Requirements for a software API to cover automotive video applications

Martin Miller
Corporate Overview

- **Leading provider of:**
  - High-performance microcontrollers, digital signal controllers and microprocessors
  - Mixed-signal, analog, interface and security solutions
  - Clock and timing solutions
  - Wireless and wired connectivity solutions
  - Non-volatile EEPROM and Flash memory solutions
  - Flash IP solutions

- ~$3.5 billion revenue run rate
- ~13,000 employees
- Headquartered near Phoenix in Chandler, AZ
Video in Automotive

- Head-unit
- Cluster / HUD
- ADAS ECU
- Rear-seat entertainment
Infotainment Apps

- Watching TV
- Navigation on all displays
- Playing movies
- Integration of mobile devices
Camera Apps

- Rear-view camera
- Top-view
- Passenger observation
- Position detection
- Digital mirror
- Park assistant
- Break assistant
- Drive assistant
- Autonomous driving
Formats

- Uncompressed
  - Bayer RGB
  - YUV
  - RGB
  - ...

- Compressed
  - MJPEG
  - H.264
  - H.265
  - ...
Clock Domains

- **Multiple clocks in the system**
  - Tuner, LTE modem, mobile device, ...

- **Synchronization**
  - **Sources**
    - Multiple cameras stitched together to one top-view image
    - Stereo-camera to measure obstacle distance
    - ...
  - **Sinks**
    - Multiple displays showing the same stream
    - Video synchronized to cabin sound
    - ...
Safety Aspects

- Pack all data for transmission
  - Clocking
  - Presentation time
  - CRCs
  - ...

- Do not show old pictures in safety critical applications
Legal/Commercial Aspects

- Support country-specific solutions
  - DVB-T
  - ATSC
  - ISDB
  - DTMB
  - …

- Use as few codecs as possible in each device to keep license cost low

- Support content protection
Content Protection

Content-owner required to protect their streams when transmitted inside car

<table>
<thead>
<tr>
<th>Medium</th>
<th>Consortium</th>
<th>Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVD</td>
<td>DTLA(5C)</td>
<td>DTCP</td>
</tr>
<tr>
<td>Blu-Ray</td>
<td>DTLA(5C) / DCP</td>
<td>DTCP / HDCP</td>
</tr>
<tr>
<td>DVB-X, …</td>
<td>CI+</td>
<td>DTCP / HDCP</td>
</tr>
<tr>
<td>Miracast</td>
<td>DCP</td>
<td>HDCP</td>
</tr>
</tbody>
</table>

Consider also new streaming services
Organizational Aspects

- Different groups provide different content parts
- Only one department is responsible for final front end

Tuner Department

GUI Department

Navi Department
Programming Aspects

- Support different languages like C, C++, Java, HTML5, Qt, …

- Small footprint to support small embedded systems

- No, or only little, dependencies

- During runtime independent from state machines
Example: Display Rear-View Camera

Function:
- Receive
- Demux
- Decode
- Rear-View App
- GUI App
- Render

Process:
- Framework
- App1
- App 2
- Framework
Example: Watching TV

- **Antenna**
  - Receive
  - Demux
  - Decode
  - Overlay
  - Encode
  - Encode
  - Send

- **Head-unit**
  - Receive
  - Demux
  - Decode
  - Embed in GUI
  - Render
  - Send PCM

- **Amplifier**
  - Receive
  - Mix
  - Render

- **Rear-seat**
  - Receive
  - Demux
  - Decode
  - Embed in GUI
  - Render
Functionalities

- **Source**
  - File
  - cdev
  - Capture
  - Socket
  - Pipe
  - Shared mem
  - ...

- **De-/multiplex**
- **Decode**
- **Encode**
- **En-/decrypt**
- **Fork**
- **Re-timer**
- **Overlay**
- **…**

- **Convert**
- **Sink**
  - Render
  - File
  - cdev
  - Socket
  - Pipe
  - Shared mem
  - …
Data Container

- **TS Packet**
- **Timestamps**
  - PTS
  - PCR
- **PES Packet**
  - H.264
  - MJPEG
  - ...
  - MP3
  - AC3
  - DTS

- **Raw**
  - RGB
  - YUV
  - ...
  - PCM
  - PDM
  - ...
Already Available Functionalities

- GStreamer
- OpenMax
- libva
- Ffmpeg™

- Unfortunately none covers all requirements listed before
Microchip developed major parts of a framework for networked video applications:

- Rear-view camera
- Head-unit delivering navigation picture
- Cluster display device
- Rear-seat-entertainment
- Cruise recorder
Example: Display

- Receive a TS from a cdev
- Recover time (PCR)
- Peel off transport layer to get elementary stream
- Decode video
- Render according to PCR

Application: Create, connect and initialize objects

Select file → SrcCdevTs → FltForkTs → FltTs2Es → FltH264DecodeYuv → SnkRender

Select PID → SnkPcrSystemTime

Set window position
Example: Camera

- Capture camera picture
- Encode to H.264
- Multiplex into a transport stream
- Send to network

Application: Create, connect and initialize objects

Set parameters
- Width
- Height
- Framerate

Select PIDs

Set file name

SrcCapture → FltEncode → FltEs2Ts → SnkCdev
Example:
Video-On-Demand

Application: Create terminal, select & position in files

Streamer

Terminal

SrcFileTs → FltReTimer → FltFork → FltTs2Es → FltTs2Es → FltEs2Ts

Terminal

SrcFileTs → FltReTimer → FltFork → FltTs2Es → FltTs2Es → FltEs2Ts

SrcMultiplexerTs

SrcProgramTs → FltEs2Ts

SrcProgramTs → FltEs2Ts

SnkCdev
Questions?
Thanks!