Leveraging Linux® to Create an Auto Infotainment Platform

Sridharan Subramanian
Senior Product Manager – Software and Platforms
This session describes the requirements and architecture of creating an auto infotainment solution based on Linux® OS. What are the right components both at the kernel and middleware layer. It also gives details on the Multimedia Automotive Reference Software (MARS) that Freescale is developing on the i.MX platform. Participants will gain insight into the capabilities of the platform that can serve as a foundation for their automotive infotainment systems.
Automotive Telematics and Infotainment
Automotive Telematics includes:

- In-car navigation and guidance systems
- Call center services
- Car/cell phone integration
- Fleet management systems
- Satellite radio
- Web services
- Automotive diagnosis and data collection
- Vehicle tracking
- In-car entertainment
- Future integrations after leaving assembly line
Traditional Infotainment Platforms

- QNX® Neutrino Realtime Operating System
- Microsoft® Auto and Windows® Automotive (CE)
- WindRiver VxWorks®
- RTOS Systems such as MARS Based on Flex OS
Requirements for Telematics Systems

► Determinism
  • Worst case interrupt response times below 50 microseconds
  • Worst case context switch times below 100 microseconds
  • 60 milliseconds worst case boot time for certain applications

► Support of Industry Standards
  • POSIX (threads, file-I/O)

► Support of Broad Range of Communication Protocols
  • Automotive: CAN, MOST, etc.
  • USB 2.0, IEEE1394™, 801.11g, Bluetooth® stack, TCP/IP etc.

► Small Footprint

► Power Management
Key Feature Requirements for Auto Infotainment

► Fast Boot

► Audio Connectivity and Telematics
  • Compressed audio playback from storage devices (CD, USB, HDD or SD card)
  • PlayFromDevice (1-wire and 2-wire support) for portable media players
  • iPod/iPhone control and playback
  • High-speed CD ripping (encode) to USB, SD/MMC or HDD for virtual CD changer
  • Audio processing for hands-free telephony: Bluetooth, AEC/NS, microphone beam forming, etc.
  • Speech recognition

► A/V Connectivity and Navigation
  • Features above plus
  • Map display and route calculation
  • Video decode and high resolution displays
  • Sophisticated graphical user interface
Linux® in Automotive Infotainment
Why Linux®?

► The Linux® Advantage

• Low-cost solution versus using proprietary SW
• Technical merits: on-demand device driver loading, communication protocols, broad range of applications, etc.
• With an open source community reviewing source code, the platform is more reliable
• Independence: with open source licenses you "own" the source code (you are not the copyright holder)

► Success of Linux in Telematics is Hampered by

• Technical concerns: real-time determinism, boot times, power management
• Business issues: intellectual property risks, technology support
## Linux® Software Stack

<table>
<thead>
<tr>
<th>HMI</th>
<th>Windows</th>
<th>Skins</th>
<th>Fonts</th>
<th>Sounds</th>
<th>Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application layer</strong></td>
<td>Apps framework</td>
<td>Player</td>
<td>Navigation</td>
<td>Mobile Office</td>
<td>Search</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Launcher</td>
<td>PIM</td>
<td>Browser</td>
<td>Java™</td>
</tr>
<tr>
<td><strong>Middleware layer</strong></td>
<td>Media Framework</td>
<td>Network Connectivity</td>
<td>Device Connectivity</td>
<td>Graphics Libraries</td>
<td>Segment Specific Libs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Core services / infrastructure DBUS, UDEV, GSM, GPS, etc.</td>
<td>Power Management</td>
</tr>
<tr>
<td><strong>OS layer</strong></td>
<td>BSP</td>
<td>SoC Drivers</td>
<td>Drivers for Connectivity, PM, etc.</td>
<td>Accelerated Codecs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bootloader</td>
<td>Kernel</td>
<td>Core Libraries</td>
<td></td>
</tr>
<tr>
<td><strong>Hardware</strong></td>
<td>Board and Peripherals</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2009.
Hardware

