

Timesys University

Track Two

Building an Internet Radio with the TI Sitara AM3517 using LinuxLink

Session 3

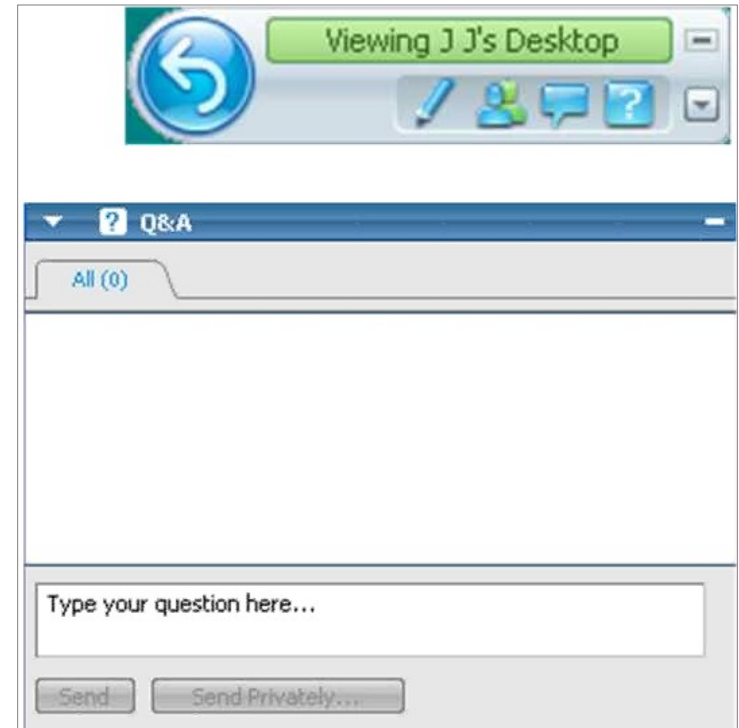
How to decode a media stream and integrate Bluetooth functionality for a remote speaker

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

Tools You Can Use

■ Q&A

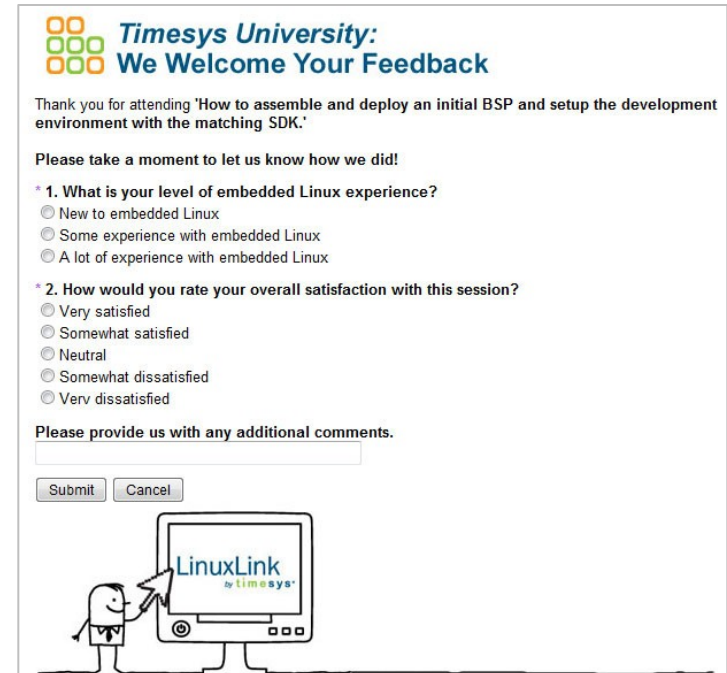
- Click on Q&A panel (?) or chat icon in the bottom right corner
- Type in your question in the space provided
- Click on “Submit”




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 – Recording Available**

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

http://www.timesys.com/embedded-linux/training/timesys-university/ti_am3517

- **Session 2 – Recording Available**

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

- **Session 3 – Today**

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

- **Session 4 – June 30 @ 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**
- **Codecs primer**
 - Filesystem integration
- **Bluetooth primer**
 - Linux kernel modifications
 - Filesystem integration (including autoboot)
 - BT Device pairing
 - BT Device setup
- **Internet Radio – GUI Bluetooth integration**
 - mplayer integration – launching another Process
 - Adding station through a web browser
- **Summary**

Session 2 Recap



What We Have Accomplished So Far

- **Learned about 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 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 (TI AM3517EVM)

