“Linux Powered Instrument Clusters: The Road Less Travelled”

By
Ashish Chavan
Sunil Beta Baskar
Prashant Deshpande
Contributors

Ashish Chavan
12 years of experience in Linux systems architecture and development. Active contributor to ALSA & PMIC drivers in mainline kernel

Sunil Beta Baskar
11 years experience, Solution architecture and Open Source evangelist

Prashant Deshpande
AVP & Practice leader for Instrument Cluster at KPIT Cummins Infosystems Ltd, India
The Road Less Travelled...

Linux Powered Instrument Clusters

- Linux for Automotive Cluster
- KPIT Experience in Linux for Cluster
- Summary
Agenda

1. Linux in Automotive
2. Innovations in Automotive Instrument Clusters
3. Linux for Instrument Cluster vs. Infotainment
4. Embedded OS market share
5. Evolution of Linux for Automotive Systems
6. Linux for Clusters - Benefits & Challenges
7. What an Instrument Cluster Needs?
8. Strategies in Adopting Linux for Cluster
9. Opportunities in Adopting Linux
What is common?

- AUDI - Automobile Simulator
- Cadillac CUE - Instrument Cluster
- Asteroid (Android for Automotive)
- FORD’s Smart Gauge cluster
- GENIVI compliant IVI systems

Linux Community already identified and addressed various core technical adoption problems
Linux in Automotive

- Increasing demand for Platform based software development
- There are multiple OS choices with varied personalities
- Software & OS Platforms enabling features have become a competitive advantage to OEMs
- Automotive industry to make their OS choice more objectively without prejudice

Linux is already running inside many cars, and it may play a much larger role soon, too!

Linux Powered Instrument Clusters: The Road Less Travelled
New trends in Instrument Clusters

- Fully reconfigurable technology
- Augmented Displays
- High-performance 2D / 3D rendering
- Theme based clusters
- Navigation
- Driver assist features
- Telematics
- Diagnostics & Remote assistance
- Clustertainment

Implications on S/W development

- Increasing Software code size
- Increasing Test Cases/ Use Cases
- Requirement Complexity Management

Next-Gen Instrument Cluster cost

- 40% Software
- 60% Others

Linux Powered Instrument Clusters: The Road Less Travelled
### Linux for Instrument Cluster vs. Infotainment

<table>
<thead>
<tr>
<th>Instrument Cluster</th>
<th>In-Vehicle Infotainment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for Real-time Applications through RTAI/POSIX.RT</td>
<td>Interactive Applications with no Real Time Barriers</td>
</tr>
<tr>
<td>Requires higher reliability with lesser modularity</td>
<td>High Modularity and Configurability supports scalable IVI platforms</td>
</tr>
<tr>
<td>Instrument Clusters need to deal with limited use cases</td>
<td>Linux can deal with a multitude of use cases</td>
</tr>
<tr>
<td>Needs extensive standardization &amp; compliances</td>
<td>Needs to support multiple features &amp; has to be flexible on standards</td>
</tr>
<tr>
<td>User interaction is extremely minimal</td>
<td>GENIVI is promoting Linux for IVI</td>
</tr>
<tr>
<td>Who will be promoting and taking this forward?</td>
<td></td>
</tr>
</tbody>
</table>
Embedded RTOS Market Share vs. Linux

Linux is leading

© KPIT Cummins Infosystems Ltd.

Source: EE Times Study (2013)
Evolution of Linux for Auto Industry Acceptance

Linux is the most scalable OS, GENIVI alternative

TIZEN, AGL, GENIVI, Asteroid/Android are Linux based solutions

Available for small footprint devices to high-end servers. Automotive in between

AGL collaboration to lead in promoting Linux in Instrument cluster programs

OEMs leveraging Linux based solutions (TOYOTA, NISSAN, JLR, GM, VW,...)
**Typical Challenges for Instrument Cluster**

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real time performance</td>
<td>• CAN/LIN/Flexray/OSEK; bus drivers, data acquisition</td>
</tr>
<tr>
<td>FAST Boot &amp; response</td>
<td>• HMI display with high responsiveness</td>
</tr>
<tr>
<td></td>
<td>• Data from the bus to be available instantaneously</td>
</tr>
<tr>
<td></td>
<td>• Low latency</td>
</tr>
<tr>
<td>Connectivity</td>
<td>• Diagnostics, connected car, digital life in vehicle</td>
</tr>
<tr>
<td></td>
<td>• Proven CAN stack</td>
</tr>
<tr>
<td>Security aspects</td>
<td>• Secure hardware / software</td>
</tr>
<tr>
<td>Compliance, Standardization &amp; legal aspects</td>
<td>• Standardized Test Framework, kernel distribution</td>
</tr>
<tr>
<td></td>
<td>• Code compliance</td>
</tr>
<tr>
<td></td>
<td>• Intellectual property related clarity</td>
</tr>
<tr>
<td>Support &amp; Long term Evolution</td>
<td>• Dynamic &amp; Rapid changes by Linux community</td>
</tr>
<tr>
<td></td>
<td>• Ecosystem of tools to realize real performance</td>
</tr>
<tr>
<td></td>
<td>• Professional support</td>
</tr>
<tr>
<td>Parameter</td>
<td>Benefits</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>Real-time Performance</td>
<td>Available in the Linux Kernel</td>
</tr>
<tr>
<td>CAN Data Capture without loss</td>
<td>CAN Data bus available as a low level protocol driver</td>
</tr>
<tr>
<td>Energy Saving</td>
<td>Power Management Framework in Linux Ecosystem is comprehensively Tested</td>
</tr>
<tr>
<td>Intellectual Property</td>
<td>Community Driven, Collaborative, open source licensed, No Vendor Lock-in</td>
</tr>
</tbody>
</table>
Strategies on Adopting Linux for Cluster

