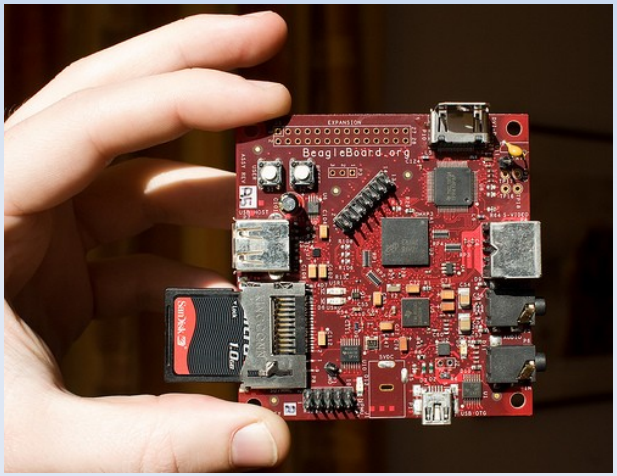
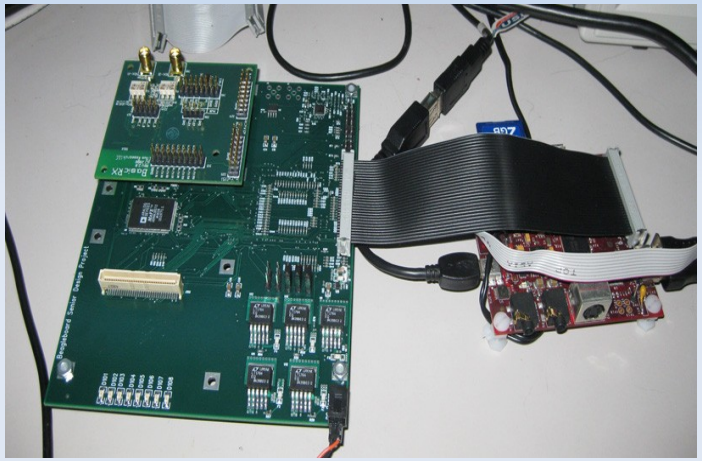
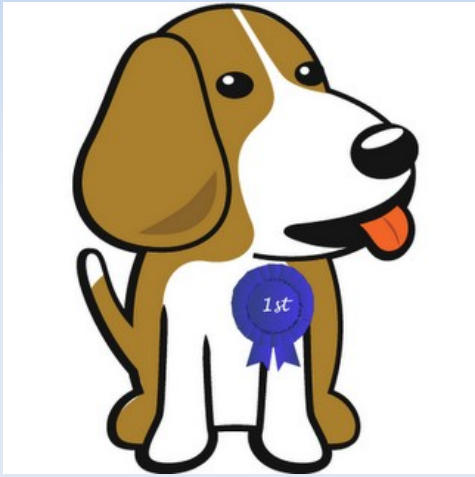
A Low-Cost Embedded SDR Solution for Prototyping and Experimentation







United States Naval Academy

Dr. Christopher R. Anderson

Ensign George Schaertl

OpenSDR

Mr. Philip Balister

USNA Wireless

Presentation Overview

Background and Overview of the Problem

Beagleboard as an SDR Platform

System Architecture / Hardware Overview

Performance Results Thus Far

SDR for military and tactical communications

Interoperability with Security – both among and between service branches as well as for coalition operations.

Highly stable networks / Highly mobile networks.

Power and Battery Life / Extremely small footprint.

Fixed and mobile infrastructure.

Every radio has networking capability.

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

Extremely low latency.

Long product lifecycles (decades or more), lifetime maintainence.

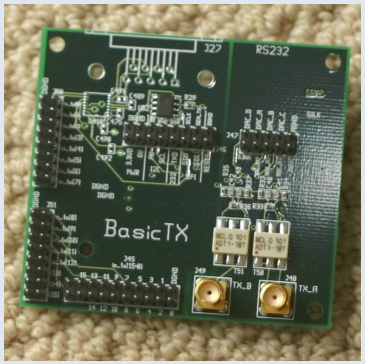
Overview of the Problem

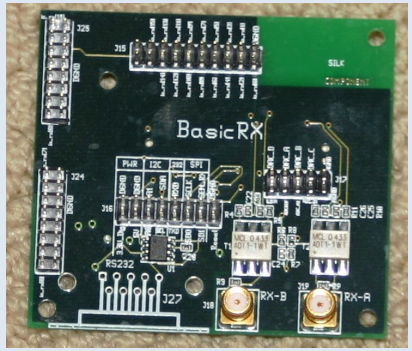
Problem:

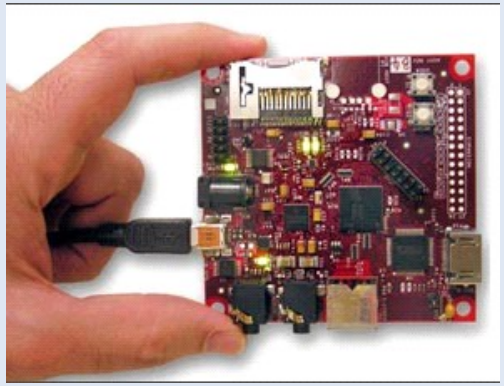
The software radio community is in need of a low-cost, moderately high performance embedded development platform.

