

Timesys University

Track Two

Building an Internet Radio with the TI Sitara AM3517 using LinuxLink

Session 1

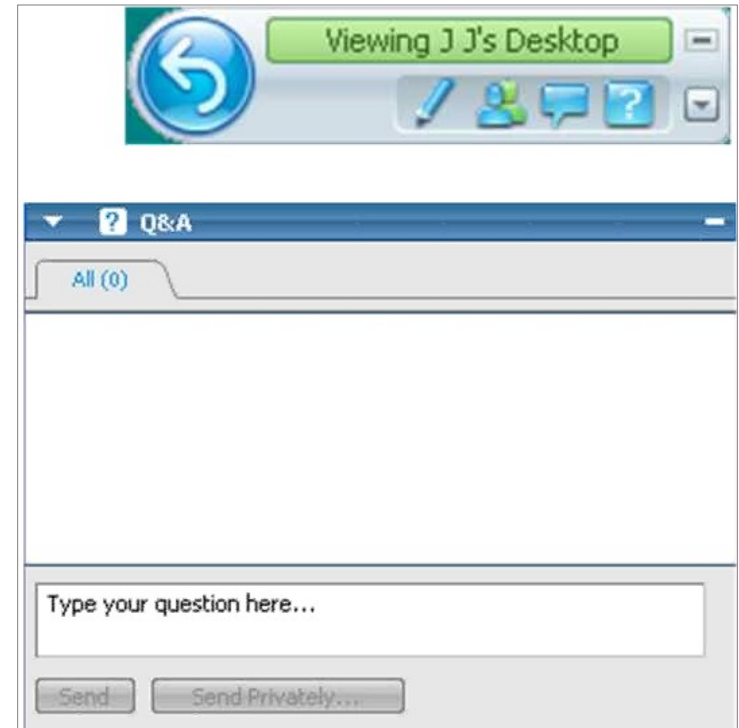
*How to assemble and deploy an initial BSP and setup the
development environment with the matching SDK*

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

Tools You Can Use

■ Q&A

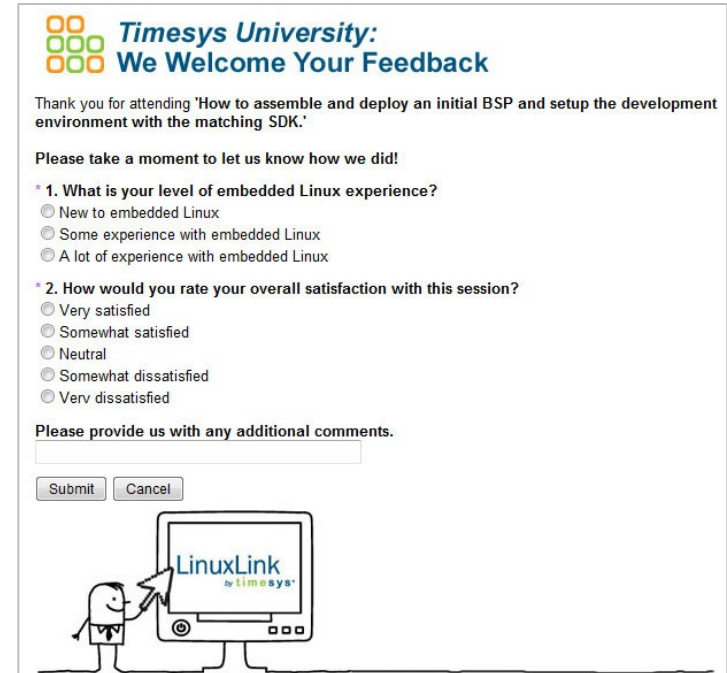
- Click on Q&A panel (?) or chat icon in the bottom right corner
- Type in your question in the space provided
- Click on “Submit”




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/ti_am3517
- **You can view a recording of today's session at** http://www.timesys.com/embedded-linux/training/timesys-university/ti_am3517
- **Today's speaker:**



Maciej Halasz
Director, Product Management
Timesys

Building an Internet Radio with the TI Sitara AM3517



- **Session 1 – Today**

How to assemble and deploy an initial BSP and setup development environment with the matching SDK

- **Session 2 – May 26 @ 1pm EDT**

How to build a modern User Interface to launch Internet Radio playback using Qt Embedded for Linux

- **Session 3 – June 13 @ 1pm EDT**

How to decode a media stream and integrate Bluetooth functionality for a remote speaker

