ECE 471 – Embedded Systems
Lecture 33

Vince Weaver

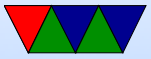
https://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

1 December 2025

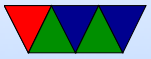
Announcements

- Don't forget course evals
- I sent out HW#6 grades
- I send out the preliminary project schedule
- Remember to return parts
- Will discuss final Wednesday
- Will give sample project presentation Wednesday
- Don't forget HW#10 (Power) due Friday. Might not have time to go over in class but will post solutions

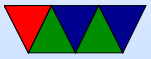


Aside on HW#6

- Decoding video? What makes "firm" real-time? It's about the results being useless after the deadline missed
- A bit different than video game degraded framerate. Video game constantly monitoring delay and if can't keep up adjusts the physics engines, etc, to use different deltaT
- Compressed video stream like MPG each frame depends on frame before. If one is late not only is it useless but the next frame possibly gets corrupted too (you've

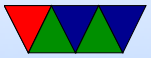


maybe seen this on TV when antenna gives out and you get weird boxy glitches all over screen)
Can lose sound sync too
Can actually see what happens when not hit 60fps. “stream killer” video on twitch, things like random static or lots of movement



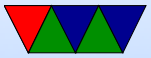
Project Update

- I sent out tentative project schedule.
- If you need to change date let me know, might be possible
- Also the order by day is arbitrary, usually I ask for volunteers
- Aim for about 7 minutes (5 presentation+2 demo/questions)
  might have to make this less on Friday
- Will give example presentation Wednesday

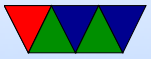


Ethics in Embedded Systems

- In the old days your washing machine was simple and designing a controller for it didn't involve a lot of ethical choices
- These days all kinds of things can go wrong
  - Should you track info like loads done and type of loads? Upload this data to the cloud? Are there privacy issues if that leaked?
  - Should you write a firmware update that starts displaying ads on the display?



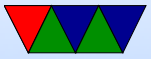
- Should you write code that blocks using any but manufacturer approved laundry detergent?
- Should you implement “laundry as a service” with an expensive subscription that’s hard to cancel?



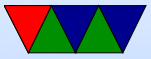
Ethics in Software Engineering

- How do you define it?
- There's an IEEE document:

  `https://www.computer.org/education/code-of-ethics`
- It's long, a few bullet points
  - Public interest (be ethical to public)
  - Client+Employer (do best for company, but not if hurts public)
  - Product (make best product you can)
  - Judgement+Integrity

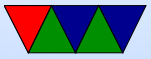


- Management (how you develop your software)
- Profession
- Colleagues
- Self

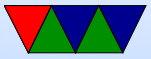


Finding Ethical Employment

- What if the only jobs are defense contractors?
- What if the only jobs are AI related?
- What if you get a job at a FAANG but it turns out to be ad/tracking software?

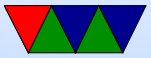


What Follows are some Examples where Computer Engineers might run into Ethical Dilemmas



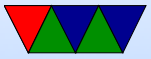
Dark Patterns in Interfaces

- Making it easy to accidentally do bad/expensive things
- Making it hard to cancel or close windows
- Make interfaces confusing to trick people
  - Swap the location of OK and Cancel buttons
  - Make OK a huge button, Cancel a tiny x off to the corner
- Have you ever tried to install Windows w/o a Microsoft account?

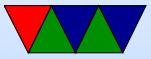


Privacy / Tracking

- Privacy? Data Logging? Tracking?
- Tracking: should smart TV's snoop on what's on the screen and report back to advertisers?
- Doorbell Cameras, other cameras – stored in cloud, should they be turned over to police
- Smart assistants – are they listening all the time? What do they do with what they hear?
- Many computer companies make most of their money by tracking and logging user activity and selling it

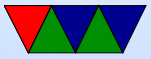


- Is this wrong?
- Probably encountered the annoying tracking buttons on websites, that's mostly because of the European GDPR privacy law



