ECE 471 – Embedded Systems
Lecture 1

Vince Weaver

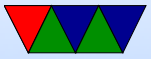
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

29 August 2016

Introduction

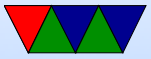
- Distribute and go over syllabus
- Talk about the class.
  - Homeworks, 50%: 11 total, lowest dropped.
    Most will involve the Raspberry Pi.
    Generally will be due on Thursday by beginning of class. Will usually have at least a week to do them.
    Submission by e-mail, grades sent in response to that e-mail, if you don't like that let me know.
    Will send out e-mail when assignment posted on



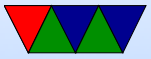
website.

- Midterm, 10%: week before fall break due to travel
- Final, 15%
- Class participation, 5%
- Project, 20%: Involves using what you learned to do a small embedded project, with a final writeup and demo the last week of classes. More details as we get closer.

- Late work penalty



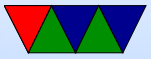
- Class notes will be posted on the website.
- Will involve C coding, plus some minimal ARM assembly language and Linux knowledge. I will review everything you need to know.
- Might have some more C instruction this year based on exit-interview feedback. Believe it or not we actually listen.
- Lab: no dedicated lab. In past students have used the electronics lab because HDMI and keyboards available, not sure situation with renovated labs (no more USB keyboards?). Xiang Guo is the TA for the class.



Raspberry Pi

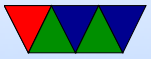
- We will be using a Raspberry Pi.
  Model B+ or Model 2-B are probably the best, but any of the models (A, B, A+, B+, 2) should work with the homeworks. No compute node. Zero probably will work but a bigger pain to use. Pi3 will probably be OK for this class too but there are a lot of non-trivial changes and I haven't test it so your job might be more difficult.

- You will also need an SD card (4GB or bigger). Older ones take the big ones, newer the small ones. Usually



not a problem as they tend to come with those adapters. You will want to install Linux (I tend to use Raspbian); getting a card pre-installed with Raspbian or "NOOBS" can save an hour or so of writing the SD card.

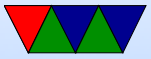
- For power you will need a USB-micro cable. You can power from any desktop or laptop (or a 1A or higher USB charger)



Other Accessories

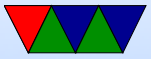
It can be fun to accessorize, but the stuff listed on the previous page is all you really need. Listed below are some *optional* extras you can get.

- A case can be useful, if only to avoid accidentally shorting out things. Many people get by just fine without one.
- A wall outlet adapter (a USB charger more or less)
- A dedicated GPIO connector to breadboard adapter
- HDMI cable and USB keyboard

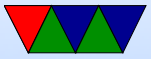


Other Hardware

- You will eventually need a breadboard. I know EE/CE students probably already have many already.
- I will loan out various devices/displays when necessary. I'll expect them back at the end of the year so try not to lose them.

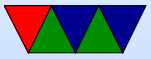


Embedded Systems



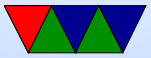
What is an embedded system?

- Embedded. Inside of something.
- Fixed-purpose.
  Why? You can optimize. For cost, for power, for size, for reliability, for performance.
- Resource constrained. Small CPU, Memory, I/O, Bandwidth
- Often real-time constraints.



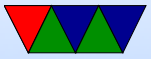
What are some embedded systems?

- Cellphone (though lines blurring, general purpose)
- Vehicles (Cars/Airplanes)
- Appliances (TVs, Washers), Medical Equipment
- Space Probes
- Video Games?



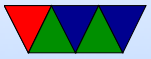
What Size CPU/Memory?

- Anything from 8-bit/tiny RAM to 32-bit 1GHz 1GB
- Performance has greatly improved over the years. ARM Cortex A9 in an iPad2 scores same on Linpack as an early Cray supercomputer

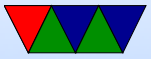


Common Low-End Embedded Architectures

- Somewhat dated list, from EE Times 2003. Multiple answers so doesn't necessarily sum up to 100%
- 8-bit processors
  - Microchip PIC – 43%
  - AVR, etc. 8051 – 55%
  - Motorola 68xx – 36%
  - Zilog Z80 – 15%
- 16-bit processors



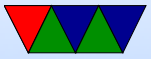
- 8086/80186/80286 – 41%
- 68HC12 – 21%



What Processors Commonly Used?

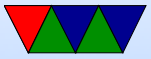
As reported by IDC at the SMART Technology conference in San Francisco for 2011

- ARM 71%
- MIPS 11%
- Other 9% (Linux supports 20+ architectures)
- x86 8% (at least Intel's desperately trying)
- Power 2%



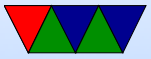
We'll Mostly Use ARM in this Class

- Widely used
- You'll see if it you move to industry
- Other classes in ECE using it



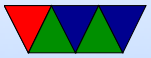
System-on-a-Chip

- Moore's law allows lots of transistors
- Discrete Chips: CPU, GPU, Northbridge, Southbridge, (and older days, FPU, MMU, etc)
- System-on-a-Chip (SoC): All parts of a computer on-chip
  CPU, DSP, memory, timers, USB, voltage regulators, memory controllers
- System-in-Package (SiP): various chips in one package

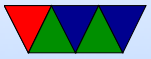


Extra Features of Embedded Micro-controllers

- Parallel and Serial I/O
- A/D, D/A converters
- GPIO pins
- i2c, CAN, SPI, 1-wire, USB busses
- FPGA?
- Low-power
- Sound
- Video, GPU

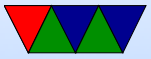


- DSP, Video Codecs
- Timers, PWM



ASIC, FPGA, Micro-controller

- ASIC – Application Specific Integrated Circuit
  direct wiring of state machines / logic on silicon die
- FPGA – reprogrammable low-level logic
- Microcontroller – can do what above do, but in software
- Why use the much more complex latter? Cost. Time to market. Bug-fixes (easier to fix in software)



Tradeoffs

It's all about tradeoffs

- Power
- Performance
- Cost
- Compatibility
- Time to Market
- Features