• Serial port communication

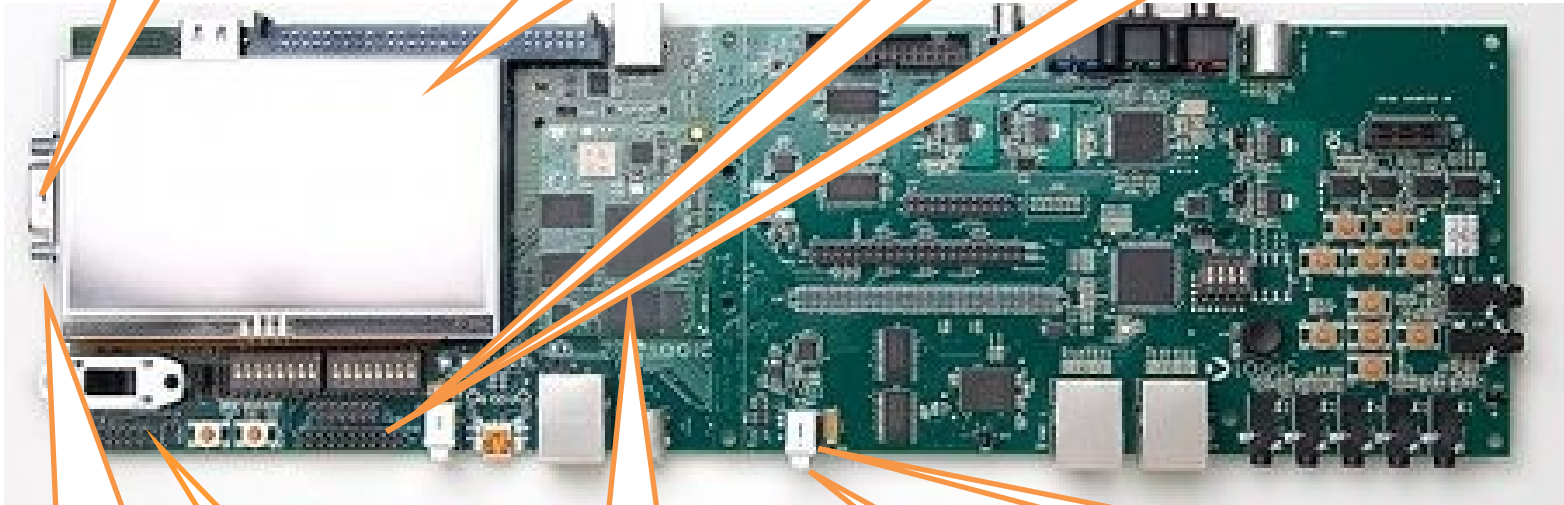
- **Graphics**
- **Touchscreen**
- **Applications**
 - Screen calibration
 - Playback control UI

• 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

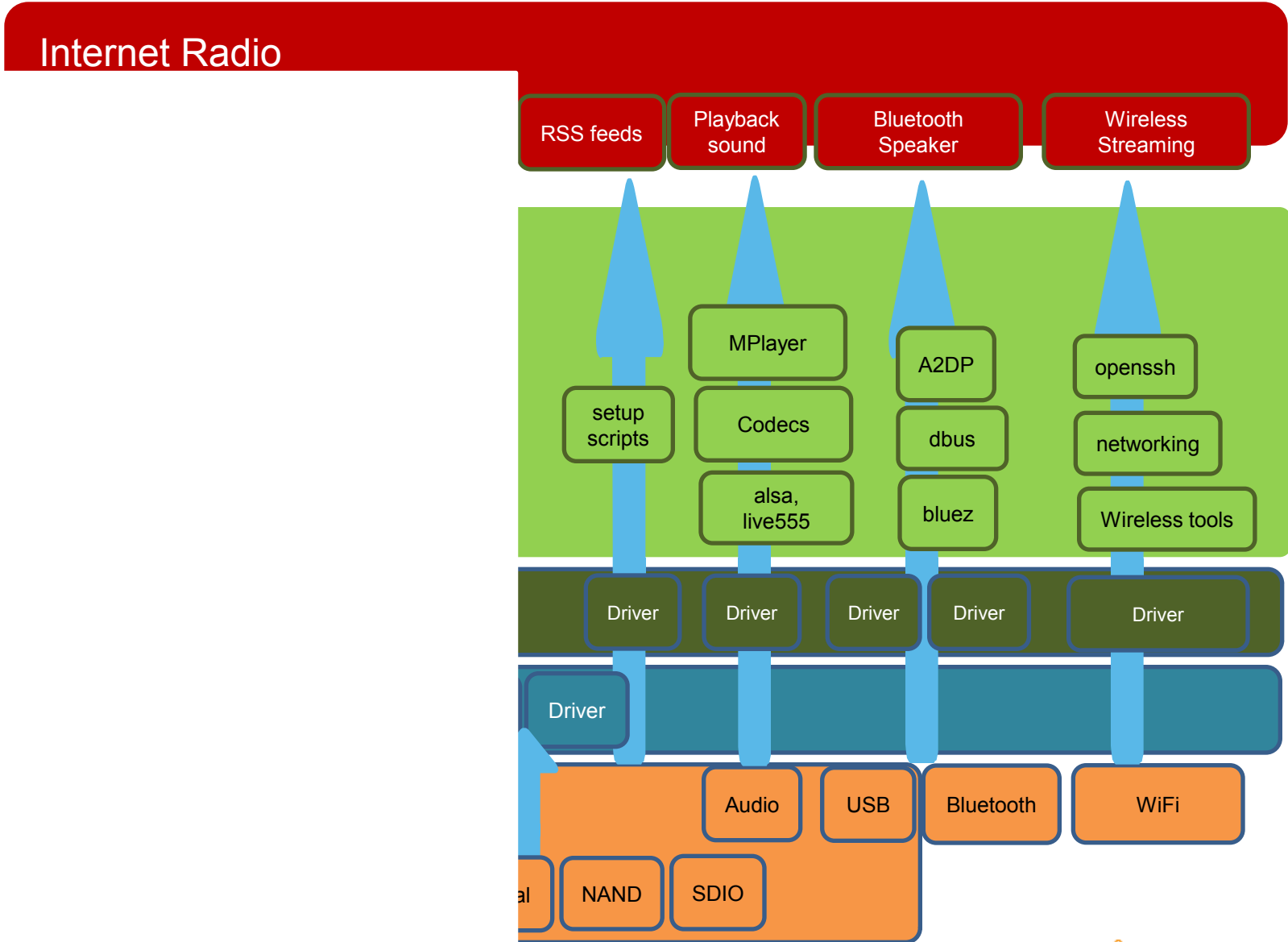
• WiFi (optional)

