Migration of a classic infotainment network to Linux/AGL
Roland Trissl, Senior Applications Engineer
June 2018
Agenda

- Classic infotainment network (MOST®)
- Migration to Linux-based MOST
- Migration to AGL-based UNICENS
- Migration to AGL-based MOST
Classic infotainment network (MOST®)
Media Oriented Systems Transport (MOST) is:

- A synchronous network based on Microchip’s Intelligent Network Interface Controller (INIC) network controllers
- Uses Microchip’s MOST NetServices library (MNS) to communicate with INIC inside every device
- An application layer standard that defines communication (function blocks) and system behaviour
**Data types**

**Control data**
Control Messages used for Function Blocks

**Streaming Data**
Synchrous streams (PCM audio etc.) or Isochronous Streams (MPEG video etc.)

**Packet Data**
Ethernet packets (TCP/IP, UDP, etc.) or other packets
Communication between devices is using *Functions* implemented in each device.

Each node contains *MOST NetServices* SW stack to communicate with network controller.

**System Setup** and establishing of **Streams** via *NetworkMaster* and *ConnectionMaster* functionality (MSMM) in Head Unit and Functions in each device.

**Device Control** via Functions.
Master node

MSMM: MOST System Management Module contains ConnectionMaster, NetworkMaster
MNS: MOST NetServices Library

OS or baremetal
MOST Apps, PM, Diag
Ether net Apps
MOST DRIVER

(I2C/MediaLB/I2S/SP1/USB) Connection to INIC Network Controller
NW functionality/interfaces

- **System Setup**
  - NetworkMaster
  - ConnectionMaster
  - MOST App
  - FBloks
  - PowerMgmt
  - Diagnosis

- **Device Control**
  - MOST App
  - FBloks
  - Ethernet Apps

- **Streaming**
  - I²S
  - MediaLB
  - USB
  - via Driver
  - App HW
Migration from Classic MOST® to Linux-based MOST
Motivation

- Replace single devices with Linux devices
  - Take advantage of Linux benefits
  - No change in network behaviour
  - Reuse as much MOST® App code as possible
- Keep some existing devices unchanged
System Setup and establishing of Streams via NetworkMaster and ConnectionMaster functionality (MSMM) in Head Unit and Functions in each device

Each node contains NetServices SW stack to communicate with network controller

Communication between devices is using App Functions implemented in each device

Still the same!

Device Control via Functions
Master node

- Instrument Cluster
- AM/FM Radio
- Amplifier
- Head Unit
- Intelligent Antenna
- Blu-ray Player
- RSE Display

**Linux**

- MOST® Apps, PM, Diag
- Ethernet Apps
- ALSA V4L2 cdev

**MOST LINUX DRIVER**

- MSMM
- MNS

(\(I^2C/MediaLB/I^2S/SPI/USB\))

Connection to INIC Network Controller

**MSMM**: MOST System Management Module contains ConnectionMaster, NetworkMaster

**MNS**: MOST NetServices Library
NW functionality/interfaces

**System Setup**
- NetworkMaster
- ConnectionMaster
- MOST App
- FBblocks
- PowerMgmt
- Diagnosis

**Device Control**
- MOST App
- FBblocks
- Ethernet Apps

**Streaming**
- ALSA
- Video4Linux
- cdev
- App HW
Linux/MOST – pros and cons

- Existing devices can still be used
- Standard MLD available
- Ethernet, ALSA, Video4Linux can interface directly to MLD
- MOST apps still interfacing to well-known MNS and MSMM API
- Reuse of classic MOST app code
- No benefits of AGL
Migration from Classic MOST® to AGL based UNICENS
Motivation

- Take advantage of AGL benefits
- Take advantage of UNICENS benefits
  - Centralized network management
  - Slim nodes without SW possible
- Build new network without legacy MOST® devices
UNICENS

- Unified Centralized Network Stack (UNICENS) is:
  - A software stack used for network setup and configuration
  - Supports a synchronous network based on Microchip’s INIC network controllers
  - Does not provide a standard for application related communication or system behaviour
Data types

Control data
Control Messages used for Function Blocks

Streaming Data
Synchrous streams (PCM audio etc.) or Isochronous Streams (MPEG video etc.)

Packet Data
Ethernet packets (TCP/IP, UDP, etc.) or other packets
**Communication between Smart devices via Ethernet or Control messages.**

**Slim Device without SW. Network controller is configured remotely from root node.** Remote GPIOs and Remote I²C available.

**System Setup** and establishing of **Streams** via **UNICENS** SW in Head Unit (root node).

**Device Control** via Ethernet or Control messages.
UNICENS Binding

- JSON API
- UNICENS Integration
  - Command Queue
  - Config XML Parser
- UNICENS Library
UNICENS Configuration

- Configuration by a single XML file
- Describes streaming and devices

```xml
<!-- UNICENS Root Node, USB Server Node, -->
<Node Address="0x200">
  <SyncConnection MuteMode="NoMuting">
    <USBSocket EndpointAddress="0x2" FramesPerTransaction="128"/>
    <MOSTSocket Route="Music" Bandwidth="4"/>
  </SyncConnection>
</Node>

<!-- 1st Microphone, Slim Node -->
<Node Address="0x210">
  <StreamPort ClockConfig="64Fs" DataAlignment="Left16Bit"/>
  <SyncConnection MuteMode="NoMuting">
    <StreamSocket StreamPinID="SRXA0" Bandwidth="4"/>
    <MOSTSocket Route="Microphone1" Bandwidth="4"/>
  </SyncConnection>
</Node>
```
UNICENS Binding – API