Goals:

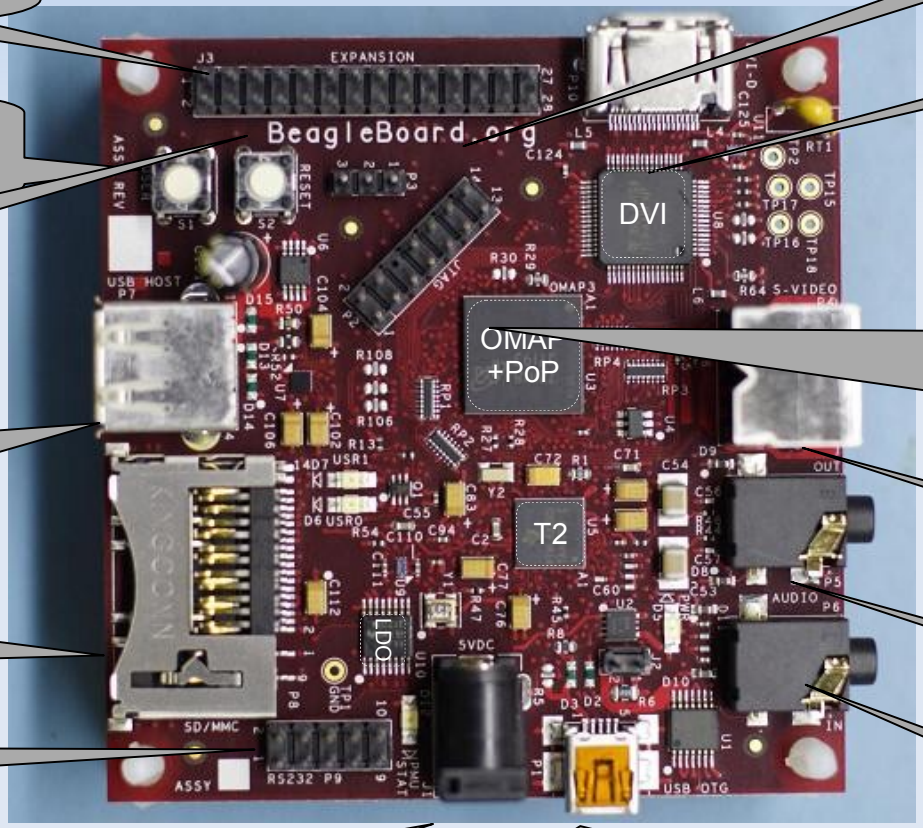
Develop an FPGA-based interface board between the Beagleboard (a $150 high-performance embedded processing platform) and COTS Analog Front Ends (AFE's).







The Beagleboard as an SDR Platform



Expansion
(I²C, I²S, SPI, MMC/SD)

User Button
(Boot Select)

Reset Button

USB 2.0 HS Host

SD/MMC+

RS-232 Serial

Alternate Power

USB 2.0 HS OTG
(Power)

JTAG

DVI-D

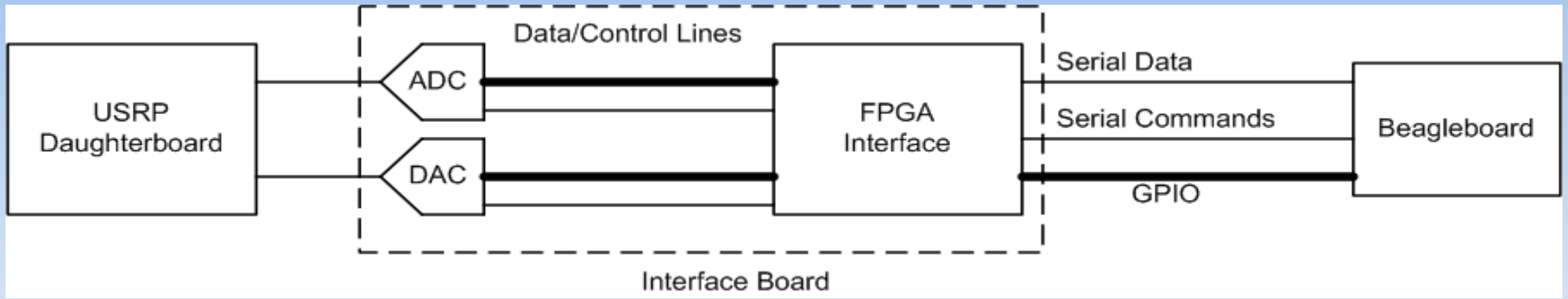
TI OMAP3530
+ 128MB DDR
+ 256MB NAND

S-Video

Stereo Out

Stereo In

Overall System Architecture



Interconnect a USRP Daughterboard with the Beagleboard Expansion Header.

