Training the Next Generation of Hardware Hackers

Teaching Computer Organization and Assembly Language Hands-On with Embedded Systems

Development

- Took about a year to develop and run through our first semester with the material
- We want to share our ideas, what we learned, and the tools we built with everyone
- We want to try to spread the hardware hacking culture

Goals

- Get Undergraduate Electrical Engineering and Computer Science students thinking “closer to the metal”
- Computer organization is an integral part understanding what your code does
- Assembly Language facilitates learning how Computers and Microprocessors work
- Why teach Assembly Language?

Assembly Language

- For the actual design process, many times it far easier and far more cost effective to use C or other high-level languages
- However, Assembly puts you down closer to the machine, and give you the perspective to understand what's happening
- Learn the hard way, then use the Enlightened way (high-level language) once you understand the fundamentals

How?

- Put a development kit in the hands of every student taking the course
- Flatten the learning curve of working with embedded systems
- Give each student a free, easy-to-use tool-chain with which to work

Learning Curve

- Embedded Development is surprisingly difficult to get into for newbies
- Embedded Systems IDEs can be very clunky and hard to use (Eyes toward Freescale)
- Projects like Arduino have made lots of headway - still not optimal for Comp. Org.
- Needed a better set of tools

The Bootloader

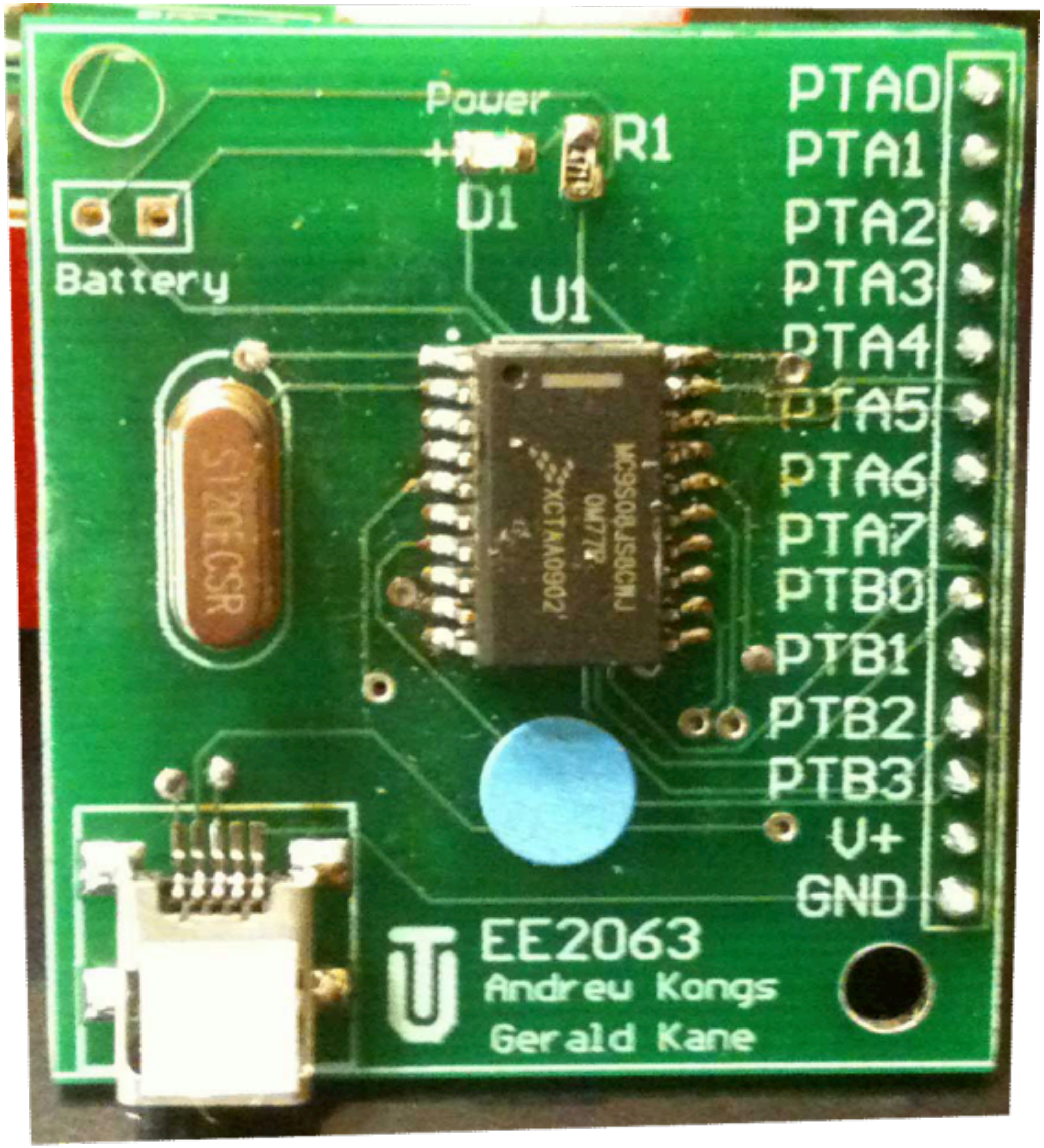
- On-chip "ROM" Bootloader from the factory
- USB boot-loader allows loading code from a student's PC
- No expensive device-specific programmer needed
- Problem: Current Boot-loader is from the manufacturer and it only works with Windows (we're going to fix that)