- **Session 4 – June 27 @ 1pm EDT**

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

Today's Agenda

- **Session Hardware/Software Requirements**
- **Embedded Linux Primer**
- **Where do I start my Linux design — Product Requirements**
- **Assemble a custom BSP using the Factory Web Edition**
 - Reflect product requirements
 - Produce complete Linux design
 - Learn how to find information
- **Application/System Development environment setup with an SDK**
- **Deploy the images on the Sitara AM3517 board**

Session Hardware/Software Requirements



What We Need to Build Our Product

■ A Host Machine

- Linux
- Windows is ok, but you'll need a VirtualBox with a Linux installation

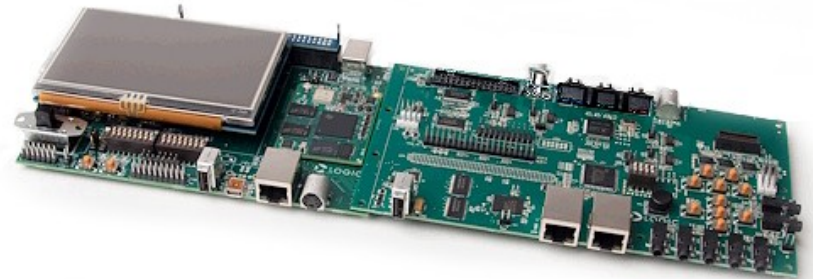
■ Cross-development environment

■ Linux source code for the product

- Bootloader
- Linux kernel
- APIs

■ Various Linux utilities

■ IDE with WYSIWYG for faster assembly



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 Logic PD Zoom AM3517EVM Development Kit**