Utilize a Cyclone III FPGA to perform initial filtering, digital downconversion, heavy signal processing.

FPGA also controls sampling clocks, data flow, and routes command & control signals from the Beagleboard.

Integrate the solution onto a small, low-cost PCB.

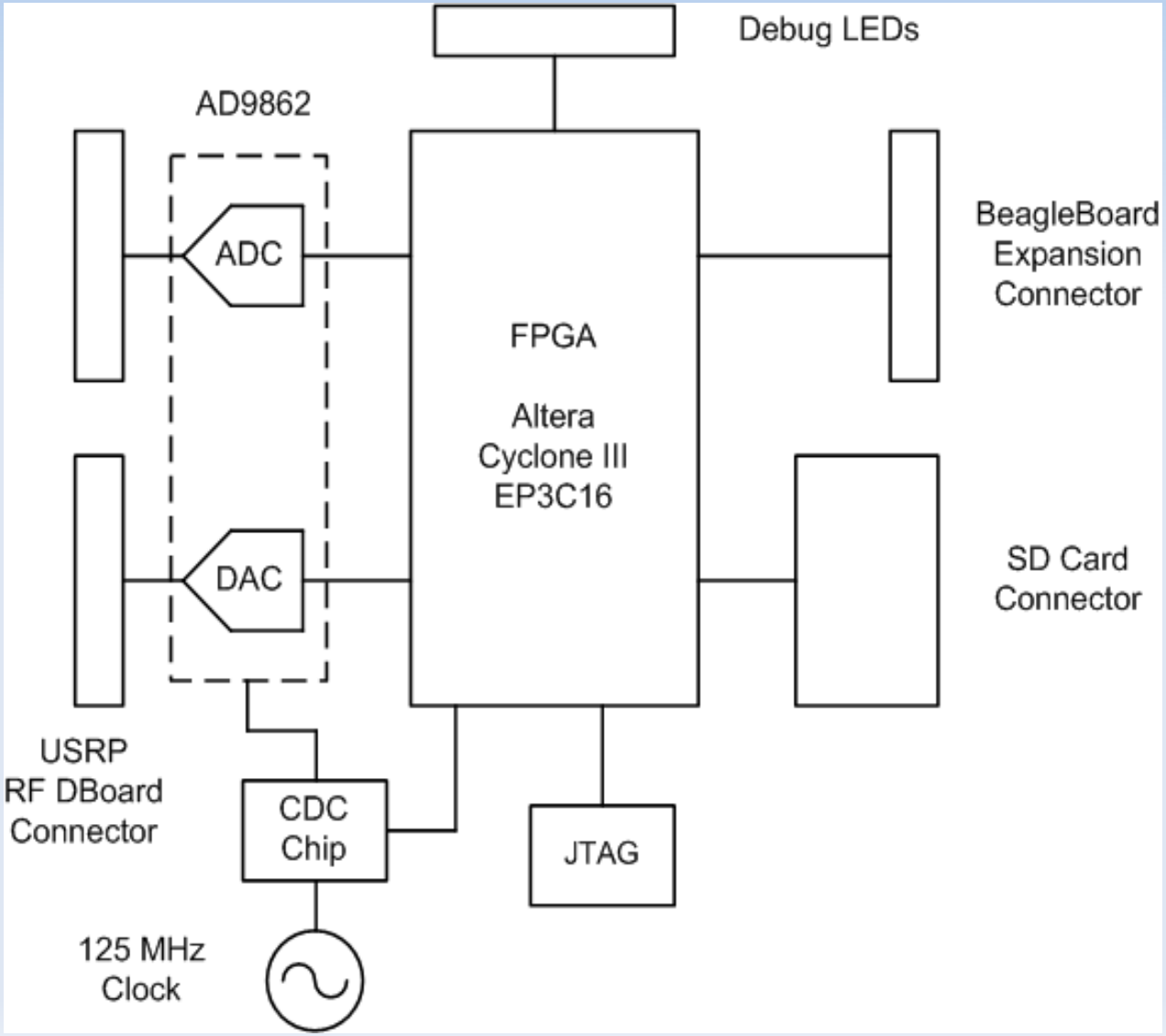
Demonstrate complete system integration with a version of GNU radio running on the Beagleboard.

