – beim Schließen des IPC-Links – auch zu jenen Objekten, die auf Nachbar-CPU's ausgeführt werden. Für eine geräteinterne MOST-Kommunikation, verborgen für Verbindungen mit einem oder mehre-

ren MOST-Ringen, müssen alle integrierten Anwendungen als parallele Instanzen der MOST-Kanäle unterstützt werden.

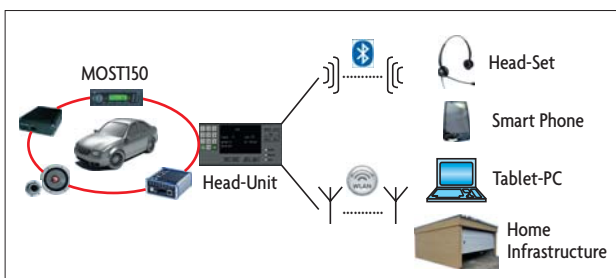
MOST erstellt die perfekte Lösung zur Realisierung all jener Anwendungsfälle und enthält alle Mechanismen, die geeignet sind, Remote-Aufrufe für automobiler Anwendungen (wie es MS-DCOM und CORBA in LAN- oder WAN-Netzwerken tun) auszuführen. Zum Austausch von signalbasierenden Informationen definiert es einen umfassenden Satz an Datentypen und Mechanismen und liefert für den Betrieb einen erweiterbaren und anpassbaren Satz vordefinierter Anwendungsschnittstellen und Nachrichtensequenzen. MOST-Protokolle sind komplett betriebssystem-unabhängig und portabel. Ein virtueller MOST-Modus ermöglicht neben physikalisch angebundenen Steuergeräten die Integration von mobilen Endgeräten über Bluetooth oder WLAN. Ein abstrahierter Überblick erläutert die Architektur eines solchen Systems (**Bild 2**).

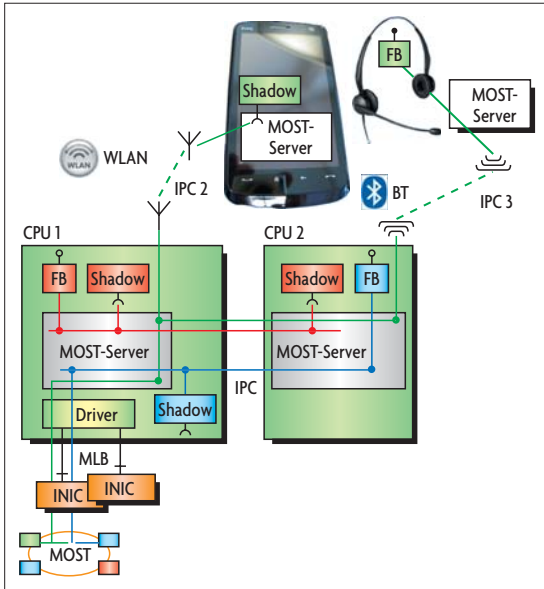
### ■ Server unterstützt parallele MOST-Instanzen

Die Funktionsblöcke (FB) und Shadows sind in diesem Überblick lokal über einen Server im Sinne eines virtuellen MOST-Modus verbunden, der ein größeres MOST-Cluster bildet. Die Server unterstützen parallele Instanzen von MOST: rein interne, die von einem virtuellen MOST-Modus betrieben werden, sowie Verbindungen mit einem externen MOST-Ring, der über INIC eingerichtet ist.

Im Vergleich bildet MOST die Basis, um neue Elemente anzubinden oder abzukoppeln (FBs oder Shadows), wobei mehrere MOST-Kanäle parallel entweder in realem oder virtuellem Modus betrieben werden können. Unnötiger Verkehr kann, basierend auf

**! Bild 1.** Anbindung eines Fahrzeugnetzwerks an verschiedene Infotainment-Geräte.





**Bild 2. Virtueller MOST-Modus:** Neben konventionellen Steuergeräten können via Bluetooth oder WLAN auch mobile Endgeräte angeschlossen werden.

dem „MOST Function ID Level“, durch Senden von MOST-Nachrichten innerhalb des Clusters vermieden werden. Ein weiterer, künftiger Anwendungsfall kann sein, MOST-Systeme in Heim-Infrastrukturen anzuwenden, um elektronische Geräte von einem zentralen Punkt aus oder mit einer Fernbedienung zu betreiben (z.B. aus der Ferne die Infotainment-Datenbank des Fahrzeugs neu zu laden, während man im Wohnzimmer bleibt).

