

Timesys University

Track Three

Building a Rich UI Based Dual Display Video Player with the Freescale i.MX53 using LinuxLink

Session 3

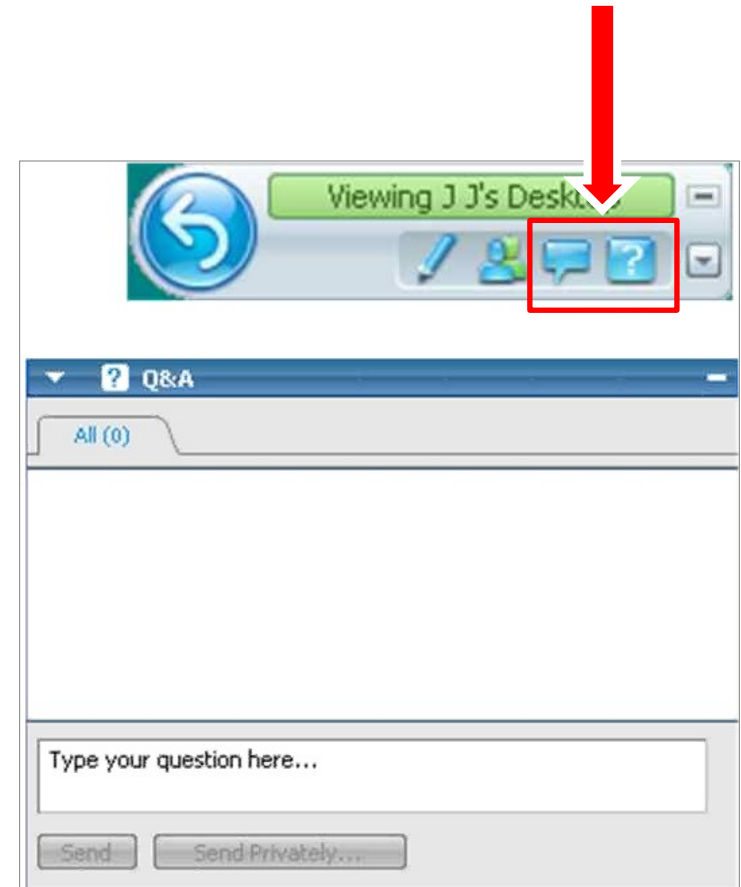
How to leverage hardware accelerated video features to play back
720p/1080p video

Audio streaming is available for this event.
Turn on your speakers to listen.

Tools You Can Use

■ Q&A and/or Chat

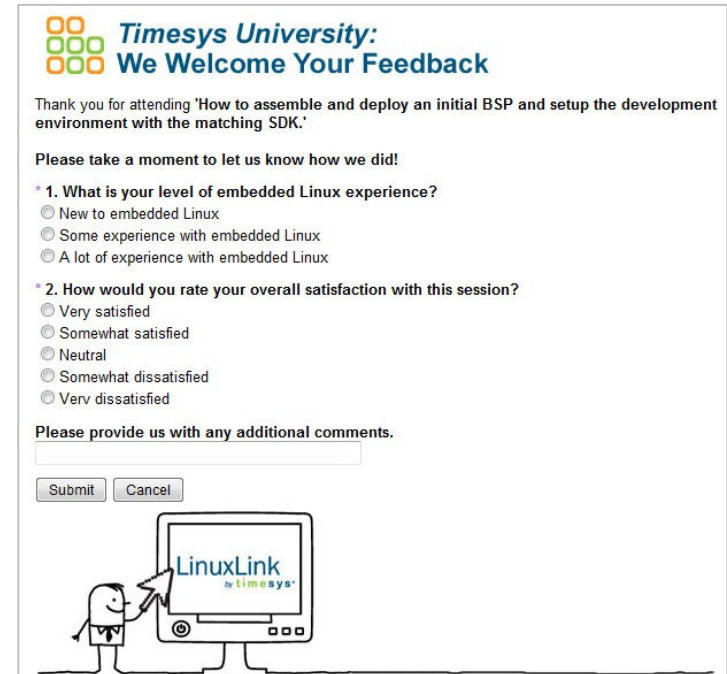
- Click on Q&A panel (?) or Chat panel icon in the bottom, right corner
- Type in your question in the space provided
- Click on “Submit”




Tools You Can Use

■ Polling

- The poll will appear on your screen
- Select your answer for each question
- Click on “Submit”



 **Timesys University:**
We Welcome Your Feedback

Thank you for attending 'How to assemble and deploy an initial BSP and setup the development environment with the matching SDK.'

Please take a moment to let us know how we did!


* 1. What is your level of embedded Linux experience?

- New to embedded Linux
- Some experience with embedded Linux
- A lot of experience with embedded Linux

* 2. How would you rate your overall satisfaction with this session?