► Each SoC needs to be defined for a target segment. Some of the customizations include:
  • Variations in core speed
  • Multimedia and graphics capability
  • Cost of end device/platform
  • Automotive qualification
  • Industrial specifications

► Peripherals vary resulting in different board configurations
  • Device connectivity like BT, USB
  • Network connectivity like WiFi, Ethernet
  • Display variations – size and type
  • Storage variations – NOR, NAND, SD/MMC, etc.
  • Memory type – mDDR, DDR2, etc.
OS Layer

► Bootloader
  • Uboot, redboot, etc.
  • Need specific modifications for boot time

► Kernel
  • Optimizations to enable more product-ready pieces

► Core Libraries
  • GNU C, libstd c++, compression, base files, core utils, core daemons, PM, ALSA, busybox, Sysvinit, Threads, Sync, memory management

► SoC Drivers
  • V4l, Framebuffer, I2C, SSI, Filesystem, RTC, etc.

► Drivers for Connectivity (BT, WiFi, USB) and PM

► Accelerated Codecs Drivers
Middleware Layer

► Core Services/Infrastructure
  • Dbus, GPS, GSM, Udev, Obex, Samba, etc.

► Network Connectivity
  • TCP/IP, WiFi, BT, RTP/RTSP (Streaming)

► Device Connectivity
  • USB, BT, device discovery, content indexing

► Graphics and Media
  • Codecs; media framework, such as Gstreamer; UI toolkits, such as GTK, Pango, Cairo, ATK; windowing systems such as X, Matchbox

► Power Management Specific Middleware

► Security
  • DRM, Cert Mgmt, SSL

► Market Specific Libs
  • GPS, Java™ VM, OTA, MOST, CAN, TAPI, messaging, speech recognition, TTS, etc.
Application Layer

► Entertainment Suite
  • Media player, image viewer, camera, video/image editing, audio control, recorder

► Personal Information Management (PIM)
  • Email, calendar, contacts, tasks

► Web browser

► Search engine

► Telephony and messaging – MMS, SMS, IM

► Launcher, installer

► Widgets, Java™ applications, navigation, TTS/speech recognition applications

► Open source applications frameworks, such as Hildon
HMI Layer

- **HMI Manager**
  - Events, touch, keyboard

- **Fonts**
  - Freetype, downloadable custom implementations

- **Sound**
  - Audio effects

- **Open Source HMI Layers Based on Toolkits like GTK+ and Gnome**
Linux® Solution Reusability

► Linux® kernel provides the ability for scalability across multiple segments, resulting in considerable reuse across the software stack.

► A layered approach with the right selection of components would enable companies to easily provide a common Linux solution that decreases investment in resources.
  • There is a perception that considerable investment needs to be done for specific Linux SDKs since the requirements are vastly different.
  • Optimally, there would be a common SDK that enables customers to easily migrate across SoCs and penetrate target markets.
Reuse of Applications Frameworks

► Various Applications Frameworks Present
  • GNOME mobile platform (http://www.gnome.org/mobile)
  • Ubuntu™ mobile platform (http://www.ubuntu.com/products/mobile)
  • Android™ platform (http://code.google.com/android)
  • Qtopia™
  • LiMo foundation

► Vast commonality across the frameworks. Gnome mobile platform presents a good base reusable framework for different target segments.

► Pros:
  • Widespread adoption and other frameworks have been built on top of it
  • GTK GUI widget
  • Gstreamer MM framework
  • Several existing applications
i.MX Linux® Platforms in Auto Infotainment
i.MX Applications Processors

Multimedia:
Convergence of Audio, Video and Connectivity

► Primary Applications
  • Media players
  • Navigation devices
  • Automotive infotainment
  • General embedded

► Performance, Low Power and Portability
  • Optimized performance per MHz
  • Low-power leadership
  • Range of audio and video formats, graphics and connectivity options
  • On-chip accelerators optimize performance and battery life
  • Supports Microsoft® Windows® CE and Linux® OS