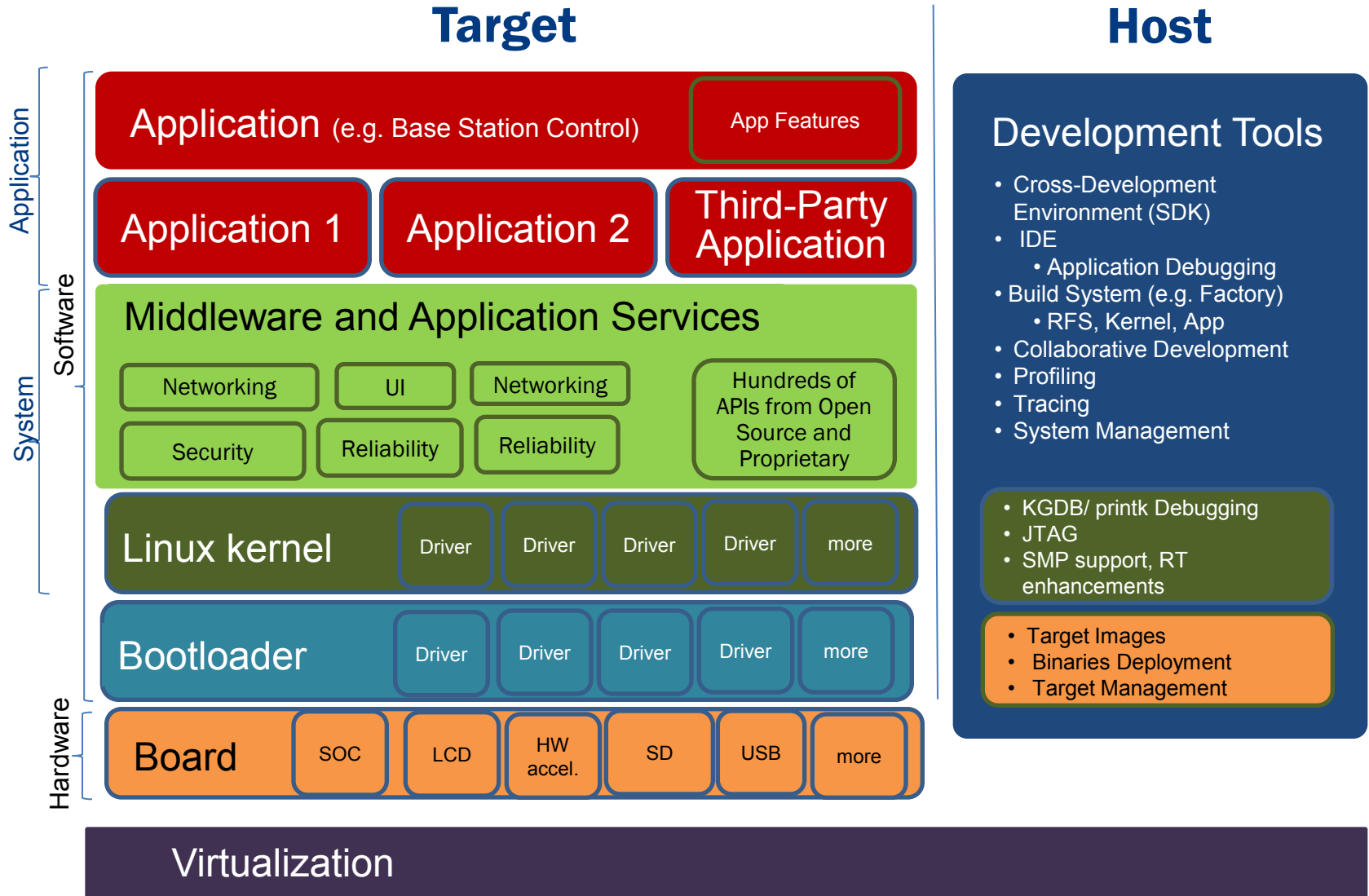


\$999 Value

Embedded Linux Primer



Embedded Linux Reference Model



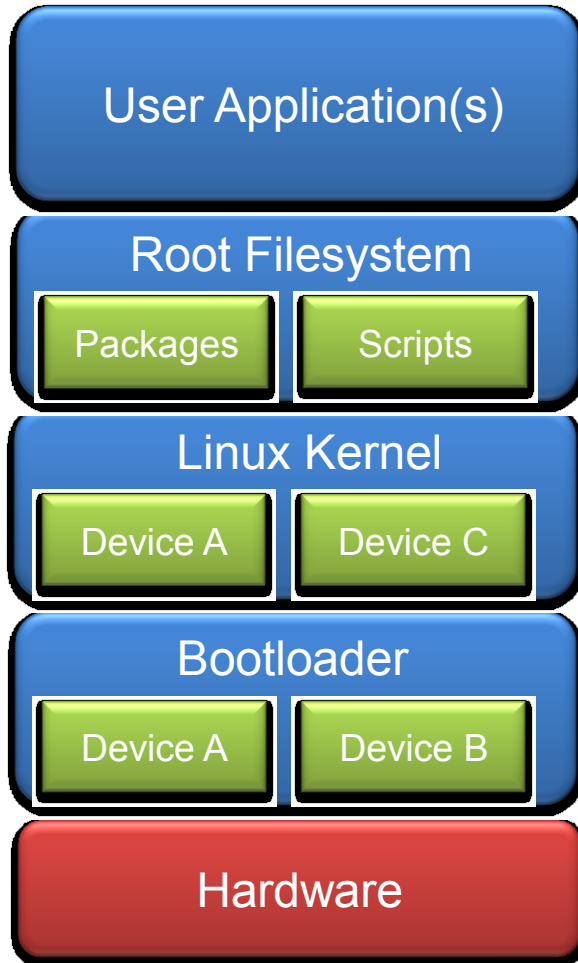
Embedded Linux Challenges

- **Assembling a Linux platform can be very complex**
 - The code is “free,” but...
 - Achieving a consistent and repeatable build can be challenging
- **Difficult to keep pace**
 - Over 40,000 independent sources on the Web
 - Maintained by thousands of developers
- **Difficult to pick the right combinations**
 - Hidden dependencies, abandoned projects
 - Numerous revision conflicts
- **Difficult to find tools that work**
 - Many open source tools are available
 - Difficult to assemble the associated patches and libraries
- **Limited-to-no support**