SPI Interface to the Beagleboard

- Linux driver based on spidev
  - No interrupt handling only useful for simple tests
  - Capable of 3.5 Mbytes/second (6 MB/s theoretical Max)
  - 875 kHz bandwidth
- FPGA code to generate data and control signals
- GNU Radio component based on spidev
- Control interface uses seperate SPI interface
- Kernel driver also developed and tested

Interface Board Block Diagram

Revision 1B Design



Board Features

Analog Devices AD9862 Operating at 62.5 MHz

Single USRP Daughterboard w/Full Duplex TX/RX

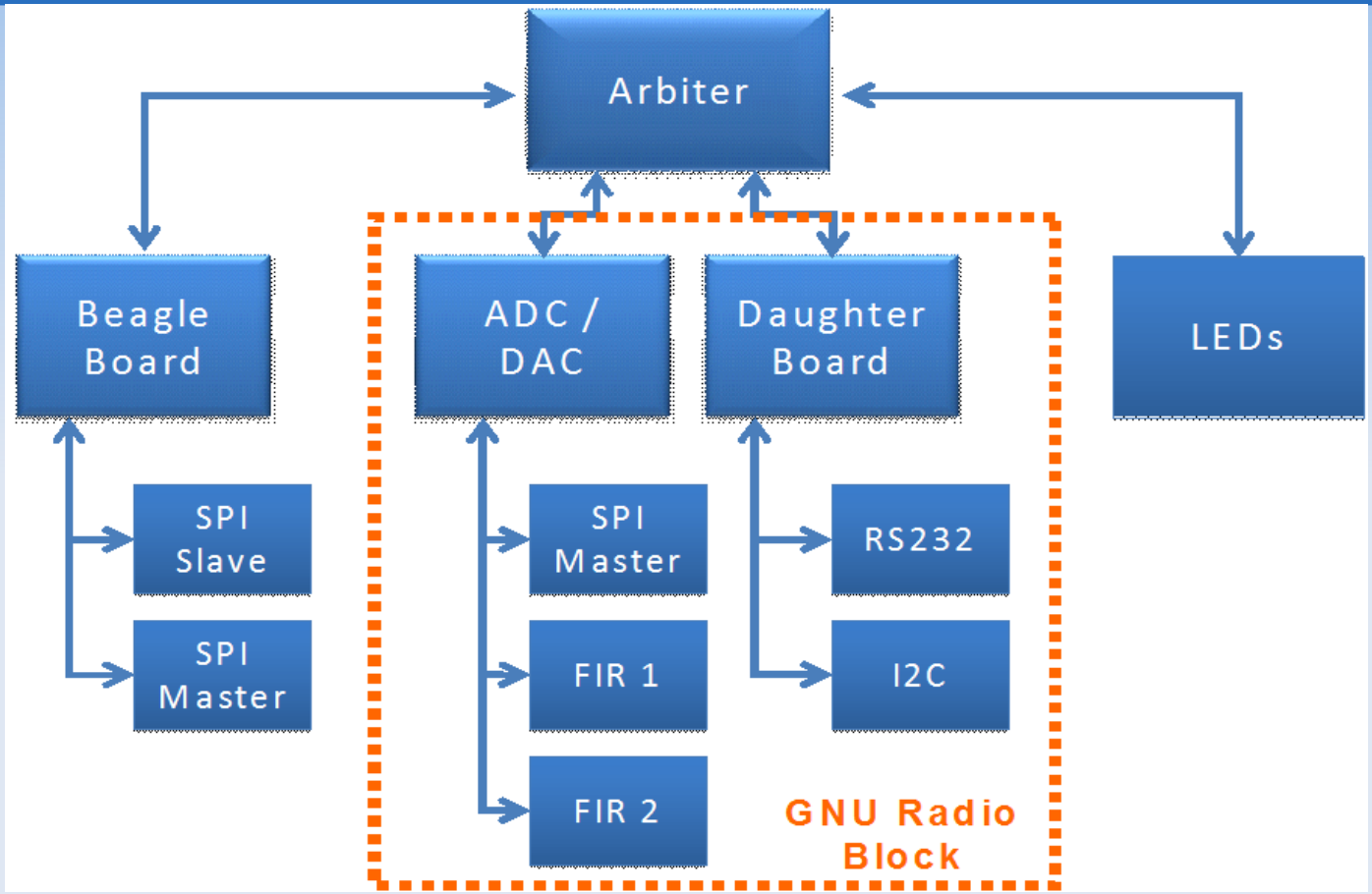
Altera Cyclone III FPGA

Master Clock of 125 MHz FPGA clock up to 250 Mhz

SD Card slot for FPGA Configuration

Connects to BeagleBoard Expansion Header

FPGA Code Design



Major Components

- Arbiter
- Daughterboard Interface
- ADC Control
- BeagleBoard Interface

Minor Components

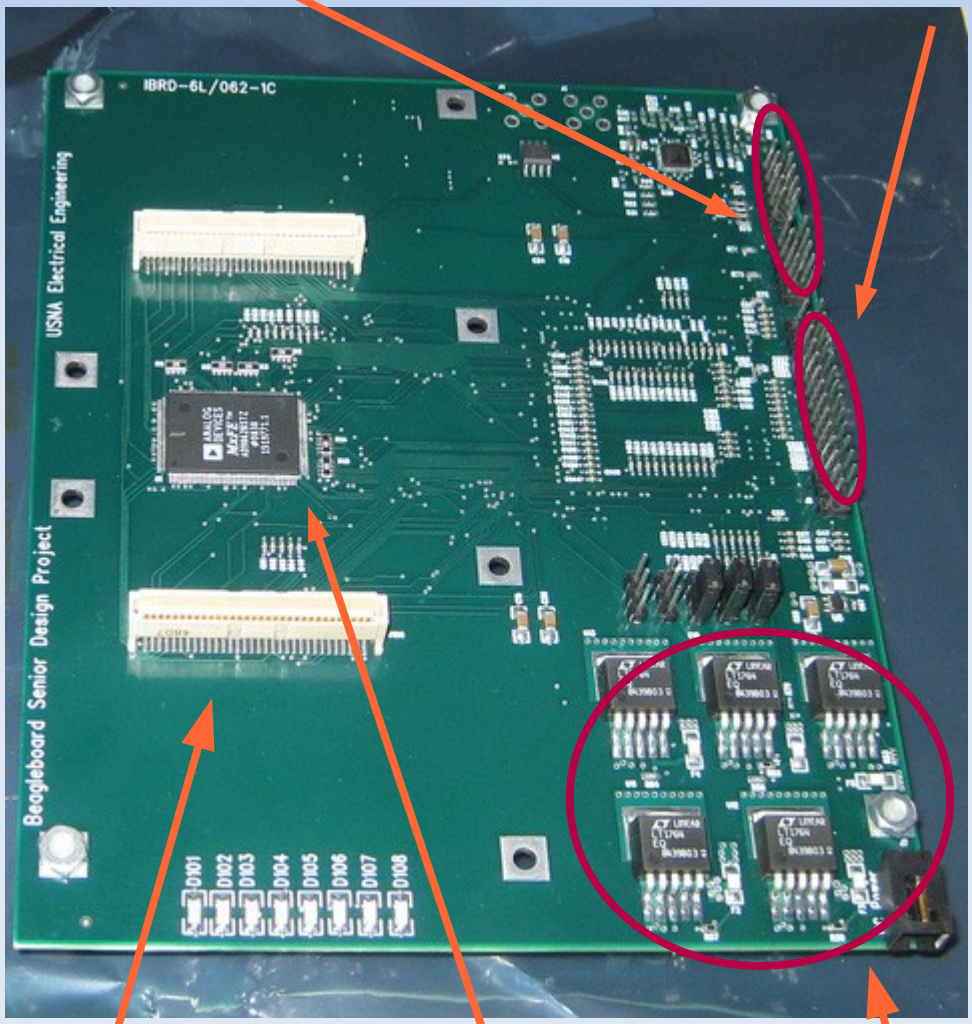
- SPI Master & Slave
- I2C / RS232
- FIR Filters
- Debug LEDs