1. Adopt Dual OS Architecture
2. Use Linux with RTOS as Low Priority Thread
3. Full Adoption

Benefits of Switching to Linux:
- Extensibility
- Faster
- Share Cost
- Better Support
Opportunities to Address Challenges

- **Technical Challenges**
  - Real time performance of the BSP/Stack
  - Boot Time Optimization for the BSP/Stack

- **Techno-commercial Challenges**
  - Optimizing Time to Market by Standardization
  - Compliance to Safety standards including
    - MISRA-C 2012
    - MISRA-C++ 2012
    - ISO 26262

- **Business Challenges**
  - Handling Intellectual property issues by adhering to Licensing models and ensuring continuing collaboration
KPIT’s Linux based Reconfigurable Cluster Experiment

Objectives, Observations, Achievements
<table>
<thead>
<tr>
<th></th>
<th>Agenda Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Case study Objectives</td>
</tr>
<tr>
<td>2</td>
<td>How did we Target Linux for the Instrument Cluster</td>
</tr>
<tr>
<td>3</td>
<td>KPIT’s Linux based Cluster features</td>
</tr>
<tr>
<td>4</td>
<td>What is the Architecture we created?</td>
</tr>
<tr>
<td>5</td>
<td>Challenges we Identified</td>
</tr>
<tr>
<td>6</td>
<td>Strategies for Boot Time Optimization</td>
</tr>
</tbody>
</table>
Casestudy Objectives

Evaluate whether Linux is a suitable platform for Instrument Clusters of the Future Connected Cars

Identify the challenges to address for wider adoption of Linux in Instrument Clusters

Design an Architecture to address identified challenges

Synchronize what we learnt and migrate to AGL with our experience

Why KPIT?

- KPIT’s GNU/Linux Passion
- Inspired by our successful GENIVI compliant K-IVI platform
Case Study: How did we target Linux for the Instrument Cluster?

- Freescale's i.MX53 based Automotive Quick Start Board
- Linux Kernel 2.6.34
- Also on i.MX6

We Selected Qt/Embedded 4.8.2 as the middleware/HMI stack

Our Solutions derived from

- File Systems derived from
  - KPIT’s IVI solution on Linux
  - GENIVI compliant Linux User-Space stack

We built

- Modular Software Architecture
- The HMI Layer
- Automotive Bus Data Layer (supporting CAN 2.0)
KPIT’s Linux based Digital Instrument Cluster

Features Integrated in our Effort

<table>
<thead>
<tr>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenGL ES 2.0 Support @50fps</td>
</tr>
<tr>
<td>Modular software architecture decoupling data &amp; HMI</td>
</tr>
<tr>
<td>Reconfigurable HMI design with multiple themes</td>
</tr>
<tr>
<td>Support for generic HID based input</td>
</tr>
<tr>
<td>ALSA2 Audio Support for Sound Alerts</td>
</tr>
</tbody>
</table>

What KPIT’s experiment created

- Linux BSP based Modernist Software Architecture for Clusters
- Fast prototyping with Qt/Embedded
- Design partnership
- Platform Specific Optimization
- Modular Architecture Decoupling Data/Business Logic from HMI Design
- Deployment support for Multiple Targets
Case Study: Our Architecture for the Linux-based Instrument Cluster

- **Reconfiguration and Software Upgrade**
- **Cluster App**
- **OEM reconfigurable user-interface (XML, scripts)**

- **LED diag.**
- **CAN handler**
- **GUI Framework / State Machine**
  - Event Handlers
  - Message Handler
- **Text-to-Speech**
- **Diagnostic Shell (Headless)**

- **CAN Socks**
- **POSIX-RT**
- **Video-FB/PAE**
- **RTAI**
- **Qt/E 5**
- **Open GLES2**
- **Android VM**
- **libstdc++**
- **GPIO-lib**
- **HID**
- **IPv4 Stack**

- **KPIT Linux-RT Kernel / Core**
- **KPIT Fastboot**
- **Linux**

- **Optimized Compile**
- **Code Modification**
- **New Code Development**

- **System Libraries**

- **Legend**

© KPIT Cummins Infosystems Ltd.
Challenges we identified for Linux as Cluster

- Reducing boot time
  - for early data capture from Automotive data buses
- Improving System Responsiveness
- Selecting best fit Application Middleware
- Ensuring File-system persistence on serial NAND Flash
### Tried and Tested Strategies for BT optimization