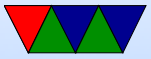
Accessibility

- Is your project usable by someone with a disability?
- What if your boss says it's too expensive to make your project accessible? (Title II: what if easier to just remove all material than make it accessible?)
- Examples
  - Interface works with screen reader
  - Interface visible to those with color blindness
  - No flashing lights
  - Will long-term use cause health problems, eye-strain,



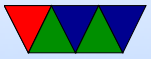
carpal tunnel

- Does game play attempt to cause addiction (gambling, loot crates, etc)



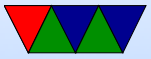
Citing Sources

- Properly citing sources/giving credit
- Properly licensing code
- Do AI companies do this properly?

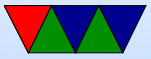


Ethics / Privacy Examples

- Unintentional security leaks: fitness trackers giving away military locations
- Thermostats: forget to change password if move or divorce, others now control your heating
- Amazon/Google devices, always listening in your house
- Get your youtube account banned, locked out of your google home, can't even contact a human to protest
- Web-cameras everywhere
- AI tracking all license plates

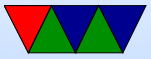


- Printers that won't let you use replacement cartridges
- Electric Toothbrushes that loudly beep if they think your brush head has expired
- Creating booby-trapped embedded systems (Lebanon pager incident)
- AI telling militaries which targets to blow up



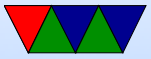
Can you control who uses your software?

- What if you have open-source code but someone uses it for evil purposes?
- What if you are a contractor and someone hires you to write evil code?



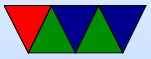
Reporting Ethics Violations

- Who can you report them to?
- What about the IEEE/ACM who claim they have ethics hotlines?
- They are manned by lawyers and possibly won't do anything if they think they might get sued
- See the Huixiang Chen / ISCA incident



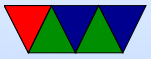
Cynical Take

- Being ethical is always a good idea but it might not make you many friends
- Being a whistleblower can unfortunately have negative career consequences



Example: FTDI USB/Serial Bricking Incident

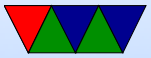
- Is cloning / counterfeiting popular chips wrong?
- What if you steal their USB-ID/ Trademarks?
- What if they fight back by having their driver disable counterfeit devices
- How do you feel if you didn't realize you had a counterfeit chip and suddenly your important embedded device stops working
- Who is to blame? Who are you likely to blame?



USB

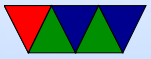
- Universal Serial Bus
- Designed to replace all of the various cables on a PC with one type (keyboard, mouse, printer, serial, SCSI, joystick)
- How successful was that?
- Way more complex than most previous interfaces

`http://www.usbmadesimple.co.uk/index.html`

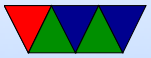


USB Power Delivery

- Originally could provide 5V at 100mA, and a device after being connected could request 5V at 500mA (2.5W)
- Older Pis this was fine, Pi3 just barely enough (get lightning bolt from the GPU usually because of voltage sag)
- Some devices draw more without asking (and some charges can provide it). Some devices had a magical set of resistors between pins that chargers could detect to enable faster charging

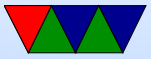


- Pi4 needs USB-C, but not implemented 100% right and might not work with all chargers
- Pi5 USB-C wants 5V@5A but that's a somewhat unusual configuration (5V@3A more common) so can also have trouble



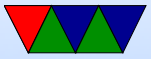
USB 1.0

- 1996
- Low Speed
  - 1.5Mbit/s (keyboard, mouse, etc)
  - Thinner, flexible cable
- Full Speed
  - 12Mbit/s (disk, USB key)
- USB 1.1 – ?

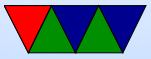


USB Physical Layer

- 2-5m cables
- 4 pins. 5V, GND, D+, D-. Differential signaling (subtract). More resistant to noise.
- Micro connectors have extra pin for on-the-go (says if A end or B end gnd vs v+)
- Unit load, 100mA. Can negotiate up to 500mA (2.5W) (more USB 3.0)
- Up to 127 devices (by using hubs). Up to 6 levels of hubs. Powered vs not.