Development Board

- Uses Freescale MC9S08JS16 chip
- Single Active IC
- USB boot-loader functionality
- 10 GPIO Pins (2 more with caveats)
- About 1.6” square
- $25 cost, in 50 qty

The Kit

- JS16 Development Board
- Small solderless breadboard
- USB Cable
- Parts for first project
- Packaged in a 6”x8” Zip-Lock Static Bag
- Various accessory daughter boards for each project (minimizes wiring, helps concentrate on the machine organization)


Power
R1
D1
Battery
U1
MC9S08JS8CNJ
0M77R
XCTAA0902
S120ECSR
PTA0
PTA1
PTA2
PTA3
PTA4
PTA5
PTA6
PTA7
PTB0
PTB1
PTB2
PTB3
V+
GND
EE2063
Andreu Kongs
Gerald Kane

Photos/Videos of Accessory boards

Assembler

- Built using python, runs on all major OSes
- translates Assembly files into Freescale's S19 file format for boot-loading
- Generates Human-Readable Listing Files
- Implements Macros, and a few custom tweaks that are unique (as far as we know)
- Use whatever text editor you have available

First Semester

- 17 Students - Electrical Engineering and Computer Science
- Each given a development board for the semester
- Four projects, One hands-on Exam
- One professor, one teaching assistant

Projects (First Gen.)

- Blink an LED (show you can use the toolchain)
- LED Patterns to simulate real (and fictitious) events -- (e.g. a traffic light)
- Scrolling Characters on an LED Display (Row/Column Arrangement)
- Crayon Vending Machine system with motor control

Lessons from 1st Semester

- Static Safety-- students need to be taught forcefully about static. An ESD warning sticker and telling people to be careful isn't enough
- Reworking boards is incredibly time consuming (for the Teaching Assistant)
- Using the manufacturer's bootloader was a limiting factor and it needs to be reimplemented

Future Work

- Rewriting the bootloader as a USB mass-storage device (cross-platform)
- More example code, continue the never-ending drive for better documentation
- Put an inexpensive In-Circuit Emulator in the hands of each student
- In-System Instruction-by-Instruction debugging is an incredible learning tool, but it is expensive for now (non-free software)

Documentation Available at:

ee.base2.us

Completely Open

- Gerber files available for PCB
- Assembler available under the GPL
- Parts List, instructions for putting it all together available online
- We want others to use, modify, and help spread the things we've developed

Questions?

Expected Question

- I want a board!

- Everything is open source - you can make one with the information provided
- We don't want or have the means to sell or make them in quantity - we can show you how to build one