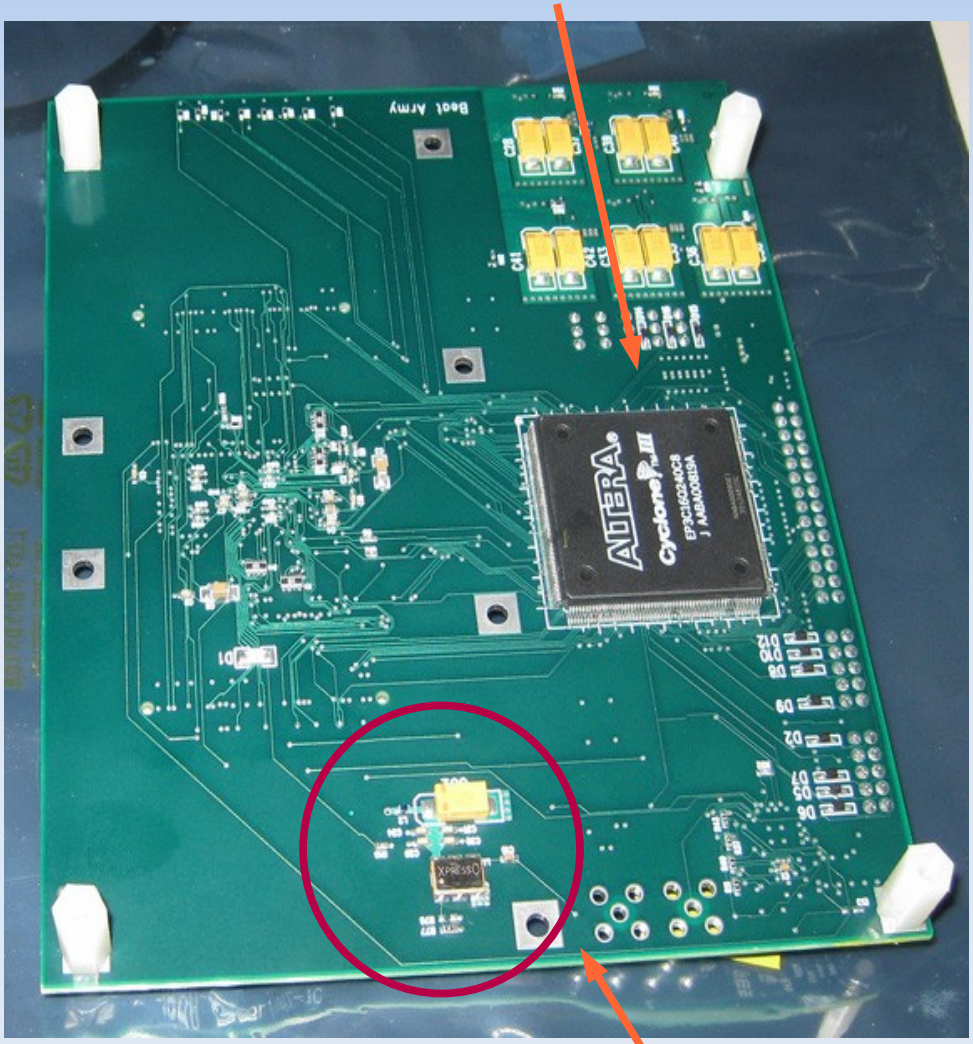
Interface Board Rev 1A

JTAG Headers

Beagle Header

Cyclone III FPGA





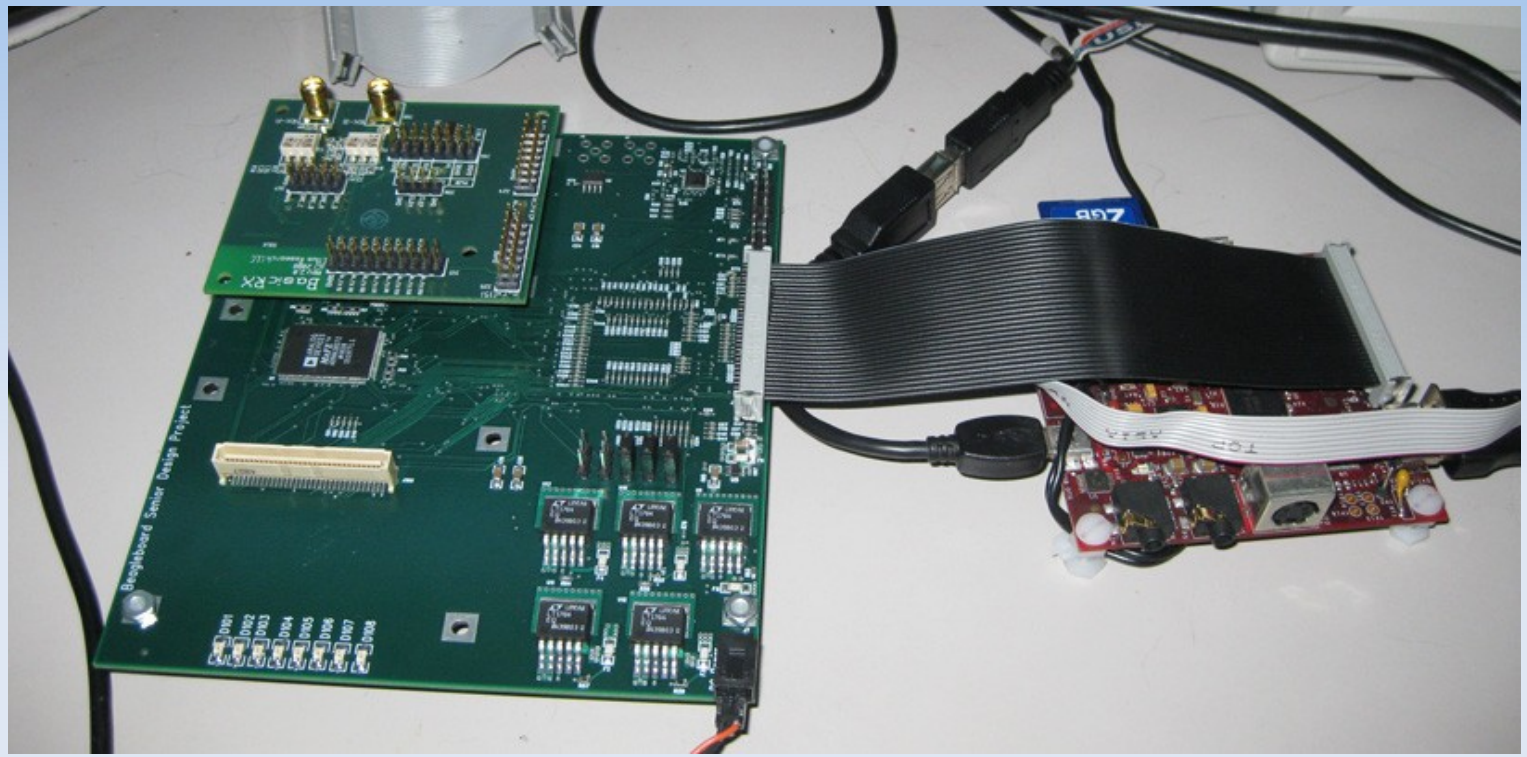
Daughterboard Connection

AD9862 ADC

Power Supplies

Clock Distribution

The Complete System



http://www.flickr.com/photos/32615155@N00/3447696549/

USRP daughterboard ($75-500): RF gain/filtering

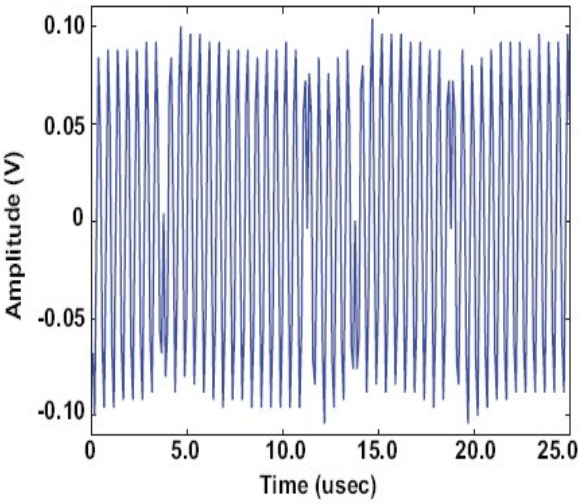
Custom board (~$500): ADC, down-conversion

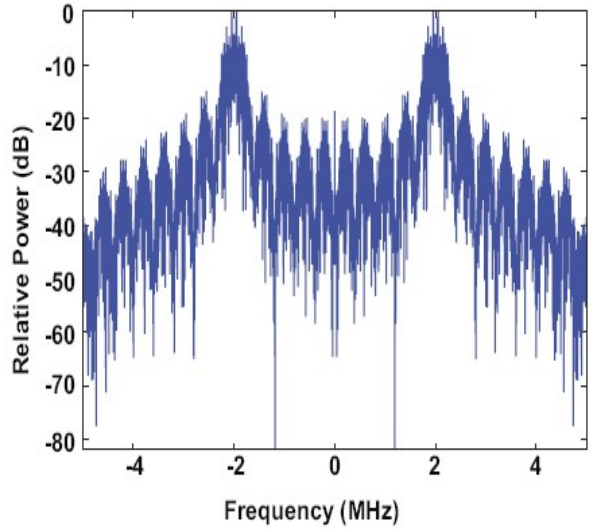
BeagleBoard (~$150)