- Very satisfied
- Somewhat satisfied
- Neutral
- Somewhat dissatisfied
- Very dissatisfied

Please provide us with any additional comments.



Session Information

- **You can download the slides for today's session at** http://www.timesys.com/embedded-linux/training/timesys-university/freescale_imx53
- **You can view a recording of today's session at** http://www.timesys.com/embedded-linux/training/timesys-university/freescale_imx53
- **Today's speaker:**



Maciej Halasz
Director, Product Management
Timesys

Building a Rich UI Based Dual Display Video Player Device

- **Session 1 – recording available**

How to capture your initial Linux based product requirements and quickly build a custom BSP/SDK to jumpstart your development

- **Session 2 – recording available**

How to build a media player control panel using Qt Embedded for Linux



- **Session 3 – Today**

How to leverage hardware accelerated video features to play back 720p/1080p video

- **Session 4 – September 7 @ 1pm EDT**

How to optimize, test and integrate the solution for fast boot and quick deployment

Today's Agenda

- **Recap of what we have done so far**
- **Freescale i.MX53 VPU overview**
- **GStreamer framework primer**
- **Add support for dual display**
- **Modify Video Player application**
 - Add code to play back video using gstreamer API
 - Add scripts to autolaunch the application at boot time
- **Configure system for dual display on single LCD**
 - Discuss other options
- **Deploy on the target board**