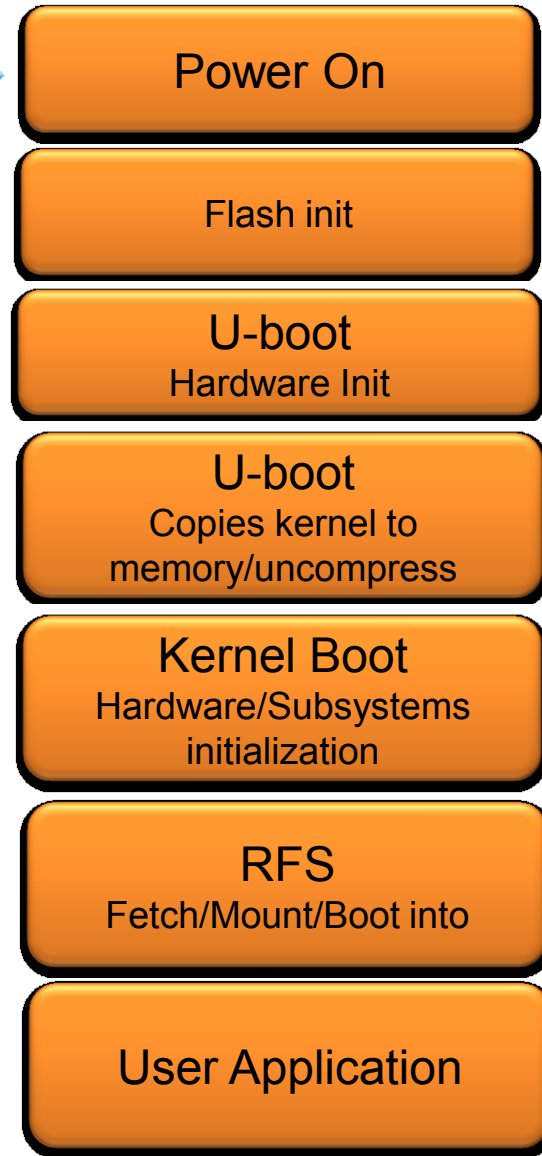


Boot Process in Short

Typical Linux System Structure



Boot Process Sequence



TIME

Product Requirements — Where Do I Start My Linux Development?



What Are We Building?

■ Streaming Internet Radio

Features:

- **GUI with the following features:**
 - Web Browser – to search for streams
 - Panel to specify/store/add/remove Radio streams
 - Playback launch capability
- **Add functionality to read RSS feeds**
- **Remote Bluetooth speaker, keyboard and mouse**
- **Network connection, preferably WiFi**
- **Boot fast from NAND flash**

Project Requirements (TI AM3517EVM)

• Serial port communication

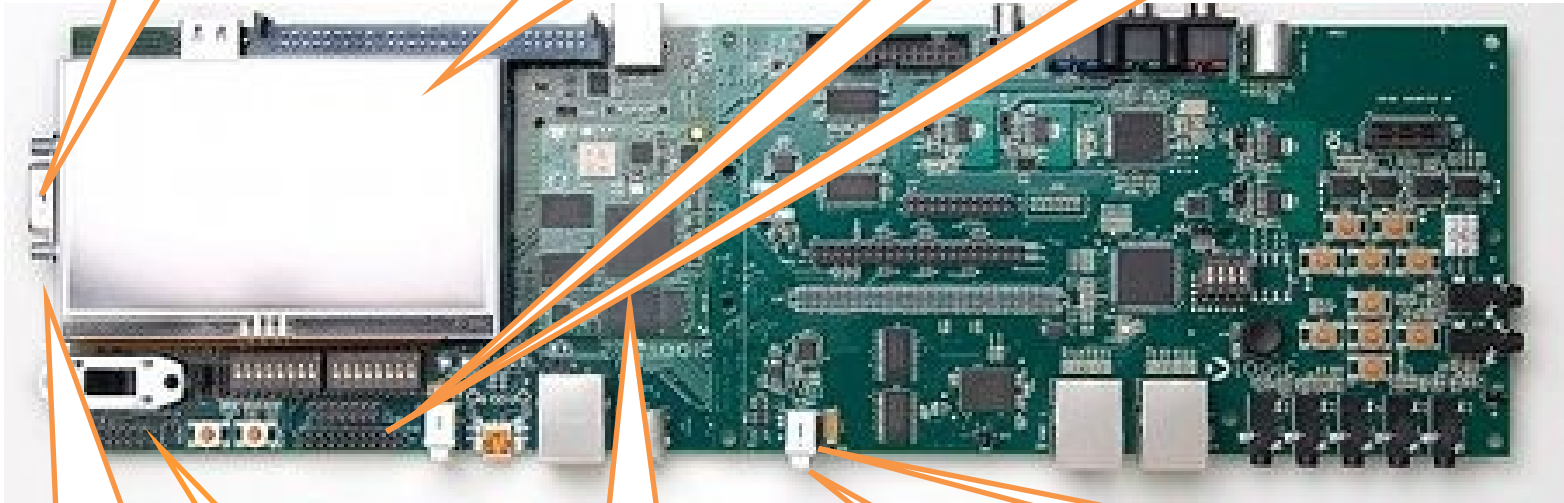
- **Graphics**
- **Touchscreen**
- **Applications**
 - Screen calibration
 - Playback control UI

• USB

- Storage (USB stick)
- Extensions

• Ethernet

- Secure Connection
- Transfer (FTP/SCP)
- Console (Telnet/SSH)



