ECE 471 – Embedded Systems
Lecture 18

Vince Weaver

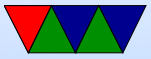
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

13 October 2021

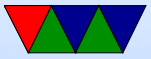
Announcements

- HW#6 still not posted yet.k Soon.
- Midterm on Friday, the 15th



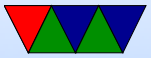
Midterm Notes

- The midterm will be in-person during class time
- Closed book/notes but you are allowed one page (8.5"x11") full of notes if you want

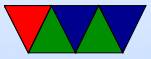


Midterm Content

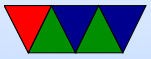
- Be sure you know the characteristics of an embedded system, and can make an argument about whether a system is one or not.
  - Inside of something (embedded)
  - Fixed-purpose
  - Resource constrained
  - Sensor I/O
  - Real time constraints (if you use this, be sure you can explain)



- Benefits/downsides of using an operating system on an embedded device
  - Benefits: "Layer of Abstraction"
  - Downsides: overhead, timing
- C code
  - Have you look at some code and know what it is doing
  - Mostly know what file I/O, loops, usleep, open/ioctl (things we've done in the homeworks)
- Code Density
  - Why is dense code good in embedded systems?
- GPIO & i2c



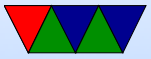
- Know some of its limitations (speeds, length of wires, number of wires, etc)
- Don't need to know the raw protocol
- Know the Linux interface (open, ioctl, write) and be familiar with how those system calls work



HW#5 Code Notes

- Datasheet Notes
  - What does 'X' mean in this context? (don't care)
  - Bits 15-8 was confusing, it's because we can ignore bits 7-0 (the i2c address and r/w) as Linux sends those for us
- Constructing constants notes
  - Enabling oscillator. If want value 2 in top 4 bits, 1 in bottom 4?

    `(0x2<<4)|(0x1)`

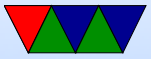


- Can we use hex or binary notation?

  The shifts make it more explicit what's going on, compiler will optimize for you
- "Magic Constants", you might instead want to do something like

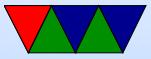
```
#define HT16K33_OSCILLATOR_ON (0x2<<4)|(0x1)  // p42 of datasheet
buffer[0]=HT16K33_OSCILLATOR_ON;
```

- i2c addresses vs registers (separate device, they aren't memory addresses). What if you use address 0?



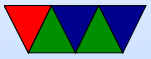
HW#5 Review – Questions

- Raspberry Pi boot odd: GPU does it. Why? Originally the chip was designed to be mostly GPU.
  sd-card is mildly unusual but not as unusual as GPU
- Program that loads kernel and jumps to it is called the bootloader
  Not start.elf. Not an init script. Not the firmware.
- Fat32: gave lots of good reasons for Fat32, but the reason boot partitions often use it is it's simple enough to be read by firmware at extreme early boot. Q wasn't



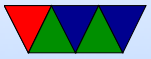
why FAT32 vs FAT16 Licensing fees

- Skipped i2c – those addresses are reserved.
  For various things, not just “future purposes”
  what happens if you have a device living at addr0?



HW#5 Review – Linux

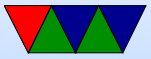
- wc, diff, piping
- You may have seen this all before in ECE331
- diff – used when making patches, also git diff
  Ask for wc -l which just shows lines. Can also show words, chars



i2c Reserved Addresses

Address	R/W Bit	Description
000 0000	0	General call address
000 0000	1	START byte (helps make polling cheaper)
000 0001	X	CBUS address
000 0010	X	Reserved for different bus format
000 0011	X	Reserved for future purposes
000 01XX	X	Hs-mode master code
111 10XX	X	10-bit slave addressing
111 11XX	X	Reserved for future purposes

10-bit addresses work by using special address above with first 2 bits + R/W, then sending an additional byte with the lower 8 bits.



Apple II Demo