Window to the CE world

Using UPnP in MOST-based infotainment systems

In this article it will be discussed how UPnP (Universal Plug and Play), a standard from the home entertainment and consumer electronics world, which provides standard interfaces for many entertainment functions, can be leveraged in an automotive MOST-based infotainment system.

By Dr. Alexander Leonhardi, Dr. Matthias Stümpfle, Rico Hauke and Thomas Gschwandtner

Vehicle infotainment systems provide entertainment, navigation and communication functions not only to the driver of a vehicle but also to its passengers. For high-end vehicles infotainment systems have become increasingly important and provide more and more advanced functions, similar to home entertainment and mobile systems [1]. For example, more and more drivers and passengers are used to bring their favorite music into the car – located on mobile media, mobile consumer electronics devices, or off-board servers accessible via mobile networks.

The MOST network provides the means to transmit multimedia streams, control commands and mass data over a single bus system using an optical physical layer. Besides the robust optical physical layer one of the biggest advantages of the MOST network is the synchronous data transmission that allows for a simple transfer of multimedia data. Additionally, the MOST standard defines interfaces for different automotive-specific applications that have been used successfully in an ever increasing number of vehicle systems. With the current speed grade of 150 Mbit/s the MOST network now also provides an Ethernet channel designed for the simple transmission of IP traffic.

UPnP was defined to connect the devices of a home entertainment network [5]. It is based on standard Internet protocols (e.g. TCP/IP and HTTP) that are not domain-specific. UPnP provides the mechanisms for detecting devices in an IP network and to provide and access services. Currently, UPnP is seeing wide-spread use e.g. for the communication between streaming servers and clients and for managing DSL routers. Using UPnP's device model the DLNA (Digital Living Network Alliance) has defined guidelines (e.g., media formats) for the interoperability between consumer electronics devices [3]. In cooperation with the UPnP forum, various interfaces for entertainment functions have been defined as well as streaming protocols and services for exchanging multimedia data. If UPnP is mentioned in this paper, the requirements for interoperability from the DLNA are automatically included.

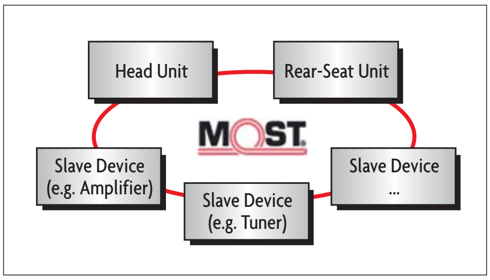

Figure 1. A typical MOST network in a vehicle.

In this article different scenarios for using UPnP in a MOST network will be discussed. In a first step UPnP is used for communication between a front and a rear-seat entertainment system. Therefore, both the standard protocols and implementations from the home entertainment world can be used as well as the robust optical physical layer of the MOST network and as out-of-band communication mechanism the light-weight synchronous communication channels of the MOST network. In a second step UPnP is used for accessing external mobile media devices and to access legacy MOST devices through a gateway device that maps appropriate UPnP services to the MOST interfaces. As a third step, the goal of using UPnP as a standard communication mechanism is suggested.

Using UPnP in a MOST-based infotainment system

In this section we propose three different steps for introducing UPnP into a MOST-based infotainment sys-

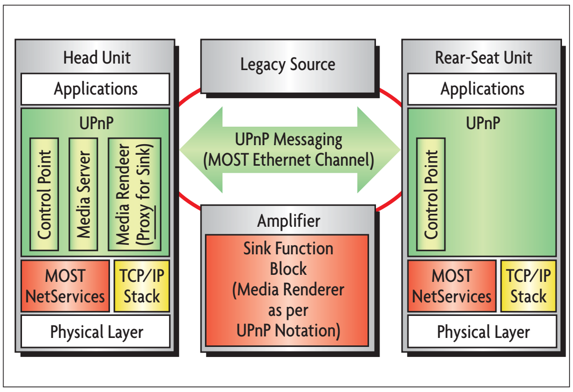

Figure 2. Step one – UPnP for communication between HU and RU.

tem. The goal is to gradually introduce it in order to allow for backward compatibility and to preserve the strengths of both standards. To achieve this, interfaces and implementations should – when possible – be changed as little as possible, even if this has some performance penalties. For example when using standard UPnP protocols in an automotive environment as UPnP is not optimized for high transfer rates.