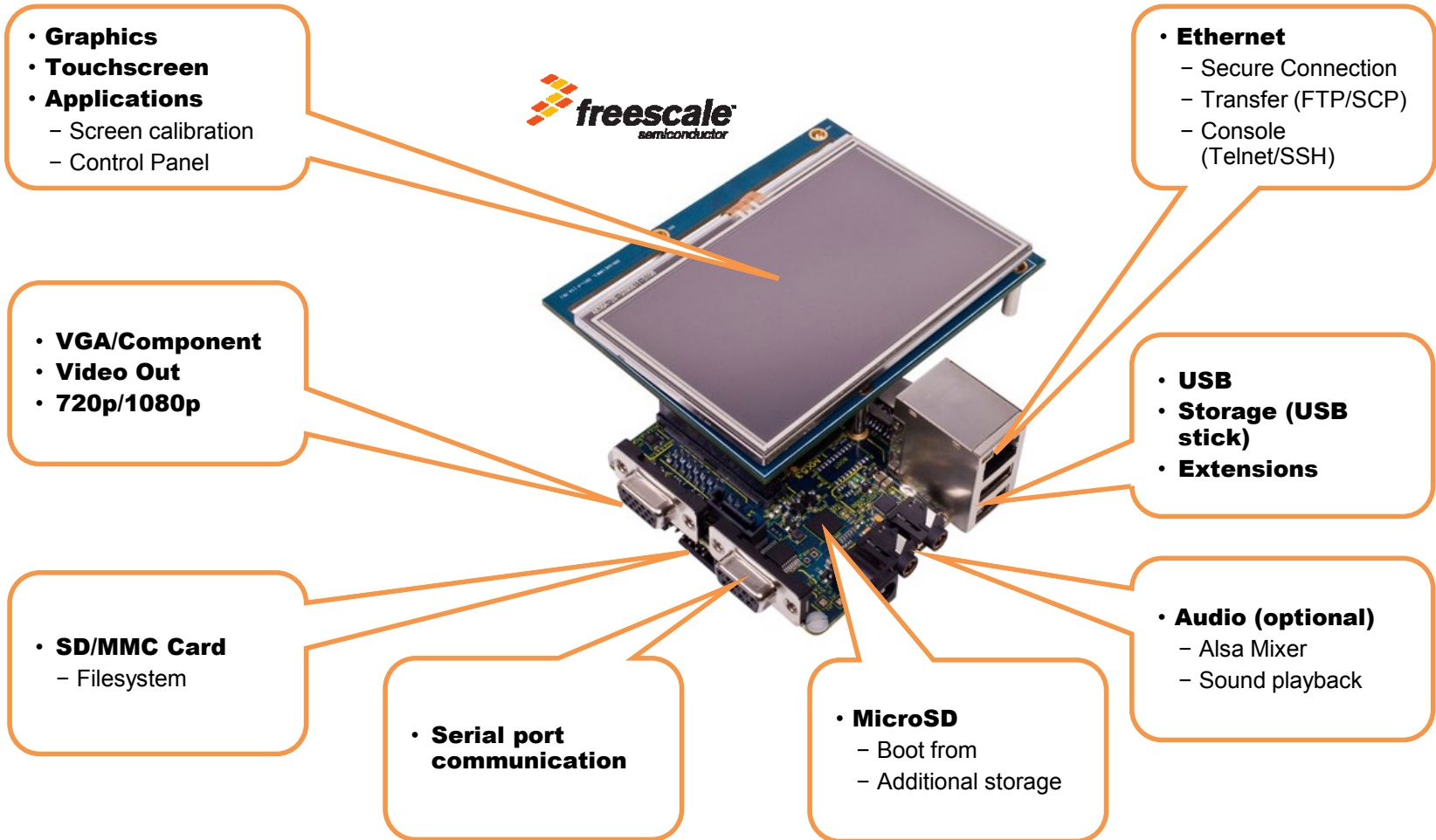
Session 1 Recap



What We Have Accomplished So Far

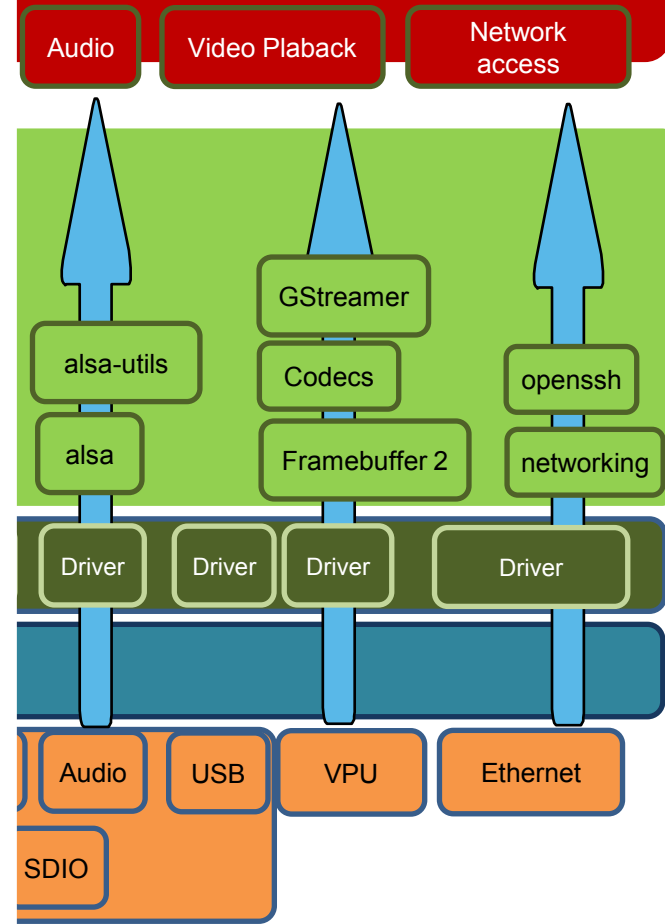
- **Learned about the Freescale i.MX53 LinuxLink and the i.MX53QSB board – needed for all exercises**
- **Reflected product requirements in Linux BSP and SDK**
- **Built a custom BSP with LinuxLink Web Edition**
 - Experiment on day one with a pre-built starting point
- **Setup a development environment (more today)**
- **Setup a Qt based IDE development environment**
 - Developed a UI with Qt widgets
 - Tested locally
- **Deployed the system on the target via NFS for future development**
 - Transferred images
 - Configured bootloader

Project Requirements (Freescale i.MX53 QSB)



Dual Display Video Player (Blueprint)