Video | Graphics | Security | Audio | Connectivity | Low Power

Portable Consumer

Automotive

Industrial

Enterprise
i.MX – Complementary Markets

Mobile Consumer
- Anytime, anywhere access to content & information
- High quality audio & multimedia
- Small form factor & low power

Automotive Infotainment
- Audio, video and navigation
- Free-to-air and subscription content models
- Mobile consumer connectivity

Auto grade
Extended temp
Connectivity peripherals

General Embedded
- High performance ARM processors
- Broad OS and ecosystem support
- Connectivity peripherals

Industrial quality
Product longevity
Larger pitch packaging

Industrial
- Point of sale terminals
- Handheld printers/scanners
- IP cameras

ARM™ processors
Broad OS and ecosystem Multimedia

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2009.
i.MX Processor Launches

**ARM™ Cortex A8**
- **i.MX27 family**
  - Cortex-A8, 800-1GHz
  - 720p Decode/ D1 Encode
  - OpenGL ES 2.0 / VG 1.1
  - C65

**ARM11™**
- **i.MX31 family**
  - ARM1136, 532MHz
  - VGA Encode
  - OpenGL ES 1.1
  - C90

**ARM9™**
- **i.MX32 family**
  - ARM926, 400MHz
  - D1 Video Enc/Dec

**ARM11™**
- **i.MX51 family**
  - ARM1176, 532MHz
  - HW D1 Decode
  - C90

**ARM11™**
- **i.MX2x family**
  - ARM926, 400MHz
  - Security
  - C90

**i.MX37 family**
- ARM1176, 532MHz
- HW D1 Decode
- C90

**i.MX25 family**
- ARM926, 400MHz
- Open VG 1.1
- C90

Left Edge = early samples

Platform Shipping
Samples Available
Features

- Out-of-the-box integrated Linux environment – tools + kernel + drivers
- Standards based
- Extensively tested, hardened and validated
- Optimized for target platforms
- Accelerated Codecs support
- Common code base across different i.MX SoCs

Packages

- Boot loader binaries and source files
- Patches which add Freescale drivers to a kernel.org kernel
- Source and patches for a root file system
- Source code for unit tests of the drivers
- Linux Target Image Builder (LTIB)
- Proprietary third-party components in binary code format
- Prebuilt binaries
- Open source tool chain for ARM9™/ARM11™/ARM12™
- BSP documentation (reference manual, user’s guide, release notes)
i.MX Optimized Multimedia Codecs

► Key Features
- Comprehensive suite of optimized codecs (~40+ audio/video/image codecs)
- Highly optimized software that is coded by Freescale processor experts
- Consistent application programming interface (API) and frameworks across all software packages including OpenMAX support
- Codec APIs have been optimized from system design perspective and achieve optimal system performance along with related middleware wrappers
- Supplemented with Freescale development tools, sample test streams and documentation

► Codec Software Packages Include:
- Codec libraries with a standard C-callable API
- GStreamer plugins that provide an API layer between the multimedia framework and the codec library
- Audio/video file containers (parsers) that support popular multimedia content, such as .aac, .avi, .asf, .mp3 and .mp4 files
- Bundle of Freescale audio/video sample test streams
- Complete documentation, including API documentation, release notes and data sheets
Multimedia Automotive Reference Software (MARS)
Multimedia Automotive Reference Software Key Features

► Boot Time Reduction

► Boot Time Critical Services Architecture (BTCS)

► Power Management

► Small Footprint Embedded GUI with MARS (Demo purposes – FB based)

► Media and Connectivity Features
Boot Time Reduction

► Requirement: Splash screen within 1 sec.; User space execution 3-12 sec.
► Problem: Linux typically takes 10-20 sec. to boot into a GUI
► Solution: Speeding up system boot duration by
  • Optimizing bootloader
    ▪ enable caches
    ▪ limited environment variables block size
    ▪ skip network and IDE detection
  • Optimizing kernel
    ▪ limit kernel to minimum needs
    ▪ compile drivers used at a later point as modules
  • Optimizing filesystem
    ▪ re-order driver, services and GUI initialization
Boot Time Reduction Architecture

Traditional boot sequence

1. Board initialization
   - scan IDE
   - setup PHY

2. Kernel decomposition

3. Boot kernel
   - scan IDE
   - setup PHY

4. Start services

5. Start GUI

Optimized boot sequence

1. Board initialization

2. Kernel decomposition

3. Boot kernel

4. Start GUI

5. Start services

6. Scan IDE

7. Setup PHY
Boot Time Critical Services Architecture

► Requirement: The target needs to respond to incoming CAN messages not later than 60 ms after power on.