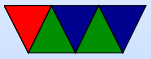


- Enumeration, vendor and device
- Connectors. A/B. Designed so only one end goes to host. Micro, mini.



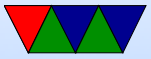
USB 2.0

- 2000
- High Speed 480Mbit/s



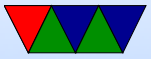
USB 3

- USB 3.0 – 2008 – SuperSpeed 5GBit/s (though hard to hit that) Full Duplex (earlier half duplex)
- USB 3.1 – 2014 – SuperSpeed+ 10Gbit/s
- Backwards compatible, has 5 extra pins next to standard micro with GND, SSTX+/- and SSRX+/- (full duplex)
- Connector often blue

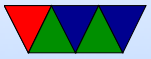


USB C

- USB-C – 2014
- 24-pin
  - 4 power/ground pairs
  - two differential non-super-speed pairs,
  - four pairs of high-speed data bus
  - two sideband pins,
  - two pins for cable orientation
- cables can be USB2, USB3, USB3.1
- up to 5A(20V=100W) but 3A more common

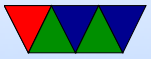


- wrong pullup can cause cable that damages hardware



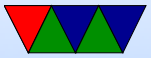
USB 4

- 2019
- Requires USB-C cable
- Can carry displayport, PCIe
- Support in Linux 5.6

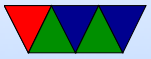


USB 1.1 and 2.0 Signaling

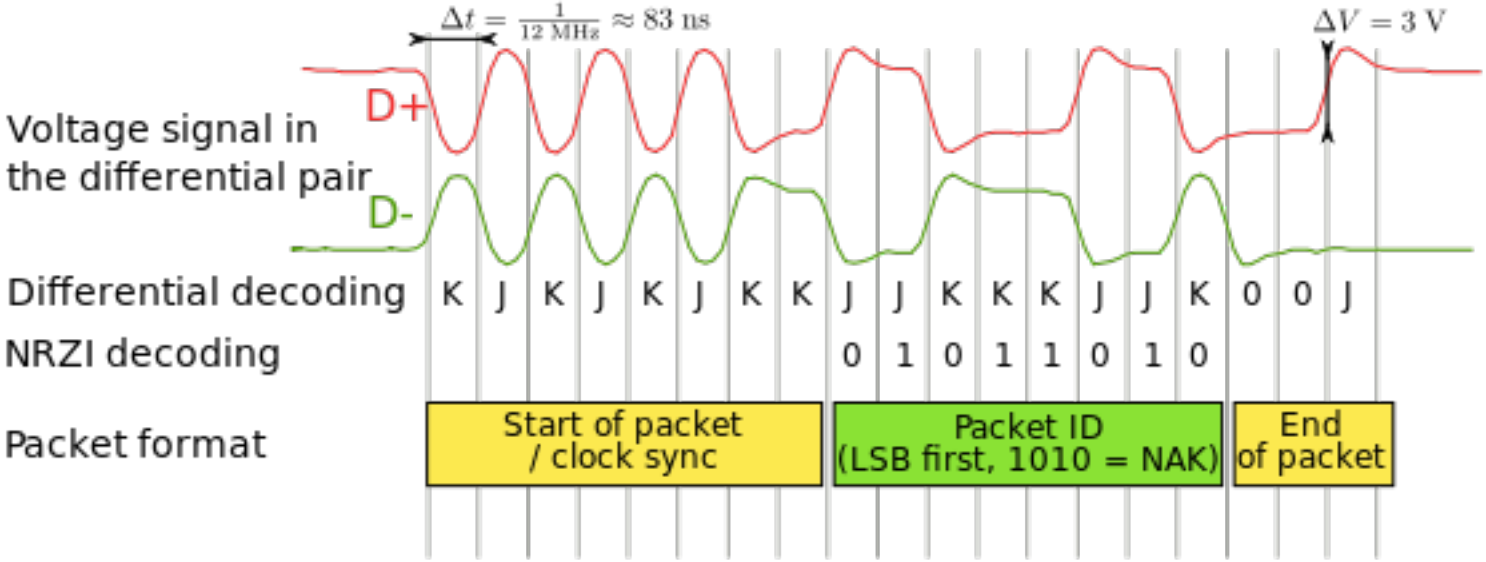
- Differential signaling, twisted pair, 90Ohm impedance
- Low+Full = 0V low, 3.3V high, not terminated
- High = 0V low, 400mV high, terminated with resistor
- Device Detection
  - Host, 15k pulldown pulls data lines to 0 (nothing connected, SE0)
  - USB device pulls line high with 1.5k which overpowers pulldown. Full bandwidth D+ high, low bandwidth D- high