Media Playback Application

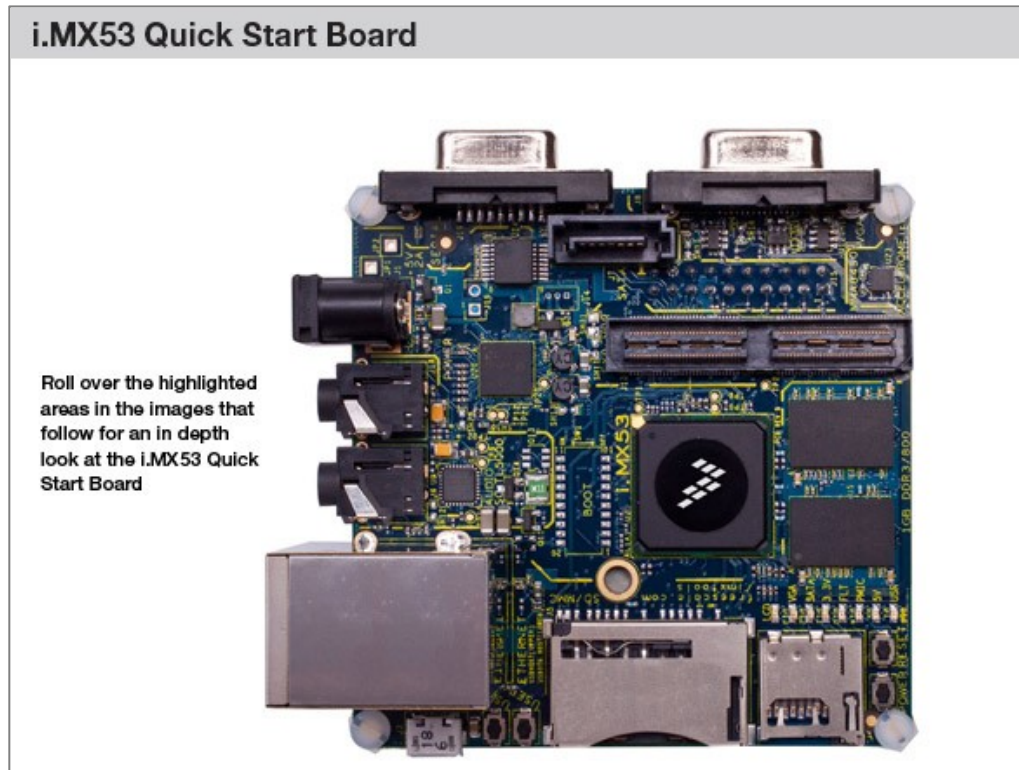


Last Session's Homework – Share Your Experience

- 1. Design your own GUI with Qt Embedded**
- 2. Adapt/design your own UI with desired set of widgets**
- 3. Create multiple windows. Connect buttons to handling routines via signals**
 - How was your design with Qt experience?
 - Did you try working with custom, third-party widgets?
 - Any questions on a signal based communication between widgets?

Giveaway

- **If you attend at least 3 out of 4 sessions in this Timesys University track, we will automatically enter you into a drawing for a chance to win a Freescale i.MX53 Quick Start Development Board**