• Audio

- Alsa Mixer
- Sound playback

• SD/MMC Card

- Filesystem
- Booting Linux

• NAND Flash

- Boot from
- Additional storage

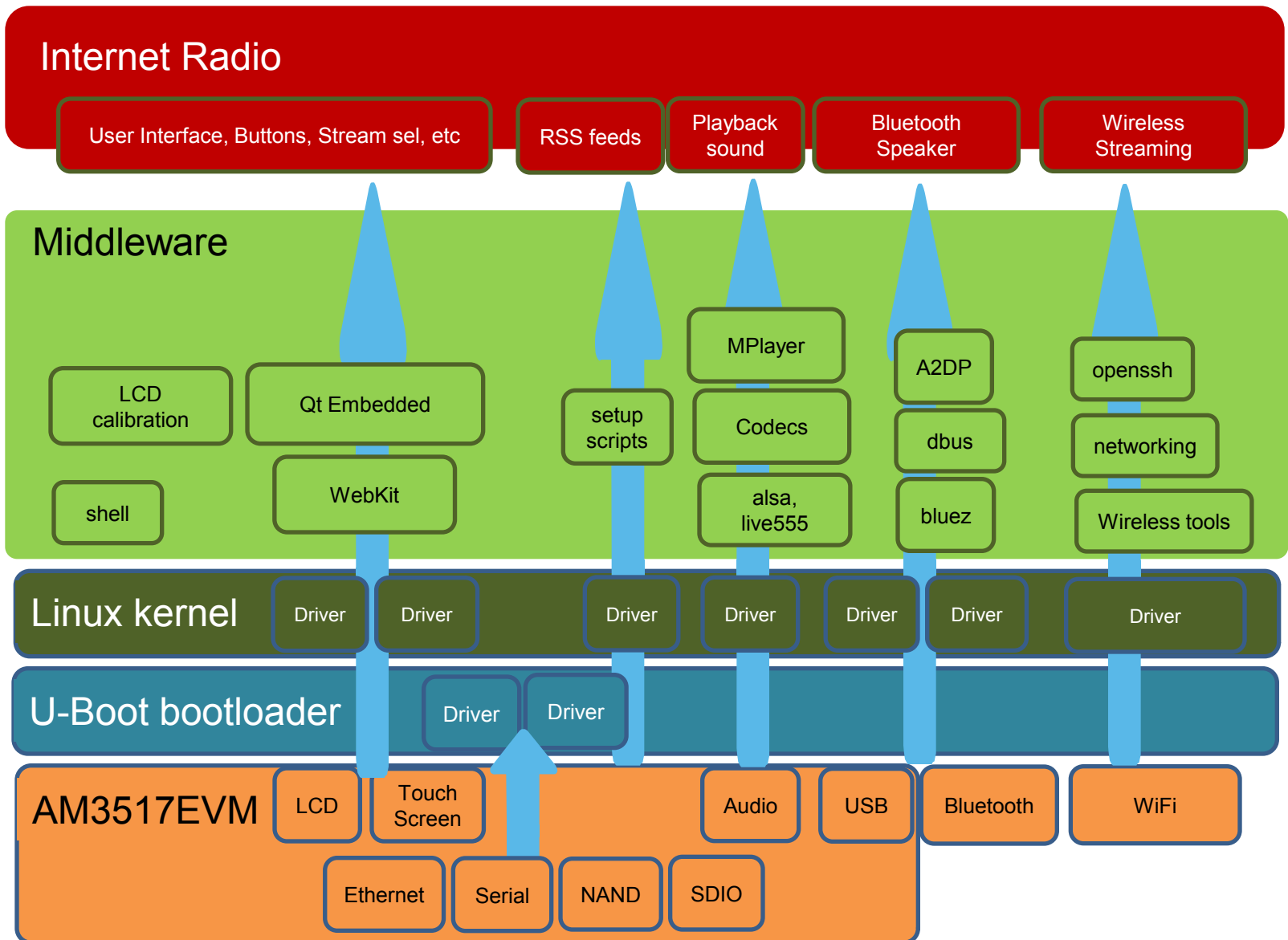
• WiFi (optional)