- Streaming audio

• Bluetooth

- Sensor connections

Internet Radio (Blueprint)



Last Week'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 Logic PD Zoom AM3517 EVM Development Kit**

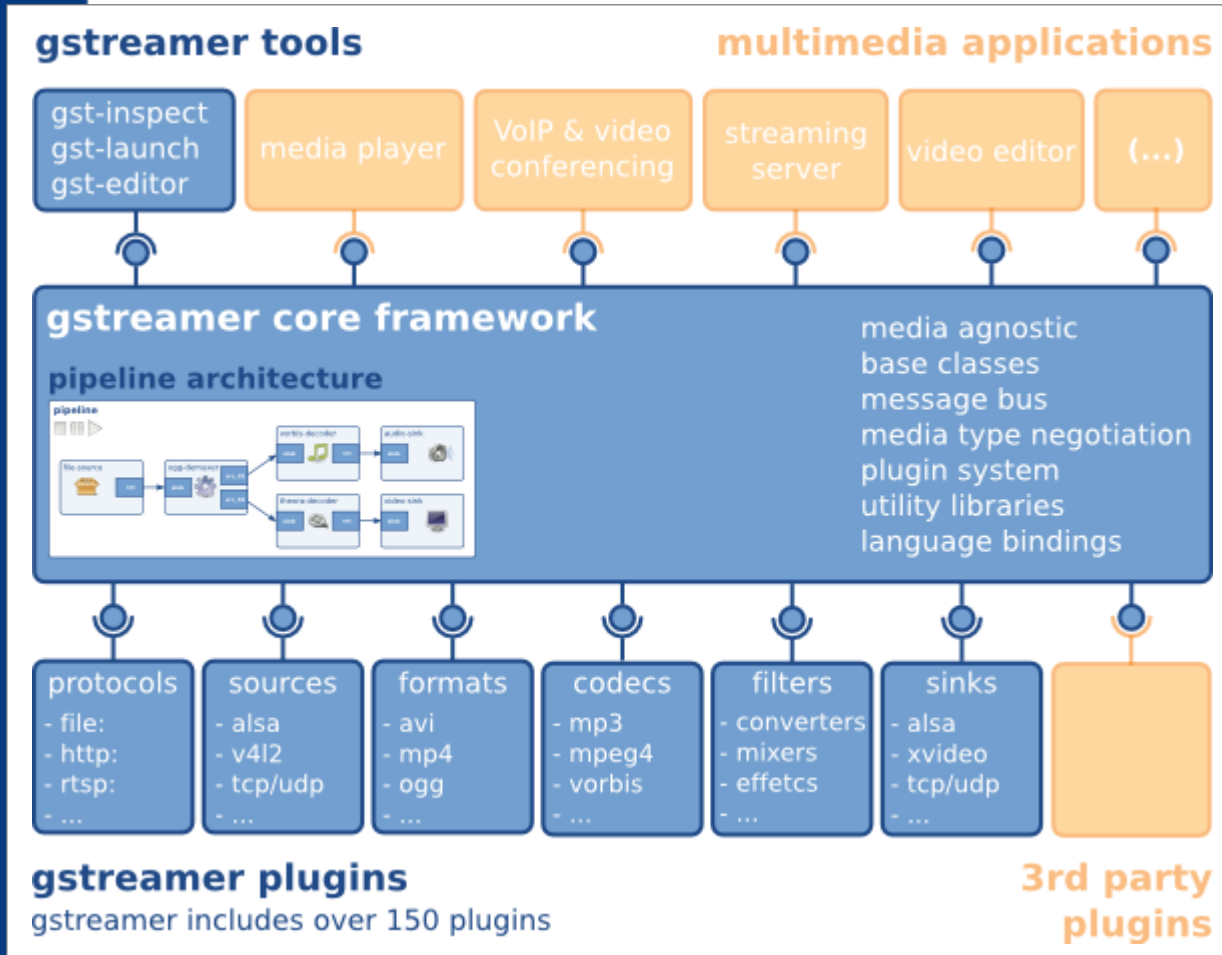


\$999 Value

Codecs 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 (available from TI)**

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**
 - Semiconductor Vendor 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.e. AM3517**
- **Integrates with Qt Phonon (for media playback)**

Non-accelerated Media Playback

- **Mplayer – GPL2**
(<http://www.mplayerhq.hu>)
- **Stand-alone media playback application**
- **Supports both audio and video**
- **Comes with its own set of codecs**
- **Leverages third-party codecs as well**
- **Supported input formats:**
 - (S)VCD (Super Video CD)
 - DVD, including encrypted DVD
 - MPEG-1/2 (ES/PS/PES/VOB)
 - AVI
 - ASF/WMV/WMA
 - QT/MOV/MP4
 - RealAudio/RealVideo
 - Ogg/OGM
 - Matroska
 - NSV (Nullsoft Streaming Video)
 - VIVO
 - FLI
 - NuppelVideo
 - yuv4mpeg
 - RoQ
 - PVA
 - streaming via HTTP/FTP, RTP/RTSP, MMS/MMST, MPST, SDP