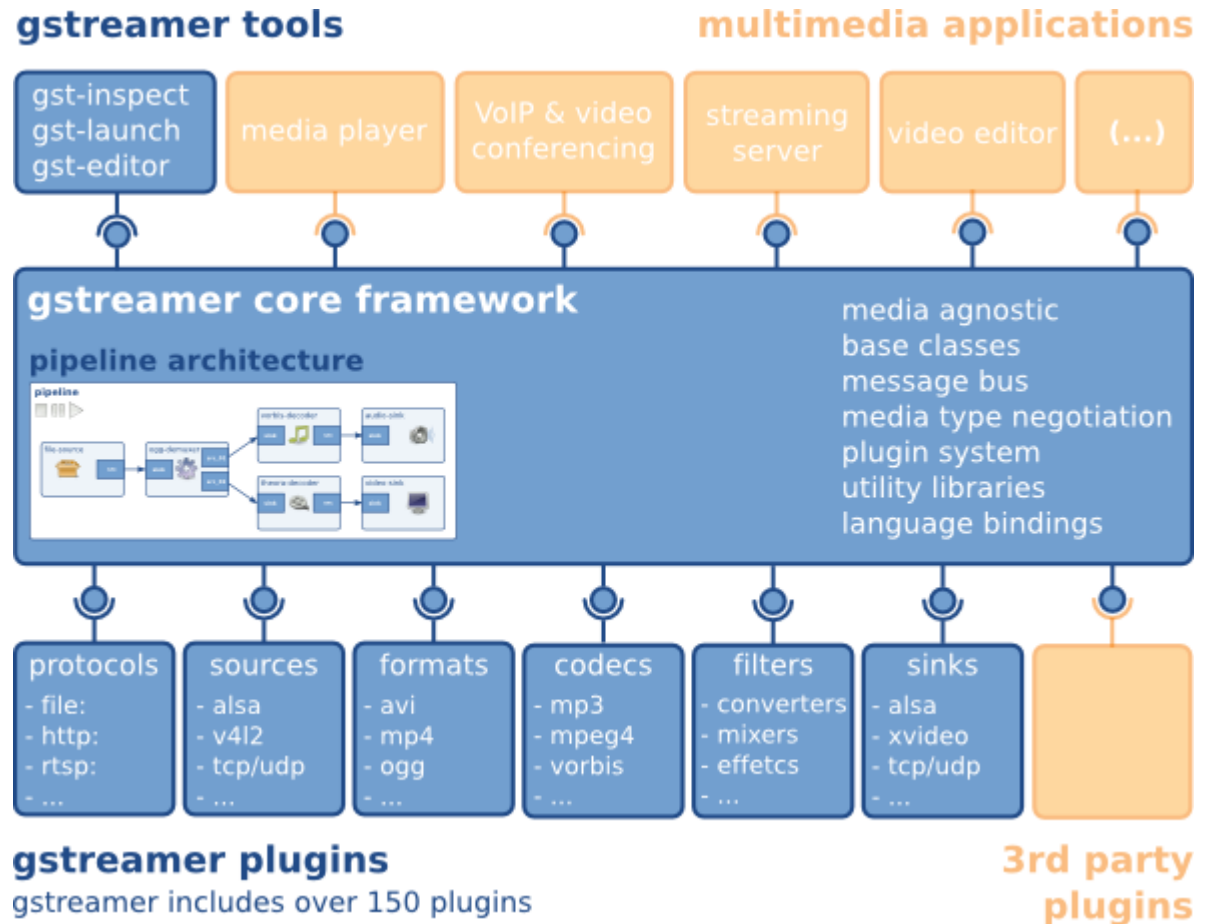


GStreamer Primer



GStreamer Ingredients

- **Video framework for “stringing together” elements**
 - Video sources, syncs, multiplexing, codecs, etc.
- **Divided into packages**
 - Baseline
 - Plugins
 - Third-party
- **Third-party plugins can take advantage of hardware specific acceleration**