- Streaming audio

• Bluetooth

- Sensor connections

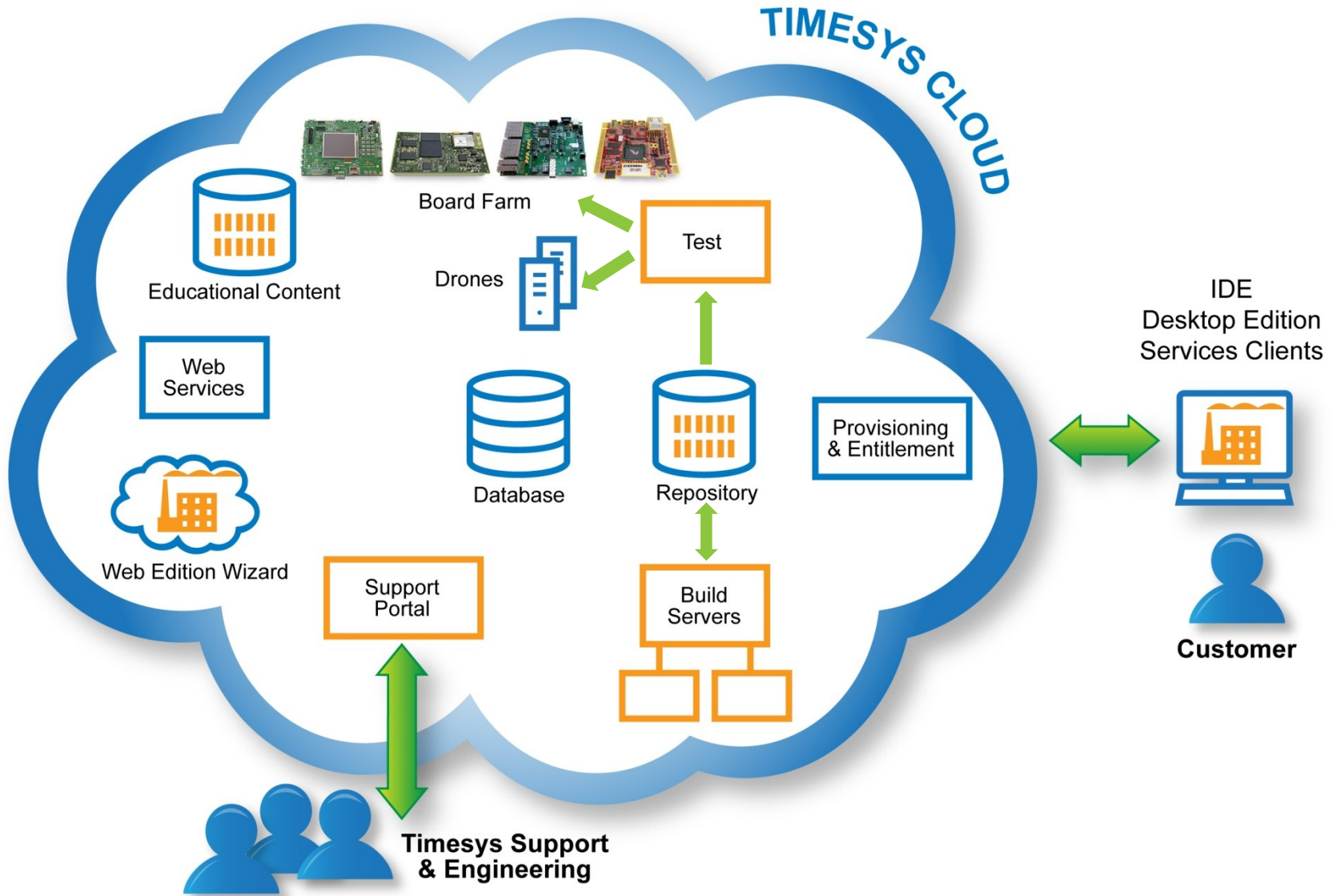
Internet Radio (Blueprint)