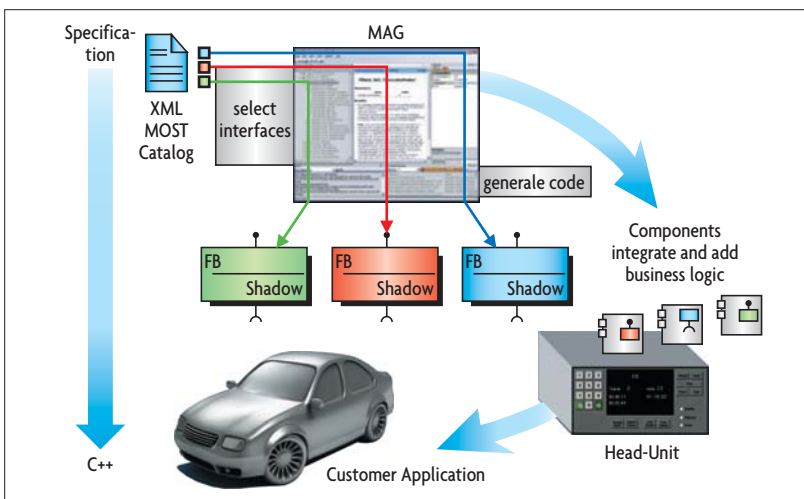
K2L begegnet der Herausforderung für eine reibungslose Kommunikation zwischen allen integrierten ECUs mit dem Automotive Communication System (ACS), um die Arbeitsabläufe sowohl für OEMs als auch für die Zulie-

ferer zu vereinfachen. K2L ACS enthält ein Framework für die embedded Applikationsentwicklung und eine Werkzeugkette, die die Produktivität von Applikations-Entwicklern steigern. Es liefert zur Vereinfachung der Applikationsentwicklung innerhalb einer umfangreichen C++-Klassen-Bibliothek einen umfassenden Satz von APIs. Die Architektur erlaubt die nahtlose Verteilung der Anwendungen auf mehrere CPUs, die sich einen physikalischen MOST-Knoten teilen. ACS erlaubt

den Entwicklern, sich auf die Programmierung von Verhaltensaspekten und Business-Logik zu konzentrieren, ohne dabei die Komplexität der MOST-Schnittstellen-Logik und Datentypen beachten zu müssen oder den MOST-Funktionskatalog neu zu schreiben. Charakteristische kundenspezifische Funktionsanfragen auf Anwendungsniveau können integriert werden, sofern sie nicht Teil der Spezifikation sind. ACS unterstützt mehrere Fälle von physikalischen oder virtuellen MOST-Netzwerken, wobei eigene Implementierungen auf nahezu jeder Ebene (z.B. UART, SPI, USB, Ethernet) anwendbar sind. Der Server wurde für die nahtlose Zusam-

menarbeit mit anderen K2L-Produkten wie den K2L-MOST-Treibern oder das Universal Gateway (UGW) entwickelt. Der Arbeitsablauf ist in Bild 3 dargestellt.

Der Benutzer, der die MOST-Technologie für eine moderne Anwendungsentwicklung anwendet, profitiert durch die verkürzte Zeitdauer zur Generierung von soliderem Code, ohne die Schnittstellen für interne oder externe MOST-Anwendungen neu zu schreiben. Schnittstellen und Sequenzen (MOST-Funktionskatalog und -Sequenzdiagramme) können klar in maschinenlesbarer Form definiert werden. Für die Erzeugung von Schnittstellen und stereotypen Mustern, z.B. die Notifikation, wird kein manuelles Codieren benötigt. Alle Datentypen für Codierung und Decodierung können mittels abstrakter und symbolischer Signal- oder Funktionsnamen angesprochen werden. Die Implementierung von Business-Logik wird durch Herleitung eigener Anwendungen mit objektorientierten Strukturen aus generierten Stub-Klassen vorgenommen. Dadurch werden Round-Trip-Entwicklungsprobleme von Spezifikations-Updates vermieden. Das MOST-Protokoll erlaubt die Bereitstellung eines vom eingesetzten Betriebssystem und den CPU-Plattformen unabhängigen, identischen Programmiermodells und der Schnittstelle zur Applikationsprogrammierung. sj



**Bild 3. Arbeitsablauf beim Automotive-Communication-System von K2L.**



**Tobias Follner**

hat einen Bachelor-Abschluss in Vertriebsingenieurwesen von der Hochschule Karlsruhe Technik und Wirtschaft und bereitet sich auf den Master in Wirtschaftspsychologie während eines berufsbegleitenden Programms an der Hochschule für angewandtes Management in Erding vor. Seit 2008 arbeitet er in der Vertriebs- und Marketing-Abteilung der K2L GmbH.