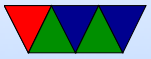


  - Some chargers use special resistors across D lines to indicate power they can draw.
- J and K states.
- NRZI line coding – 0 signaled by J to K (switching state). 1 signaled by leaving as is
- Bit stuffing – after six consecutive 1s must include 0
- starts with 8 bit synch – 00000001 which is KJKJKJKK. Data then sent. End marked by 00J.
- Reset by 10ms SE0
- Highspeed uses "chirping" to negotiate speeds, during reset chirps J and K



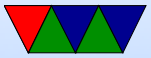
- Example from Wikipedia CC0:





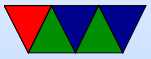
USB 3.0 Signalling

- SuperSpeed, separate lines, but original lines used to config
- SuperSpeed uses 8b/10b encoding (limits bandwidth), CRC, other features
- SuperSpeed+ uses 128b/132b encoding

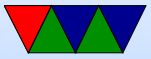


Latency

- For Low and Full shortest transaction time is 1ms. Can this be a problem?
  - Low latency gaming keyboards `https://danluu.com/keyboard-latency/`
  - Keyboard latency (scan keys, report keycode)
  - USB latency to system
  - Interrupt latency
  - OS update screen
  - LCD monitors often buffer a few frames

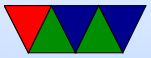


- Old Apple II (1MHz 8-bit) could go from keypress to screen faster than modern keyboards even finish scanning



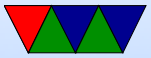
USB Protocol

- Various packets sent
  Huge list of them
- Checked by CRC
- Low power suspend mode, no more than 2.5mA
- Signal sent to all devices on USB bus, all but addressed one ignores



USB Design

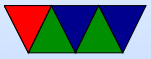
- Each device has endpoint
- isochronous – guaranteed data rate but with some potential data loss (video)
- interrupt – low-latency, like keyboards
- bulk – disk access



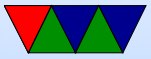
USB Linux

- Linux drivers
  - Device classes – HID, audio, etc. One common driver can handle all devices of a class
  - Specific – device driver is board specific and must have a list of all vendor/device IDs that are supported

- libusb
  Allow direct userspace access to USB interface
  Used by low-level things that might not need driver

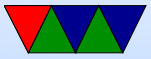


old cameras (not standardized), custom hardware



USB on Rasp-pi

- Pi4 has USB 3.0
- USB-OTG – on the go. Allows device to act like a host (so can hook up devices as per normal) or as normal USB device. Decides which based on whether A or B cable plugged in, check ID pin (micro/mini have 5th pin)
  The Pi-B does not support running in gadget mode externally (a hub in the way) and the OTG hardware requires more software support than (it is simpler) than



regular USB.

- USB 2.0 (sorta). Cannot supply full power (why? Only 1A power supply typical). Also cannot handle high-bandwidth things like audio cards and USB-cameras well.

- USB-host – standard USB port. Cannot provide high current, so use a powered hub if using anything more than keyboard or mouse