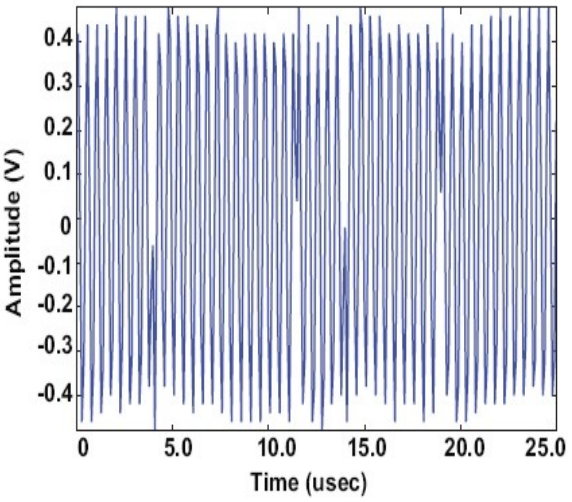
Kernel SPI Driver Results

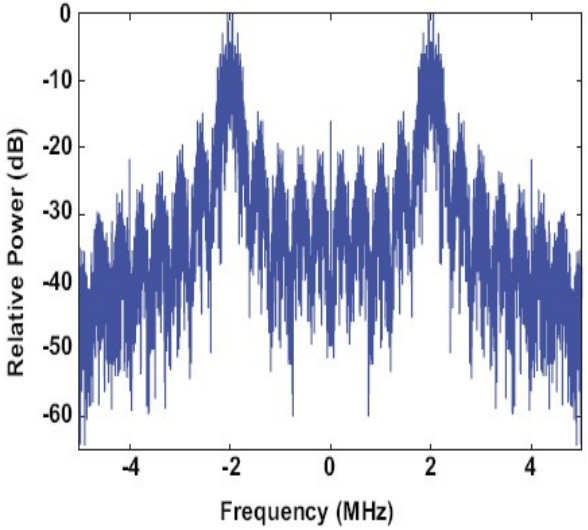
- Supports transfer rate of 875 KSPS
  - Convert low level kernel drive to use DMA
  - Enable HW FIFO
- Compared processor usage with USRP
  - SPI – 0.42 seconds of system time
  - USRP – 8 seconds of system time
  - Total transfer time – 132 seconds
- SPI interface has much less overhead

System Performance Results









Front End Performance

System operating in a loopback mode.