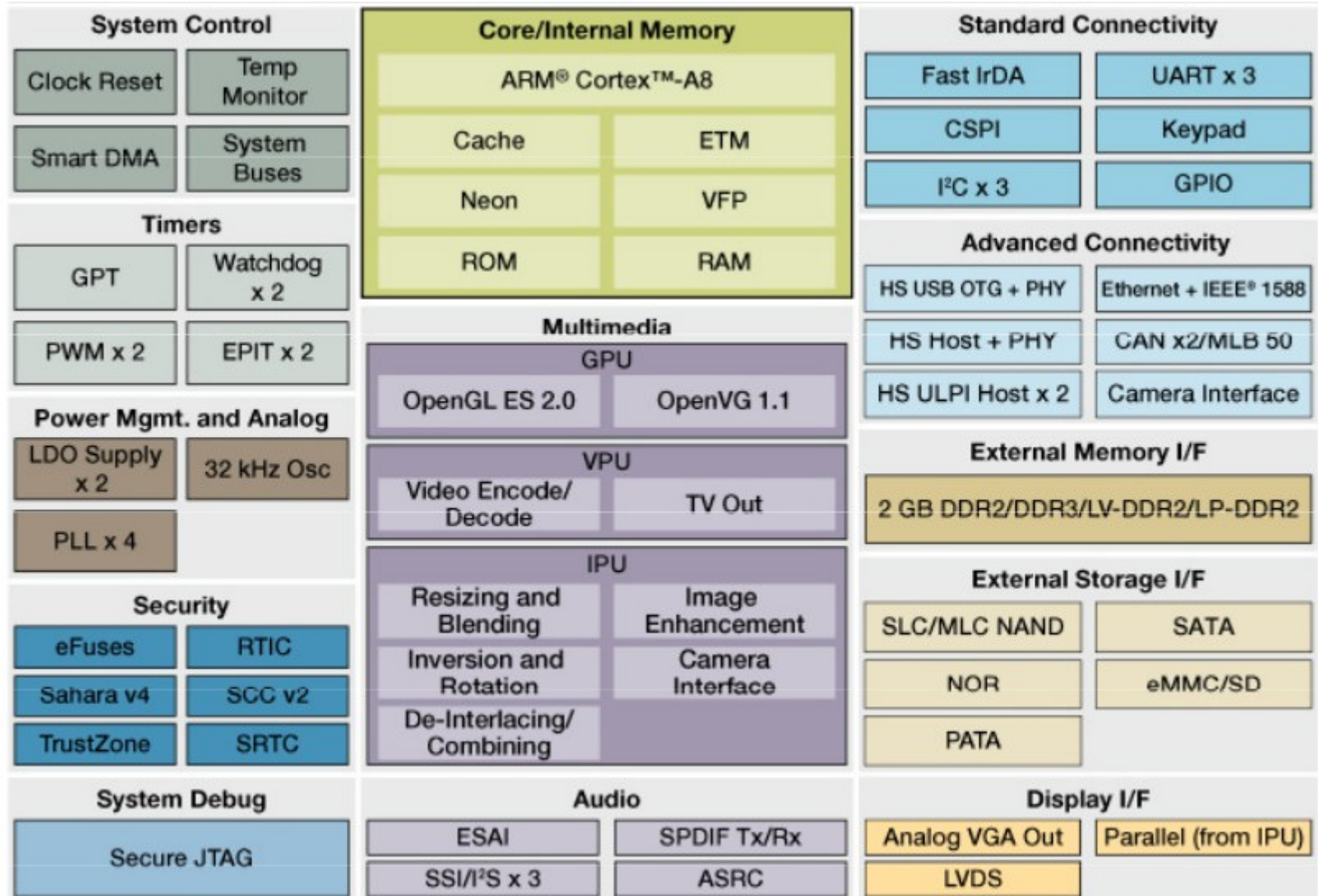
GStreamer: Linux Side

- **Delivered as a number of packages**
 - GStreamer
 - OSS Plugins
 - **Base:** Set of well-maintained plugins (coders/decoders for various formats, includes APIs for additional extensions)
 - **Good:** All tested and available codecs/formats (LGPL)
 - **Bad:** Not fully tested/verified codecs (GPLv2)
 - **Ugly:** Good quality, tested codecs (Challenge = ability to distribute / patent license)
 - **FFMpeg**
 - Freescale Plugins
- **Auto-discovery of best plugins for the job**
- **Code/Decode Audio/Video**
- **Plugins are version sensitive**
- **Can provide decoding features on non-accelerated hardware**