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	glibc	Toolchains
Touch screen	tslib	Packages
Qt	qt-embedded-linux	Packages
Bluetooth	bluez (will select dbus)	Packages
Python	python	Packages
Media playback	MPlayer/GStreamer	Packages
Codecs	mpg123 libmad faac faad2	Packages
Sound	alsa	Packages
Streaming	live555	Packages
System initialization	busybox	Packages

Bluetooth



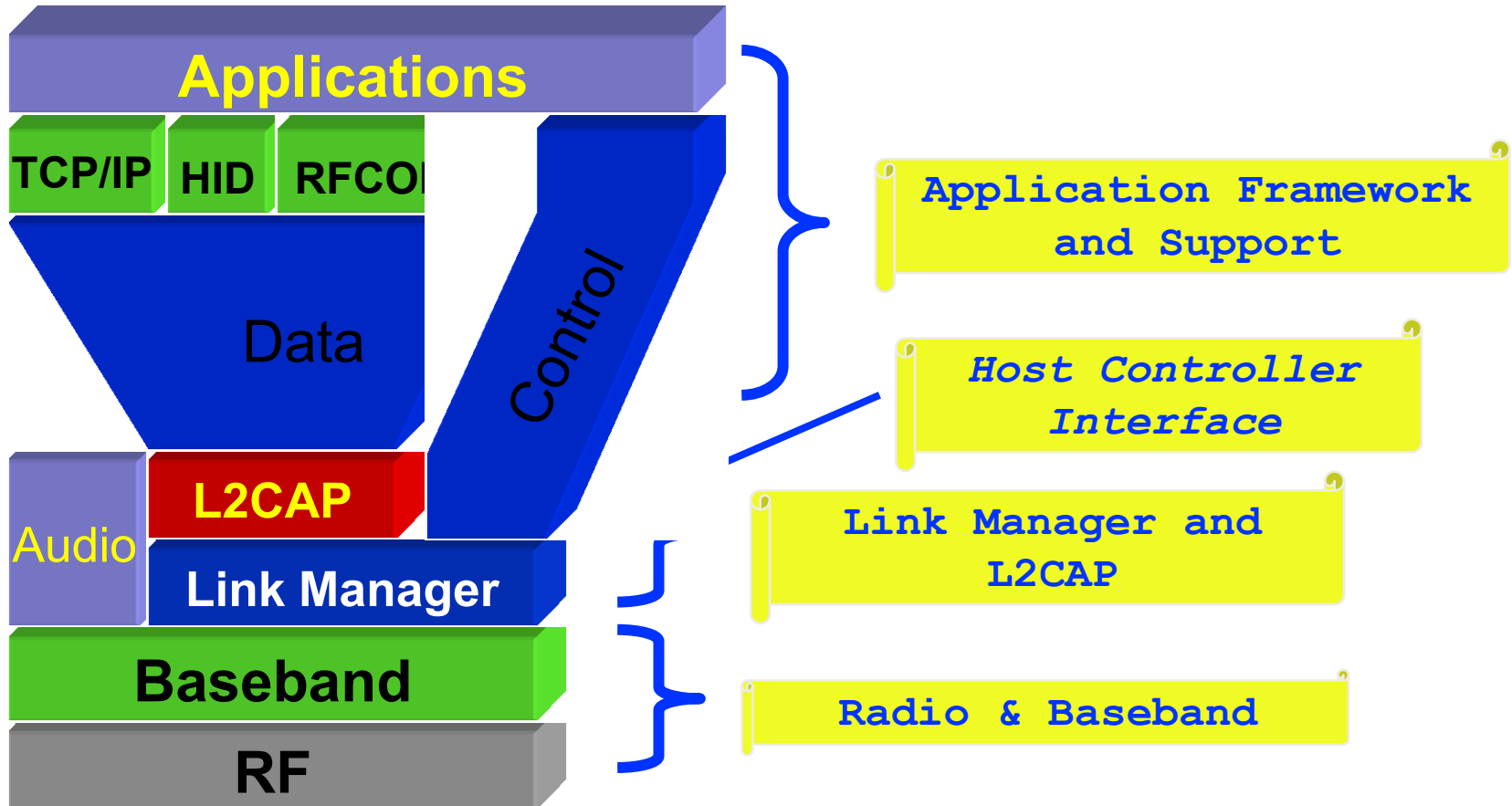
What Is Bluetooth?

- **Designed to be used to connect both mobile devices and peripherals that currently require a wire**
- **Simplifying communications between:**
 - Devices and the Internet
 - Data synchronization
- **“USB without wires”**
- **Short range wireless radio technology**
 - Operate range of 10 meters