2.0 MHz carrier BPSK modulated with 200 kbps data stream.

ADC parameters controlled by GNU Radio running on Beagle (via SPI through FPGA).

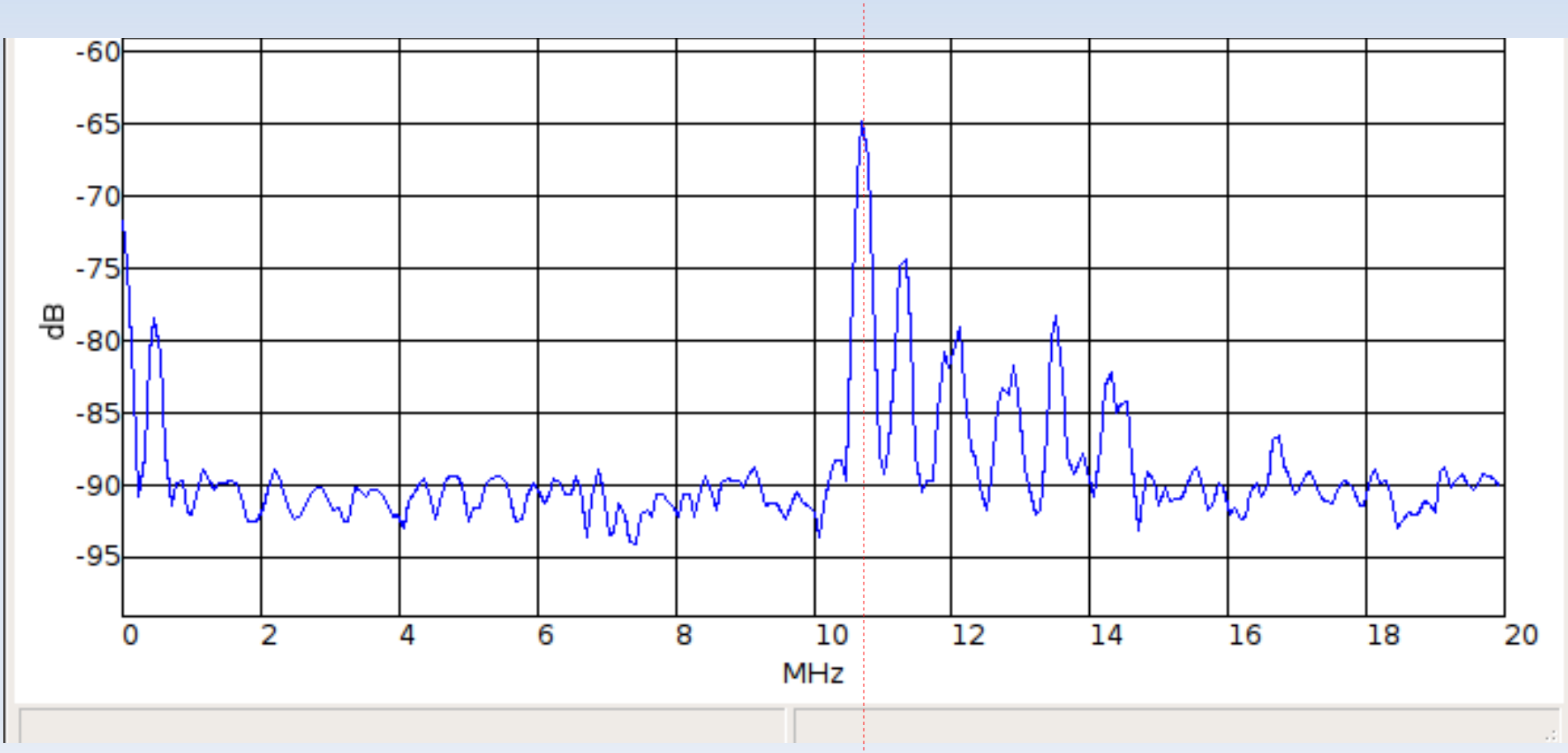
Data successfully rebroadcast (no bits corrupted, no increase in EVM)

Other Performance Results

AM/FM Signals downconverted to 10.7 MHz via an ICOM IC-R9000 receiver.

Successfully demodulated signals on the Beagleboard – played back audio.

Successfully upconverted stored audio samples and broadcast AM/FM.



Conclusions

SDR end users are demanding low-cost, low-power, highly portable platforms: a need that is not currently being met.

We were able to demonstrate the ability to create a low-cost, small SWAP, reasonable performance Embedded SDR Platform.

Performance results thus far indicate the system should be capable of supporting up to 1.5 MSPS (16-bit complex samples) via SPI driver improvements.

Further improvements and performance testing is currently ongoing.