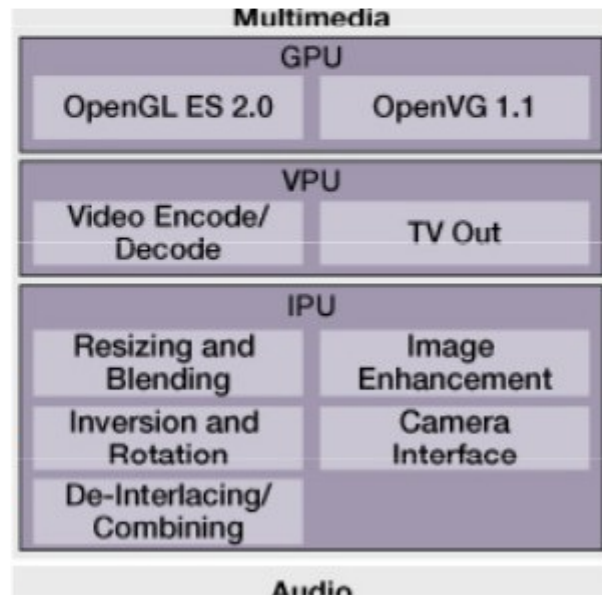
i.MX53 VPU



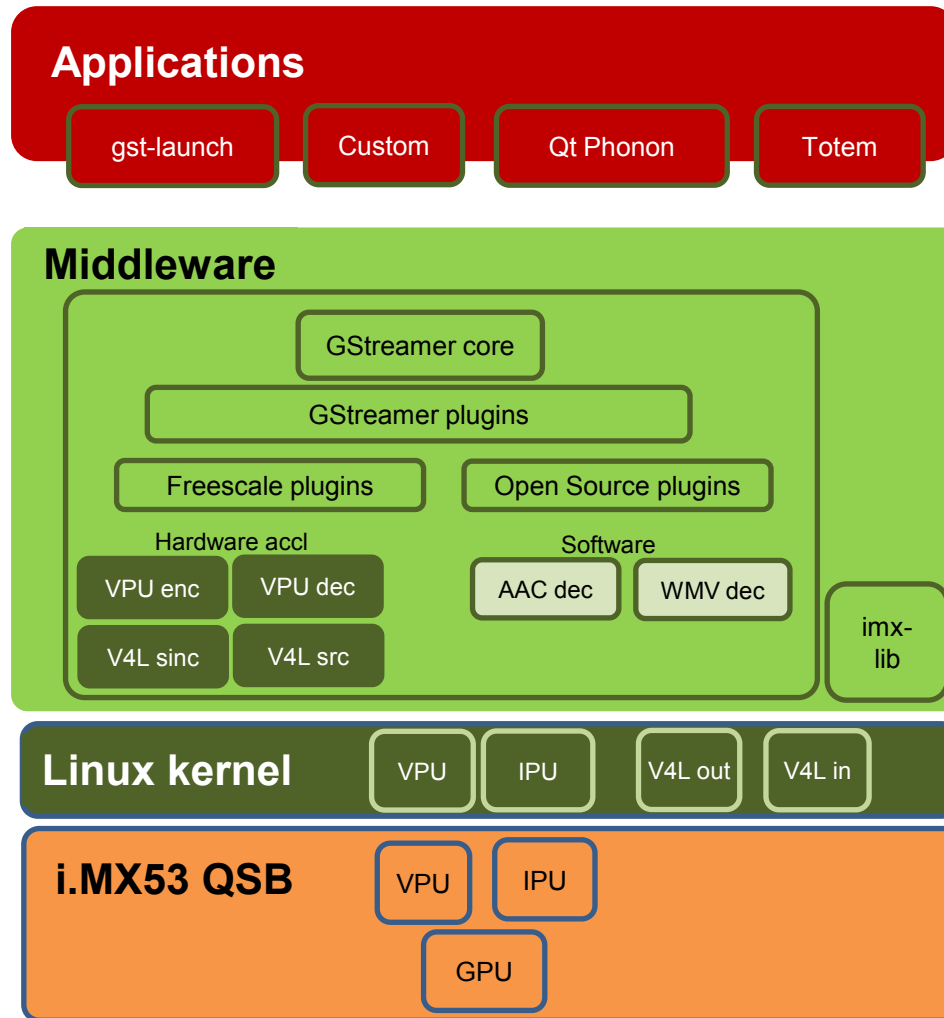
i.MX53 Hardware Block Diagram



i.MX53 Hardware Block Diagram

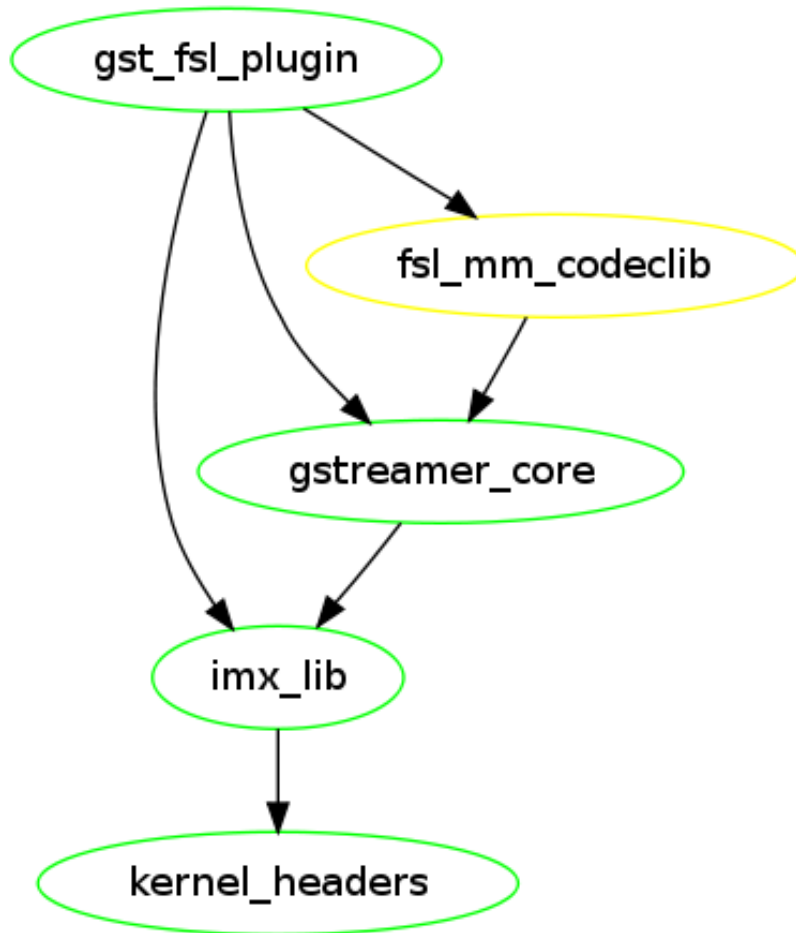


VPU + GStreamer = low CPU usage



Multimedia codecs

Package dependency



- **Includes Freescale plugins for codecs, sinks etc. used in application to build pipes**
- **Glue between codecs that talk to drivers**
- **Core functionality – modified to accelerate codecs – memory copying, etc.**
- **Wrappers for various hardware and software components, this is where software leverages VPU**
- **Headers with appropriate definitions needed by VPU**

Software Modifications



Requirements Helper

Feature	What do I need	Where to select
Linux Kernel	Latest, 2.6.35 kernel	Linux kernel
Cross Toolchain	GCC, Binutils, GDB	Toolchains
C Library	glibc	Toolchains
Touch Screen	tslib	Packages
Qt	qt-embedded-linux	Packages
Dual Display Demo	timesys-theatre-1080p	Packages
GStreamer	gstreamer gst-plugins (base, good) gst-fsl-plugin	Packages
Media Codecs	fsl-mm-codelib firmware-imx	Packages
Sound	alsa-utils	Packages
System Initialization	busybox	Packages

Linux Kernel Adjustments

- **Add functionality to support dual display**
- **Use the patch command**
- **Leverage Desktop Factory to achieve repeatable build**

Adjust kernel command line:

```
console=ttymx0,115200 video=mxcdi0fb:RGB24,SEIKO-WVGA root=b302 rw
```

Example 2

VGA Output of SXGA resolution (primary display), and SEIKO WVGA LCD Panel

```
video=mxcdi0fb:RGB24,SEIKO-WVGA video=mxcdi1fb:RGB24,VGA-SXGA  
vga di1_primary
```

Video Player Control Panel – GUI Modifications



What We Have Accomplished



What We Have Accomplished

- **Learned about hardware accelerated i.MX53 components**
- **Discussed the GStreamer framework**
- **Modified code**
 - Patched Linux kernel — added support for dual display
 - Added code to our application
- **Cross-compiled and deployed on i.MX53QSB target**



Next Session

■ September 7 @ 1pm EDT

How to optimize, test and integrate the solution for quick deployment

- How to achieve quick boot time
- U-Boot modifications
- Linux kernel optimizations
- Filesystem design changes
- SD Card deployment
- More fun to come...