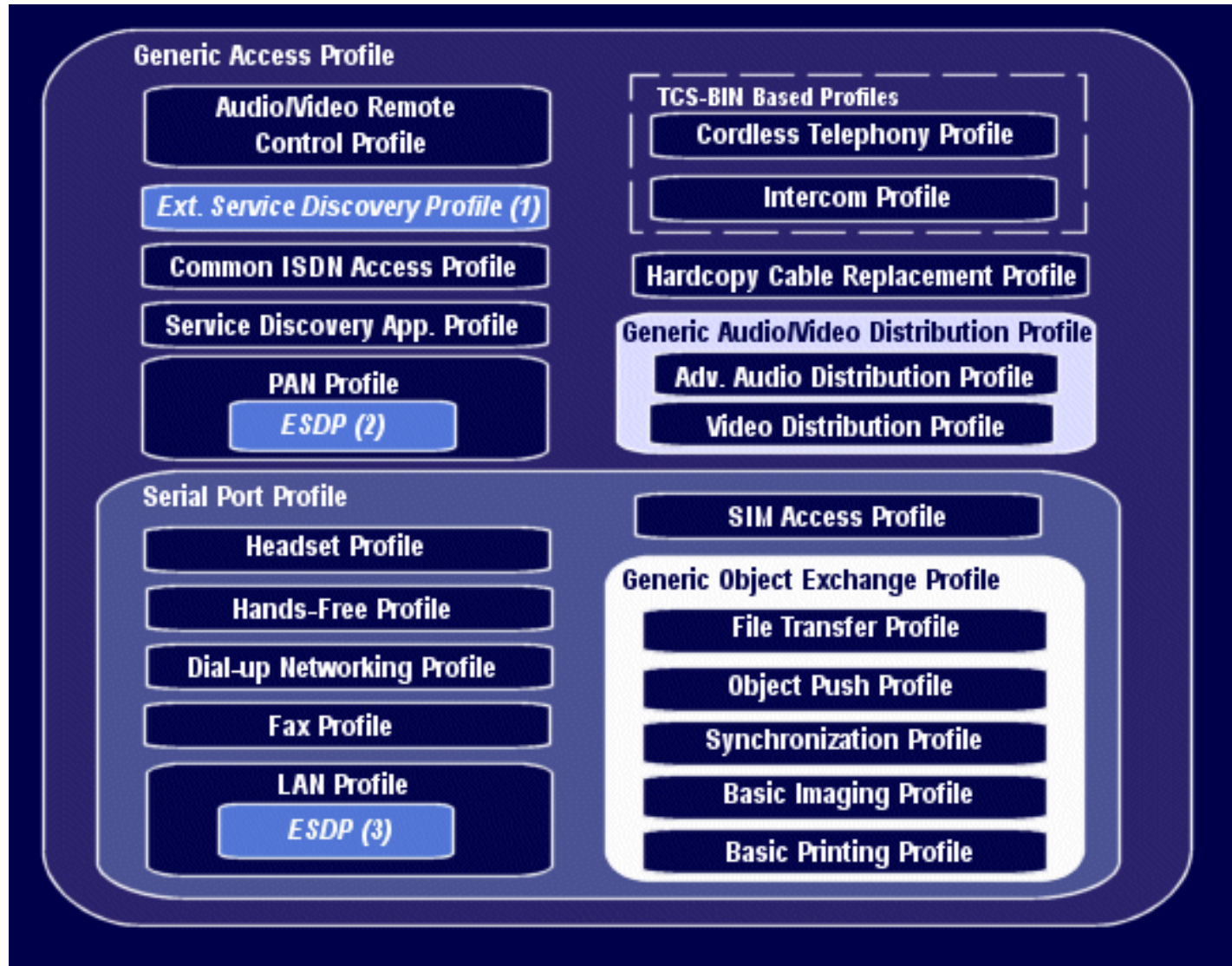


 **Bluetooth®**

Bluetooth Overview



Bluetooth Profiles



List of Profiles

- **Advanced Audio Distribution Profile (A2DP)**
- **Audio/Video Remote Control Profile (AVRCP)**
- **Basic Imaging Profile (BIP)**
- **Basic Printing Profile (BPP)**
- **Common ISDN Access Profile (CIP)**
- **Cordless Telephony Profile (CTP)**
- **Device ID Profile (DID)**
- **Dial-up Networking Profile (DUN)**
- **Fax Profile (FAX)**
- **File Transfer Profile (FTP)**
- **General Audio/Video Distribution Profile (GAVDP)**
- **Generic Access Profile (GAP)**
- **Generic Object Exchange Profile (GOEP)**
- **Hard Copy Cable Replacement Profile (HCRP)**
- **Hands-Free Profile (HFP)**
- **Human Interface Device Profile (HID)**
- **Headset Profile (HSP)**
- **Intercom Profile (ICP)**
- **LAN Access Profile (LAP)**
- **Object Push Profile (OPP)**
- **Personal Area Networking Profile (PAN)**
- **Phone Book Access Profile (PBAP)**
- **Serial Port Profile (SPP)**
- **Service Discovery Application Profile (SDAP)**
- **SIM Access Profile (SAP, SIM)**
- **Synchronisation Profile (SYNCH)**
- **Video Distribution Profile (VDP)**
- **Wireless Application Protocol Bearer (WAPB)**

