ECE 471 – Embedded Systems
Lecture 17

Vince Weaver

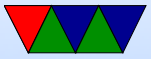
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

7 October 2022

Announcements

- Project info coming soon.
- HW#5 was due
- No HW due next week
- Midterm is a week from today, the 14th
- Review on Wed
- No class Monday (Fall Break)

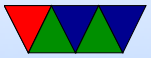


Midterm Info

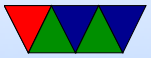
- In class
- Closed books, closed notes

  Can have one page of notes, 8.5"x11", one-sided
- Rough outline of things covered
  - Characteristics of embedded system
  - Benefits of having OS
  - Be able to read C code and know what it's doing
  - GPIO and i2c

    know the limitations

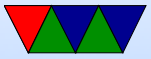


be familiar with Linux C code for accessing



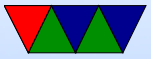
Firmware

- What is firmware?
- Code that runs on an embedded system
- Traditionally is a binary “blob”
- Often in ROM or Flash and can be hard to update



Device Firmware

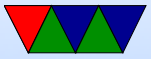
- Devices are their own embedded systems these days. May even have full CPUs, etc.
- Need to run code. Firmware.
- In ROM? Or upgradable? Why might you want to upgrade? (bug fixes, economy, etc.)
- Talk about recent USB firmware malware



Boot Firmware

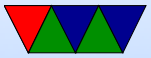
Provides booting, configuration/setup, sometimes provides rudimentary hardware access routines.
Kernel developers like to complain about firmware authors.
Often mysterious bugs, only tested under Windows, etc.

- BIOS – legacy 16-bit interface on x86 machines
- UEFI – Unified Extensible Firmware Interface
  ia64, x86, ARM. From Intel. Replaces BIOS
- OpenFirmware – old macs, SPARC
- LinuxBIOS



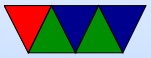
Binary Blobs

- What is in the firmware?
  Can you modify it yourself?
- Can it contain a full operating system?
  ThreadX on Pi, Minix on Intel servers?
- Hardware in flash, completely software upload at runtime



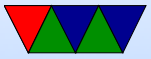
Firmware Licensing Issues

- Is a computer system truly free software if binary blobs running (like on a Pi)
- Debian tried to have a firmware-free install by default, but had to give up as not practical (especially at install)
- If someone ships firmware that has GPL code in it, what are their responsibilities?



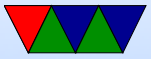
Trusted Firmware

- Firmware can be dangerous: runs below/outside of the Operating System, doesn't matter how secure your OS is if firmware compromised
- Can you trust your firmware to be not-evil?
- Evil Maid problem – what if someone breaks into your hotel room and replaces your firmware – could you tell?



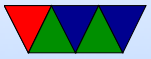
Signed Firmware

- Best you can do is trust it to be the same firmware released by your vendor (you still have to trust them)
- Use cryptographic signing. Hardware will only run code "signed" by a trusted entity.
- A signed firmware can run a signed bootloader which can run a signed operating system which can run signed apps



Signed Firmware Tradeoffs

- Downside: no longer general purpose, average person cannot run code they wrote unless they can get it signed
- Code still has to be well written. "jailbreaks" on phones and video game consoles are due to trusted code having bugs and then jumping into unsigned code.
- Will you still be able to run Linux?
  Trust Microsoft to keep signing bootloader for us?
- Walled gardens, restricted App stores (see Apple / EPIC lawsuit)



Firmware on Desktop/Laptop Systems

- What is the Pi GPU doing?
- What about the T2 processor on macs?
- New for ARMv8: ARM Trusted Firmware (ATF). Two standards, vendors have possibly made a mess of it already.
- Other platforms have it too. DRM to keep you from copying movies or video games.
- Windows 11 requiring TPM2 module