Homework

1. **Work with the GStreamer framework**
2. **Setup playback pipelines from within your application**
3. **Take a look at how Desktop Factory helps with Linux kernel modifications**

Let us know if you run into any challenges. We have setup a dedicated communication channel to share questions, comments and replies.

Please subscribe at <https://lists.timesys.com/listinfo/timesys-university>

While your questions will be answered by Timesys expert engineers, we encourage you to answer questions asked by others. Sharing experiences is always the best way to learn.

Glossary

LinuxLink (Web Edition) – Web-based version of LinuxLink

LinuxLink (Desktop Edition) – Local version with full customization and third-party tools integration

Workorder – Stores definition of your software – filenames, versions

Bootloader – Runs first, initializes necessary hardware, loads Linux

Linux kernel – Operating system that manages hardware access and other features for higher level software

Device Driver – Code that's part of a Linux kernel, defines how software accesses specific hardware

File System – All files (libraries/utilities/scripts/etc.) combined on a single storage, e.g. NAND flash

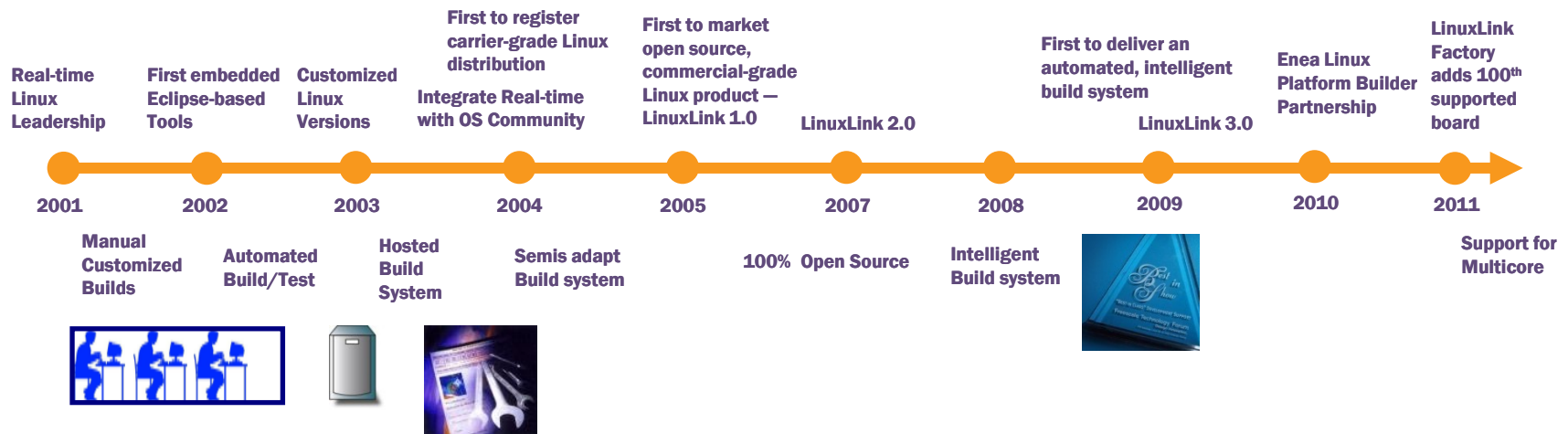
Middleware – Complete frameworks including APIs, utilities that provide specific functionality, e.g. Qt

API (library) – Used by applications, provide functionality, abstract hardware access

Toolchain (cross) – The most important part of the development environment. Used to compile source code into binaries.

About Timesys

- **Carnegie Mellon University spin-off in 1996**
- **First real-time embedded Linux distribution**
- **First to register carrier grade Linux (CGL)**
- **First to market with an open source, commercial-grade embedded Linux development framework (LinuxLink)**
- **First to develop and deliver an award-winning, automated, intelligent, embedded Linux build system (LinuxLink 3.0)**



More Info

- **You can download the slides for today's session at** http://www.timesys.com/embedded-linux/training/timesys-university/freescale_imx53
- **You can view a recording of today's session at** http://www.timesys.com/embedded-linux/training/timesys-university/freescale_imx53

Stay Online for Q&A!