► Problem: It takes about 800 ms before the kernel can perform user space applications (kernel threads: 600 ms).

► Solution: Implement a mechanism that allows certain tasks to be performed in parallel to the booting or running bootloader and kernel.
Boot Time Critical Services Architecture

► A “Boot Time Critical Service” (BTCS) is setup very early during the boot process.

► The BTCS code and data is located at a reserved memory area not to conflict with bootloader and kernel.

► The BTCS is implemented as callback function that can be invoked from either the bootloader or the kernel.

► The BTCS can signal the bootloader or the kernel through a “backcallback” (ossignal) function.
MARS Middleware Platform Features

► Audio/Video Playback of Compressed and Uncompressed Content:
  • MP3, MP2, AAC, MP4, WMA/ASF, SBC

► Audio Post Processing
  • Predefined and custom graphic equalizers, digital volume control, sample rate conversion

► Multichannel Audio Playback
  • Multiple play sessions with independent play controls

► Image Decode of Compressed and Uncompressed Image Content
  • JPEG, BMP, simultaneous audio playback in the background

► Line-in
  • Playback from line-in, recording of audio input stream from line-in (mp3)

► Device Connection Support
  • SD card, USB {Mass storage, MTP, iPod, Zune}, audio CD {CDM-M10 via I2C/I2S}

► Play Control
  • Play/Pause/Stop, FFWD/FRWD, Repeat, Shuffle, ntrscan
GENIVI and High-Tier Auto Infotainment Stack
Additional Infotainment Software Pieces

▸ Third party partnerships for advanced proprietary features
  • Speech recognition
  • Text to speech
  • Bluetooth® and AEC
  • MOST netservices
  • Media database
  • CAN
  • Flash

▸ Partner with full stack providers or system integrators
  • To provide missing middleware pieces
  • To indemnify and warranty software
Freescale GENIVI Status

► Freescale is an active member of the GENIVI Alliance
► Freescale is participating in GENIVI working groups – core member of the multimedia and systems infrastructure groups
► Freescale will collaborate with OS/Software providers and Tier1’s to complete a GENIVI platform based on i.MX processors

**i.MX Auto infotainment**

Ubuntu/Debian subset + MARS + Genivi

<table>
<thead>
<tr>
<th>Window Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Media Player</td>
</tr>
<tr>
<td>Rhythmbox Audio</td>
</tr>
<tr>
<td>Firefox Browser</td>
</tr>
<tr>
<td>Image Viewer</td>
</tr>
<tr>
<td>Flash &gt;= v9</td>
</tr>
<tr>
<td>Navigation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Audio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech Recognition *</td>
</tr>
<tr>
<td>Navigation *</td>
</tr>
<tr>
<td>Bluetooth *</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trio A/V Connectivity Engine</td>
</tr>
<tr>
<td>AEC/NS *</td>
</tr>
</tbody>
</table>
Freescale Introduces Product Longevity Program

► The embedded market needs long-term product support, which allows OEMs to provide assurance to their customers.
► Freescale has a longstanding track record of providing long-term production support for our products.
► Freescale is pleased to introduce a formal product longevity program for the market segments we serve.
  • For the automotive and medical segments, Freescale will manufacture select devices for a minimum period of 15 years.
  • For all other market segments in which Freescale participates, Freescale will manufacture select devices for a minimum period of 10 years.
► A list of applicable Freescale products is available at www.freescale.com.
Thank you for attending this presentation. We’ll now take a few moments to review the audience questions, and then we’ll begin the question and answer session.