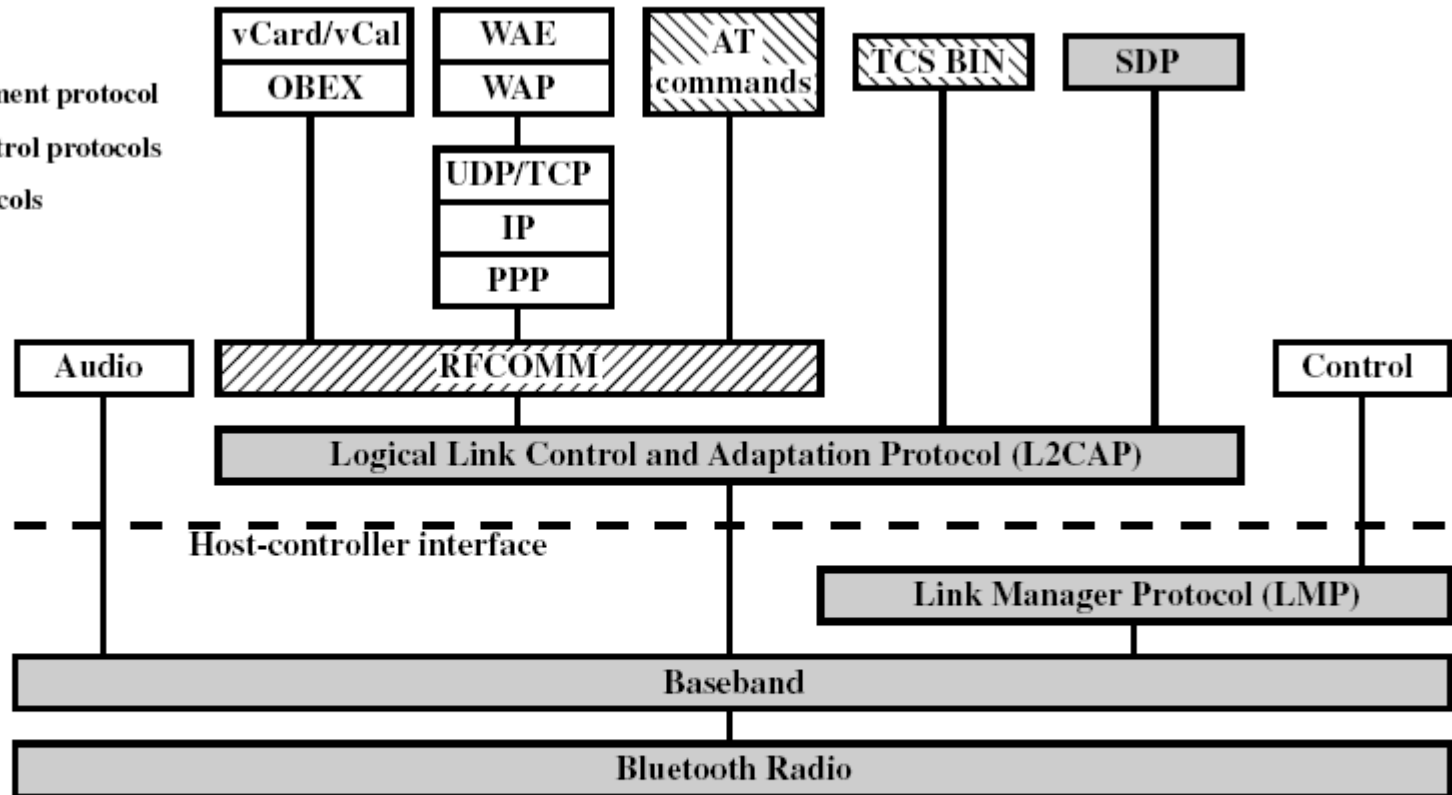
Protocol Stack

■ = Core protocols

▨ = Cable replacement protocol

▩ = Telephony control protocols

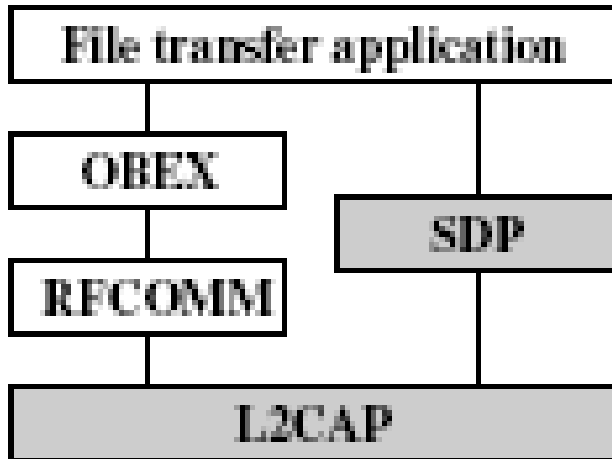
□ = Adopted protocols



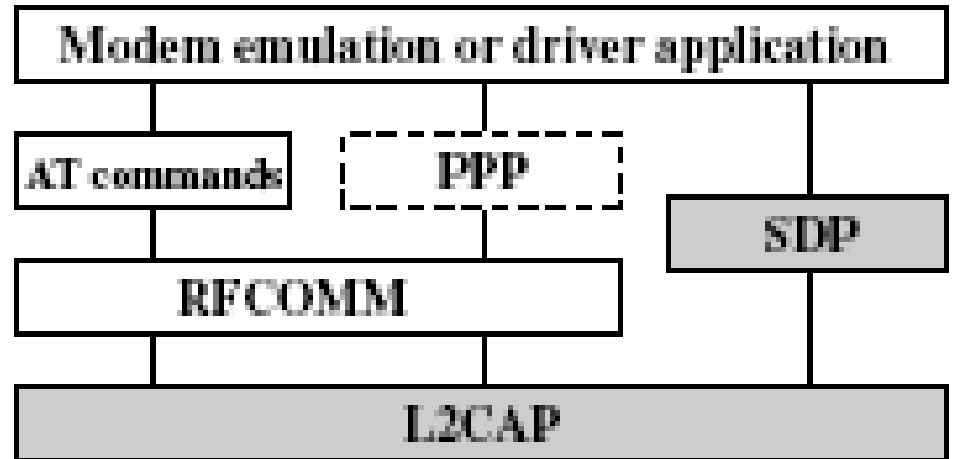
AT = attention sequence (modem prefix)
 IP = Internet Protocol
 OBEX = Object exchange protocol
 PPP = Point-to-Point Protocol
 RFCOMM = radio frequency communications
 SDP = service discovery protocol
 TCP = transmission control protocol