<table>
<thead>
<tr>
<th><strong>Init Script</strong></th>
<th><strong>File system</strong></th>
<th><strong>Kernel / Bootloader</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Optimize shell scripts (Reduce undesired fork/exec)</td>
<td>• Optimize services by keeping them lean and realtime</td>
<td>• Place kernel/boot-loader in high-speed storage medium (e.g. NOR/NAND flash device)</td>
</tr>
<tr>
<td>• Remove unused or un-required services or start them later</td>
<td>• Reduce filesystem journal playback time</td>
<td>• Quiet boot, Avoid Verbose Debug Messages</td>
</tr>
<tr>
<td>• Improve startup parallelism using RC scripts (minit Make style init)</td>
<td>• Use UBIFS on UBI, rather than YAFFS2 and JFFS2</td>
<td>• Reduce probe delays (Preset information for known hardware)</td>
</tr>
<tr>
<td>• Optimizing udev triggers, &amp; RC script</td>
<td>• Reduce mount time</td>
<td>• Remove unused kernel functionality/drivers</td>
</tr>
<tr>
<td></td>
<td>• “Read-only” filesystem mounts faster</td>
<td>• Defer module initialization (init_call optimizations)</td>
</tr>
<tr>
<td></td>
<td>• Read ahead and Cache filesystems</td>
<td>• Tune kernel size preferable to boot without compression</td>
</tr>
<tr>
<td></td>
<td>• Schedule Journaling Threads only on Soft Realtime Use Cases.</td>
<td></td>
</tr>
</tbody>
</table>
KPIT Linux Fast Boot Optimizations Steps

1. Uboot
   - NOR Boot 256KB
   - Redundant CRC32 for Kernel Removed
   - Uncompressed Kernel Image

2. GNU/Linux/Kernel
   - Support VPU/IPU/GPU Loaded on Demand
   - "__init" RT_PREEMPT handler
   - "quiet" boot and "calibration delay loop" added
   - I2C/SPI/worker threads

3. GNU/Linux/System
   - "init" replaced by "cinit"
   - Ramdisk/ro to Ext3/ro
   - SD/MMC media pre-mapped blocks
   - Splashscreen

Target

- 2.4 second
- 26 second
- 78% saving

Linux Powered Instrument Clusters: The Road Less Travelled
“Linux Powered Instrument Clusters: The Road Less Travelled”
Recap, Summary, Questions ...
Future of Instrument Cluster

UX possibilities

- Augmenting senses
  - Audio, Vision, Tactile/Haptic feedback
- Converge features with IVI
- Auto chargeable smart device that can be carried by driver in pocket

System level Possibilities

- Single ECU for Cluster & Infotainment
- Detachable/ Plug & Play display
- Open collaboration on diagnostics

This is what Linux can address to make this happen sooner...
Need for an Open Standard for Automotive Diagnostics and Telemetry - to avoid Vendor Lock-in

Make Possible Quick Integration of Multiple Features to a Demanding Market

The Connected Car without disparate buses/communication systems - to assure connectivity across OEMs/Vehicle-Classes

Innovation & Development Opportunities for the community
The Road Ahead ...

- **Opportunities** for further innovations
  - Create a *Certified* version on a Stable Kernel
  - Create a *Certifiable* version/branch of mainstream Linux tree
    - Enable adoption of new features/patches from upstream changes
  - MISRA C/C++ Compliance: compiler tool-chain, kernel, BSP stack
  - Safety Standards Compliance: ISO 26262 Integration to Linux Device Driver Model

Addressing each of the above opportunities would help early adoption and lesser uncertainty
Conclusion

- Linux is the only scalable OS with community support today
- Instrument cluster is actually an IVI features subset. Linux can be tuned with right engineering efforts to make it work for Cluster.
- Roadmap of Linux will continue to address the challenges thereby enabling adaptation of Linux in Automotive Industry
- We believe, Linux WILL soon Power next generation Instrument Clusters by converging benefits of IVI
- Auto industry can safely rely on scalability, flexibility and open innovation to realize futuristic use cases
- POWER of Linux is in its open collaboration

Let’s make Linux Drive The Cluster!
Milestones we hope to see ...

- Linus Torvalds posts famous message: “Hello everybody out there…” and releases first Linux code
- Slackware becomes first widely adopted distribution
- Tech giants begin announcing platform support for Linux
- IBM runs famous Linux ad during the Super Bowl
- Linux turns 20 and powers the world’s supercomputers, stock exchanges, phones, ATMS, healthcare records, smart grids; the list goes on
- Linus licenses Linux under the GPL, an important decision that will contribute to its success in the coming years
- Linus visits aquarium, gets bit by a penguin and chooses it as Linux mascot
- Red Hat goes public
- Linux appears on the cover of BusinessWeek with a story that calls Linux as a business success
- The Linux-based Android OS outships all other smartphone OSes in the U.S. and climbs to dominance
- AGL is born and adopted for multiple automotive targets

© KPIT Cummins Infosystems Ltd.
Thank You

© KPIT Cummins Infosystems Ltd.