Embedded Software Defined Radio: Design, Development, and Case Studies

United States Naval Academy
Dr. Christopher R. Anderson
Ensign George Schaertl

OpenSDR
Mr. Philip Balister
Introduction

Dr. Christopher R. Anderson
Tutorial Overview

Introduction and Overview – Chris Anderson
Overview of the OMAP3 – Philip Balister
Hardware Design – Chris Anderson

Break

FPGA Interfacing – Philip Balister
GNU Radio on the Beagle – George Schaertl
System Demonstrations – Schaertl/Balister

Note: Presentation slides available on www.opensdr.com
Listserv subscription information at www.opensdr.com/node/10
Overview of the Problem

Problem:
The software radio community is in dire need of a low-cost, moderately high performance embedded development platform.

Goals:
Develop an FPGA-based interface board between the Beagleboard (a $150 high-performance embedded processing platform) and COTS Analog Front Ends (AFE’s).
Why Embedded SDR?

- Processing power per area
- Processing power "efficiency"
Interoperability with Security – both among and between service branches.

Highly stable networks / Highly mobile networks.

**Power and Battery Life.**

Fixed and mobile infrastructure (have to bring infrastructure with you when you enter a theatre).

Every radio has networking capability with Mbps data flow to the tactical edge.

Need worldwide coverage and instant availability ("push-to-talk")

Extremely low latency – users must get critical information in a timely manner.

Long product lifecycles (decades or more), lifetime maintainence.
SDR for consumer grade communications

Coverage with fixed infrastructure.
Capacity to handle peak user demand.
Intuitive, easy to use user interface.

**Power and Battery Life.**

Data (text and internet) has surpassed voice communications.
Integrated services and applications (phone, data, text, GPS).
Latency not a major issue.

Short consumer product life cycles (18-24 months), long infrastructure life cycles (multiple years).

Consumers may want SDR flexibility but not currently willing to pay for it.
Why the BeagleBoard?
What does it provide?

B ring your own peripherals
E ntry-level cost ($149)
A rm Cortex-A8 (600MHz, superscaler)
G raphics and DSP / Video accelerated
L inux (and WinCE) ports
E nvironment for software innovators
A detailed look at the Beagleboard

- Expansion (I²C, I²S, SPI, MMC/SD)
- User Button (Boot Select)
- Reset Button
- USB 2.0 HS Host
- SD/MMC+
- RS-232 Serial
- Alternate Power
- JTAG
- DVI-D
- TI OMAP3530 + 128MB DDR + 256MB NAND
- S-Video
- Stereo Out
- Stereo In
- USB 2.0 HS OTG (Power)
Beagleboard Specifications

Industry’s highest performance ARM
600MHz ARM Cortex-A8 (>1GHz ARM11 MIPS)
NEON and VFP extensions for additional acceleration
State of the art POWERVR™ graphics hardware
10 million polygons per second
Advanced tile-based rendering
Industry leading C64x+ DSP & video acceleration
Streaming, portable media player, and high-res video
Tiny 3” x 3” PCB that can fit in your pocket
Yet support for 1280x1024 DVI-D monitors
Why GNU Radio?

Popular Open source SDR project
Makes extensive use of floating point
This has limited its use on embedded systems
Built on Beagle, limited testing
Need to work out USB port issues
What to do
Add NEON specific signal processing code
Embedded SDR is a capability that is (or will soon be) in high demand by the SDR user community.

Key features are battery life and performance that is "good enough".

Platforms such as the Beagleboard or Overo can provide a development environment for these embedded applications.
Tutorial Overview

Introduction and Overview – Chris Anderson
Overview of the OMAP3 – Philip Balister
Hardware Design – Chris Anderson

Break

FPGA Interfacing – Philip Balister
GNU Radio on the Beagle – George Schaertl
System Demonstrations – Schaertl/Balister