TCS BIN = telephony control specification - binary
 UDP = User Datagram Protocol
 vCal = virtual calendar
 vCard = virtual card
 WAE = wireless application environment
 WAP = wireless application protocol

Usage Models

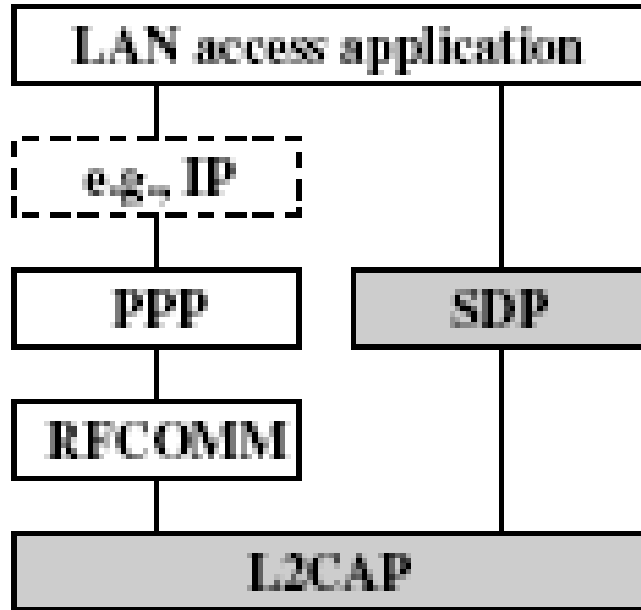


(a) File transfer

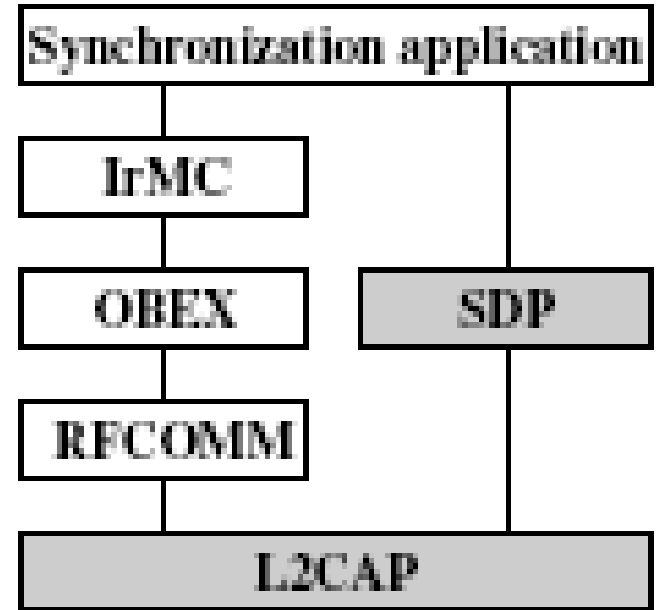


(b) Dial-up networking

Usage Models

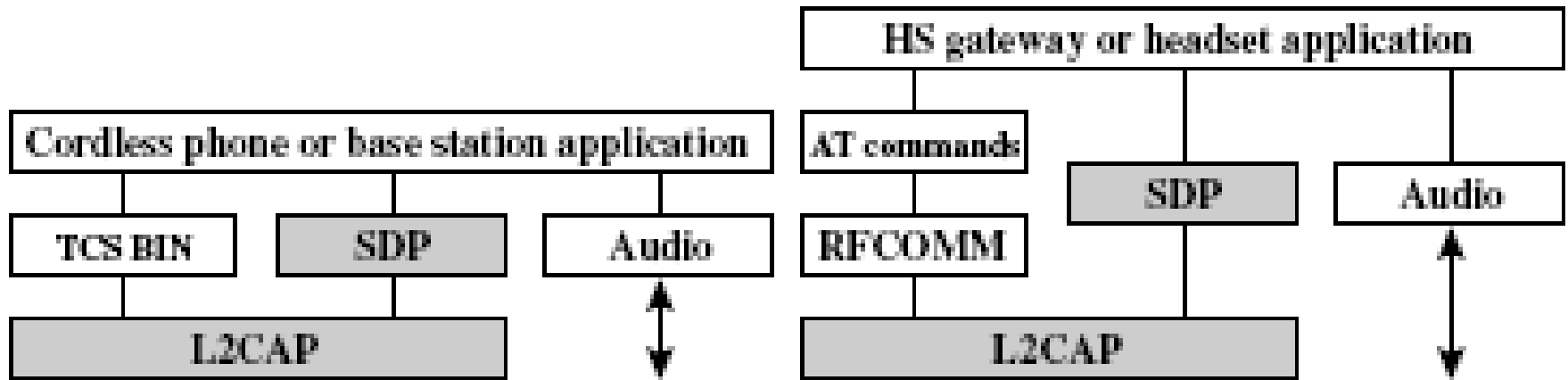


(c) LAN access



(d) Synchronization

Usage Models



(e) Cordless phone and intercom

(f) Headset

Bluetooth Tasks

■ Add packages

- BlueZ – already in our design (using version 4)
- Python – provides APIs for BlueZ scripts