Assemble a Custom BSP Using the LinxuLink Web Edition



LinuxLink Architecture



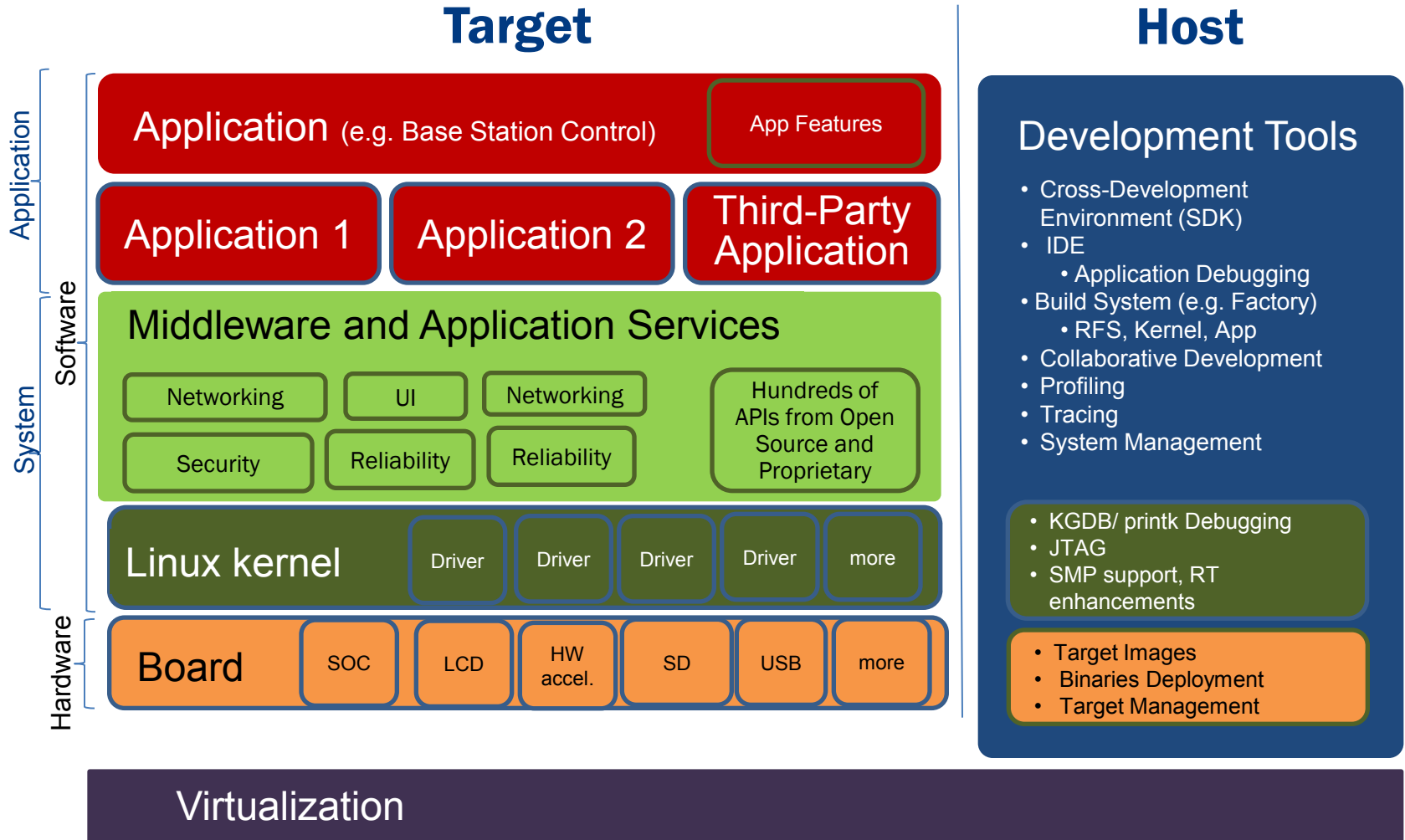
Requirements Helper

Feature	What I need	Where to select
Linux kernel	Latest, 2.6.3x kernel	Linux kernel
Cross-Toolchain	GCC, Binutils, GDB	Toolchains
C Library	uClibc	Toolchains
Touch screen	tslib	Packages
Qt	qt-embedded-linux	Packages
Bluetooth	bluez (will select dbus)	Packages
Media playback	Mplayer/GStreamer	Packages
WiFi	wireless_tools wpa_supplicant chipset dependent	Packages
Sound	alsa	Packages
Streaming	live555	Packages
System initialization	busybox	Packages

Application/System Development Environment Setup with an SDK



Embedded Linux Reference Model



Embedded Linux Reference Model

Host

Development Tools

- Cross-Development Environment (SDK)
 - IDE
 - Application Debugging
 - Build System (e.g. Factory)
 - RFS, Kernel, App
 - Collaborative Development
 - Profiling
 - Tracing
 - System Management
- KGDB/ printk Debugging
 - JTAG
 - SMP support, RT enhancements
- Target Images
 - Binaries Deployment
 - Target Management

Development Environment Setup

Build Output

To learn how to boot your target board with the build output, [read this doc.](#)

Filename	Size (kb)	Description
ulmage-2.6.32-ts-armv5l (md5)	2108	Kernel Image
rootfs.tar.gz (md5)	2536	Root Filesystem
	DIR	Bootloader
	DIR	Individual Software Packages
	DIR	Individual Software Sources
timesys-development-environment.sh (md5)	143312	SDK Installer, contains kernel, RFS, and toolchain
	56396	Toolchain Archive
factory.tar.gz (md5)	3476	Factory Build System
BUILD-SUMMARY.txt	4	Build Summary

Runtime images
Can be used to run
Linux on the target

Software Development Kit
(SDK)
• Install on your host PC

Desktop Factory tools
• Install on your host PC

Can find instructions
on how Texas
Instruments board

SDK will setup your complete Linux cross-development environment including:

- Compiler
- APIs header files and libraries

Deployment



Deployment

■ We'll use...