- JSON API using defined verbs
  - Automatic system setup
  - Notification on network events
  - Tx/Rx of control messages
  - Tx/Rx of remote commands
  - Switching of streaming routes
  - […]
Smart node

- Head Unit
- Intelligent Antenna
- Instrument Cluster
- AM/FM Radio
- Amplifier
- Blu-ray Player
- RSE Display

AGL Linux

App Framework

- Apps
- Diag
- PM
- ALSA
- V4L cdev

MOST® LINUX DRIVER

Connection to INIC Network Controller

(I²C/MediaLB/I²S/SPI/USB)
Slim node

- Instrument Cluster
- AM/FM Radio
- Amplifier
- Head Unit
- Intelligent Antenna
- Blu-ray Player
- RSE Display

No SW!

(I\textsuperscript{2}C/MediaLB/I\textsuperscript{2}S/SPI/USB)
Connection to INIC Network Controller

App HW
NW functionality/interfaces

System Setup
- UNICENS Binding
  - PowerMgmt Diagnosis

Device Control
- Control Apps
- Ethernet Apps

Streaming
- ALSA
- Video4Linux cdev
- App HW
AGL/UNICENS – pros/cons

- Centralized network management
- Easy network configuration (XML)
- Slim nodes without SW possible
- UNICENS Binding seamlessly integrates in AGL security architecture
- JSON interface to applications
- No legacy MOST devices supported
Migration from Classic MOST® to AGL based MOST
Motivation

- Replace single devices with AGL devices
  - Take advantage of AGL benefits
  - No change in network (MOST®) behavior
  - Reuse MOST application code whenever possible
- Keep some existing devices unchanged
Communication between devices is using *Functions* implemented in each device.

*System Setup* and establishing of **Streams** via *NetworkMaster* and *ConnectionMaster* functionality (MSMM) in Head Unit and Functions in each device.

Each node contains *NetServices* SW stack to communicate with network controller.

**Device Control** via Functions.
Master node variant #1

Instrument Cluster
AM/FM Radio
Amplifier
Blu-ray Player
RSE Display

Head Unit
Intelligent Antenna

AGL Linux
App Framework
- Apps
- Diag
- PM
- ALSA V4L cdev

MOST® LINUX DRIVER
(1^C/MediaLB/1^S/SPI/USB)
Connection to INIC Network Controller
Master node variant #2

- Instrument Cluster
- Amplifier
- AM/FM Radio
- Blu-ray Player
- Head Unit
- Intelligent Antenna
- RSE Display

App Framework
- MSMM
- Apps
- Diag
- PM
- ALSA
- V4L cdev

MOST® LINUX DRIVER

Connection to INIC Network Controller

(I2C/MediaLB/I2S/SPI/USB)
Slave node variant #1

Instrument Cluster
AM/FM Radio
Amplifier
Head Unit
Intelligent Antenna
Blu-ray Player
RSE Display

AGL Linux
App Framework
- Apps
- Diag
- PM
- ALSA V4L cdev

MOST® LINUX DRIVER
Connection to INIC Network Controller

(I²C/MediaLB/I²S/SPI/USB)
Slave node variant #2
NW functionality/interfaces

<table>
<thead>
<tr>
<th>System Setup</th>
<th>Device Control</th>
<th>Streaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetworkMaster</td>
<td>MOST App FBlocks</td>
<td>ALSA</td>
</tr>
<tr>
<td>ConnectionMaster</td>
<td></td>
<td>Video4Linux</td>
</tr>
<tr>
<td>MOST App</td>
<td>Ethernet Apps</td>
<td>cdev</td>
</tr>
<tr>
<td>FBlocks</td>
<td></td>
<td>App HW</td>
</tr>
<tr>
<td>PowerMgmt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagnosis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AGL/MOST – pros/cons

- Existing devices can still be used
- MLD available
- MNS/MSMM, MNS or MNS Basic Binding needs to be developed
- Classic MOST® apps need to be adapted to Binding
- Binding might be app-specific
Summary
**Summary – Linux/MOST®**

- If you go Linux you can:
  - Take advantage of the MLD
  - Keep MOST as a network
  - Keep single MOST devices unchanged
  - Transfer existing MOST apps to Linux
Summary – AGL/UNICENS

• If you go AGL with UNICENS you can:
  • Take advantage of the MLD
  • Implement a new cost effective network
  • Use the available AGL UNICENS binding
  • Easily configure the network
  • Interface applications via Json
  • Easily interface Ethernet applications
Summary – AGL/MOST®

• If you go AGL with MOST you
  • Can keep MOST as a network
  • Can keep single MOST devices unchanged
  • Can take advantage of the MLD
  • Need to develop a Binding that interfaces MOST applications to the AGL framework
  • Choose from different Binding approaches
Thank you!
The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud, ChipKIT, ChipKIT Logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.


SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2018, Microchip Technology Incorporated, All Rights Reserved.