■ Reconfigure the Linux kernel

- Add device drivers for Bluetooth
- Add support for the Broadcom chipset we are using

■ Setup the Bluetooth interface

- Bluetooth tools primer
- Setup interface hci0
- Connecting your bluetooth devices
 - Run Bluetooth
 - Use BlueZ test programs



Internet Radio Application – Enhancements



Internet Radio Application (Cont'd)



■ Launch MPlayer from as external process

- Use QProcess class
- Specify the arguments

```
#mplayer -ao alsa:device=bluetooth URL
```

■ Adding a station from a web browser

- Create a handling routine
- Connect signal handlers

Autoboot

- **Create a custom SXX startup script in /etc/init.d/**
 - Export QWS and TSLIB variables
 - Start the internetradio application

- **Create an overlay that includes the following:**
 - Custom startup script
 - Internetradio application
 - Bluetooth pairing scripts
 - Create a tar file with overlay directory structure

- **Using desktop factory merge the overlay into the system's RFS**
 - Using Desktop Factory Interface navigate to:
 - Target Configuration->Build RFS->RFS Content Tarball
file:///<fully qualified path to the overlay tarball

What We Have Accomplished



What We Have Accomplished

- **Cross-compiled/Deployed/Debugged our Internet Radio application on the Target**
- **Learned about how to add Bluetooth support**
 - User Space
 - Kernel Space
 - Autoboot
- **Added codecs/applications for media playback**
- **Modified Internet Radio application**
 - Launch external application
 - Handle Play/Stop actions
 - Adding a new radio station from a browser

Next Session

■ June 30 @ 1pm ET

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. Setup a generic Bluetooth network**
- 2. Pair your system with a Bluetooth device of choice**
- 3. Look at available Bluetooth profiles**
- 4. Use QProcess to launch other apps**


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.

New Timesys Subscription Model

- **Best price/performance**
- **No complex contracts or usage terms**
- **Every seat includes live, unmetered support**

LinuxLink	LinuxLink + Pro Services 	Pro Services Only
Starts @ \$5495/seat/year	Starts @ \$7495/seat/year with 16 hour block of pro services	Project based, discounts for larger projects
Includes live, expert support, multiple seats discount	Includes live, expert support, additional bundles available	
Perfect for engineering teams : <ul style="list-style-type: none"> • Looking to build embedded linux products and expertise in-house • Reduce the time & uncertainty associated with open source development • Want expert support when they get stuck 	Perfect for engineering teams <ul style="list-style-type: none"> • Wanting to focus in-house teams on differentiated development • Want to develop in-house linux expertise fast • Want expert support when they get stuck 	Perfect for engineering teams : <ul style="list-style-type: none"> • Having tight deadlines • Needing proven linux expertise – fast • Looking to eliminate development risk

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

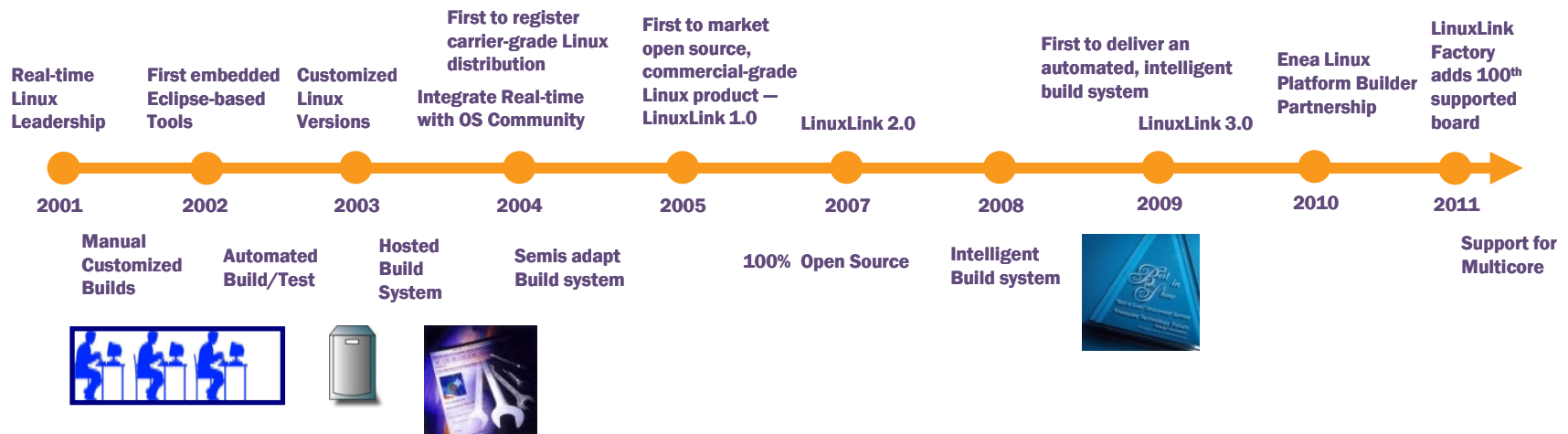
Middleware – Complete frameworks including APIs, utilities that provide specific functionality, e.g. Qt

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!