The following model of an infotainment system is assumed (**figure 1**): a head unit (HU) provides the user interface for the driver as well as the processing power for the corresponding applications. Similar to the HU a rear-seat unit (RU) provides the user interface and applications for the rear-seat passengers. HU and RU are connected via a MOST150 network and can access various specialized slave devices such as an amplifier or a digital radio tuner

Based on this system model the following three steps for introducing UPnP mechanisms are proposed:

- Step one (**figure 2**): The HU and RU will both implement a UPnP stack and realize appropriate functions by communicating via UPnP mechanisms. For example the RU can use the standard UPnP mechanisms for browsing and accessing music files stored on the HU, with the HU acting as a UPnP AV media server. To realize step one, UPnP control commands are exchanged via TCP/IP on the MOST Ethernet channel provided by MOST150. To transmit streaming data, however, the UPnP streaming mechanisms will be mapped to the MOST synchronous (audio) or isochronous (video) channels. Additionally, the HU provides a media renderer proxy for the external MOST amplifier.
- Step two (**figure 3**): Applications on the HU and RU are implemented completely in a UPnP environment. To this end either the HU or RU need to provide proxy functions for the MOST slave devices. Because standard UPnP services often do not offer the functions necessary for a vehicle environment or its specific user interface requirements, appropriate services have to be defined corresponding to the MOST FBlocks of the slave devices. Using these profiles external UPnP devices will be able to access the functions of the vehicle infotainment system (e.g., providing sound output via the vehicle's amplifier system). In this context, UPnP security mechanisms will have to be considered in more detail.
- Step three (**figure 4**): Finally, UPnP is used as the general mechanism for communication inside a vehicle infotainment system. All applications are implemented in a UPnP environment and all control commands are exchanged using UPnP mechanisms. Therefore, also the MOST slave devices have to implement a UPnP stack and provide services according to the UPnP services defined in the previous step. For example an amplifier, which has very little processing power, will still be accessed through a proxy service on the HU.

Implementing UPnP in a MOST environment

In this section, we will discuss basic aspects that have to be addressed to realize the concepts described above. An example can be found in the article „on the integration of the MOST network into the HAVi (Home Audio Video Interoperability) standard“ [2] [6].

System architecture

The system architecture for using UPnP in a MOST environment as indicated in figures 2 to 4 is as follows: The basic requirement for UPnP is the ability to communicate via TCP/IP. On a MOST system the HU and RU, and for step three also slave devices, will implement a TCP/IP stack on top of the MOST Ethernet channel [4]. The UPnP commands for discovery, description, control, and eventing of devices and services are exchanged via this channel.

Both HU and RU applications implement control point functions, which are used to access services in the UPnP network. During startup of the MOST system, the devices in the network are detected using the regular MOST network management mechanisms. For step one and two, the respective proxy services are then initialized on the HU (or the RU) presenting the devices' functions – corresponding to their FBlocks – as appropriate UPnP services (e.g., the content directory service provided by a media server allowing a control point to enumerate the content that the media server can offer). A proxy is responsible for translating UPnP requests to corresponding MOST commands sent to legacy devices and notifications received from legacy devices into corresponding UPnP events.

When setting up the communication between devices, the control points

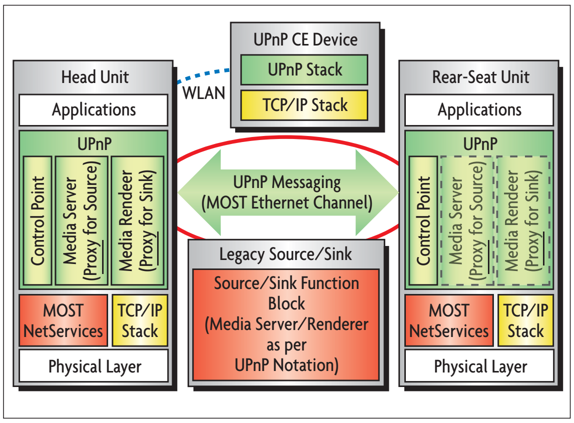

Figure 3. Step two – UPnP used as programming environment and to access mobile devices.