- U-Boot bootloader to load the Linux kernel
- NFS mounted filesystem

■ Setup entries for the NFS daemon

- Edit /etc/exports
- Add the following entry

```
home/timesys-university/session1/factory/rfs *(rw,no_root_squash)
```

- Restart the NFS daemon

```
/etc/init.d/nfs-kernel-server restart
```

■ Enable your tftp server

■ U-Boot environment variables

```
> setenv bootfile ulmage-am3517
```

```
> setenv bootargs console=ttyS2,115200n8 ip=10.0.0.10 root=/dev/nfs rw nfsroot=10.0.0.1:/home/timesys-  
university/session1/factory/rfs rw
```

```
> setenv bootcmd bootp\;bootm
```

```
> setenv loadaddr 80000000
```

```
> saveenv
```

What We Have Accomplished

- **Learned about the TI AM3517 LinuxLink – 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**
 - System level development and optimizations
 - Development of a value-add software (applications)
- **Deployed the system on the target via NFS for future development**
 - Transferred images
 - Configured bootloader

Next Session

■ May 26th @ 1pm EDT

How to build a modern User Interface to launch Internet Radio playback using Qt Embedded for Linux

- Install TimeStorm with Qt development tools
- Design a UI for the Internet Radio with features to specify, store and playback streams
- Development and testing using host environment
- Add needed code to launch a playback application
- Modify a filesystem to add support for streaming
- More fun to come...

Homework

1. **Outline your own requirements**
2. **Assemble a BSP that matches your needs**
3. **Setup your development environment**
4. **Run your custom Linux images on your hardware**

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

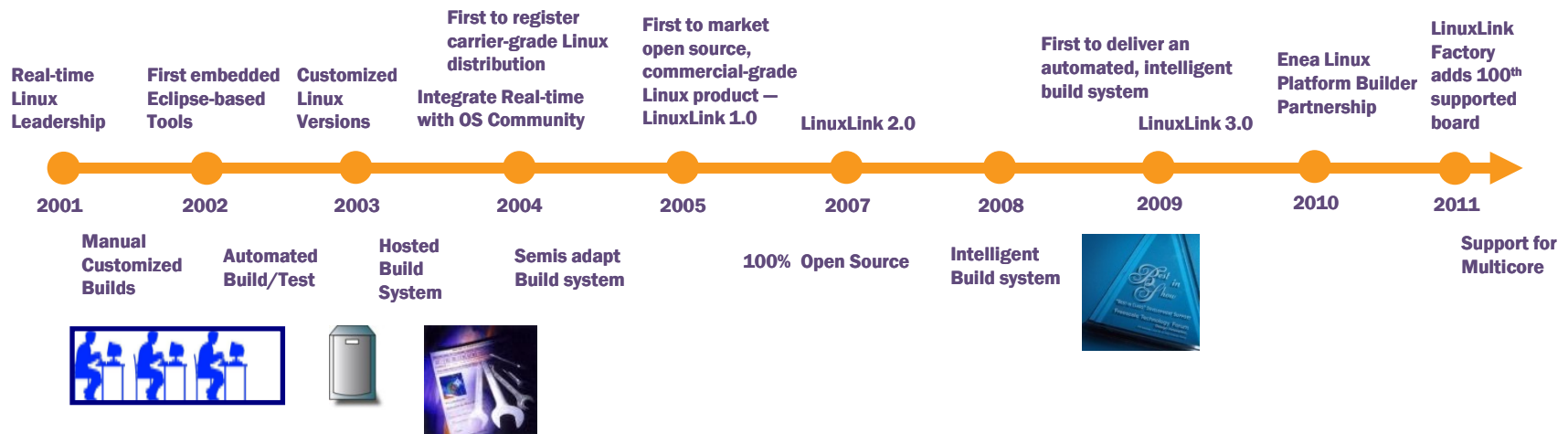
Middleware – Complete frameworks including APIs, utilities that provide specific functionality, e.g. Qt

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/ti_am3517
- **You can view a recording of today's session at** http://www.timesys.com/embedded-linux/training/timesys-university/ti_am3517

Stay Online for Q&A!