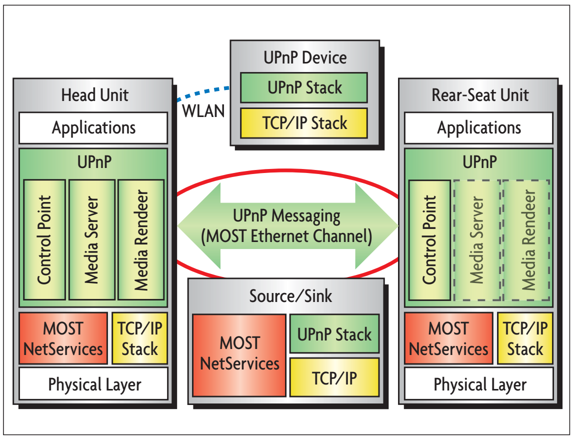

Figure 4. Step three – UPnP as general communication mechanism for a MOST system.

have to be aware of the MOST resources and, for example, set up streaming connections accordingly.

Stream management

As one of the most important aspects, the streaming management functions of UPnP have to be able to use the synchronous and isochronous channels of the MOST network for streaming audio and video data respectively. To this end the UPnP resource description will have to be extended to define a MOST synchronous/isochronous channel and its required parameters (e.g., sampling rate). To identify a channel on the MOST network the MOST connection label is contained in the ConnectionId of the respective Uniform Resource Identifier (URI).

A UPnP control point in a MOST system will integrate MOST resources when matching protocols and formats for establishing a connection between a media server and renderer. The media server translates the corresponding PrepareForConnection request into the appropriate Allocate command on MOST and returns the connection label as part of the ConnectionId. When this request is sent with the ConnectionId to the media renderer afterwards, it is able to perform a MOST Connect with this connection label. To close the channel, media server and media renderer execute the corresponding MOST DisConnect and DeAllocate commands upon the ConnectionComplete request from the control point.

To integrate devices that support UPnP streaming via HTTP in step two, for example when a mobile device uses the amplifier of the vehicle system for sound output, the appropriate media renderer and media server will have to be able to receive and to generate HTTP streaming data. In this case the media renderer proxy for the MOST amplifier on the HU will have to translate the HTTP streaming data for a MOST channel to the amplifier that is established when the appropriate PrepareForConnection request is received. *sj*

Dr. Alexander Leonhardi

is responsible for system architecture of infotainment systems in the development Electrical/Electronic Telematics unit at Daimler AG. He represents Daimler AG in the MOST Cooperation Technical Coordination Committee.

alexander.leonhardi@daimler.com

Rico Hauke

is responsible for networking architecture of infotainment systems for the C-Class and E-Class at Daimler AG. He represents the Daimler AG in the MOST Device Architecture working group.

rico.hauke@daimler.com

Dr. Matthias Stümpfle

is head of the department Telematics System Architecture and Platforms in advance development at Daimler AG. He is a member of the MOST Cooperation Steering Committee.

matthias.stuempfle@daimler.com

Thomas Gschwandtner

works in advance development at Daimler AG in the department Telematics System Architectures and Platforms on the topics of software architecture and networking including the MOST network. He represents the Daimler AG in the MOST Device Architecture working group.

thomas.gschwandtner@daimler.com

Literature + Links

[1] *Leonhardi, A.:* Networking Vehicle Infotainment Systems with MOST. 1st International MOST Conference and Exhibition. Stuttgart 2009.
[2] *Leonhardi, A.; Gschwandtner, T.; Simons, M.; Vollmer, V.; de Boer, G.:* Networking In-Vehicle Entertainment Devices with HAVi, in Advanced Microsystems for Automotive Applications 2004. P. 295-312. ISBN 3-540-20586-1.
[3] www.dlna.org
[4] www.mostcooperation.com
[5] www.upnp.org
[6] *Vollmer, V.; Simons, M.; Leonhardi, A.; de Boer, G.:* Architekturkomponenten für zukünftige Multimediasysteme im Fahrzeug. Konferenzbericht zur 24. Tagung Elektronik im Kfz. Haus der Technik. Essen 2004.