ECE 471 – Embedded Systems
Lecture 29

Vince Weaver

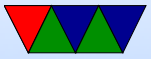
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

20 November 2017

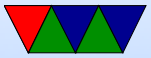
Announcements

- No class Wednesday (Thanksgiving)
- Remember your project status reports, due today
- Hand back and review Midterm
- HW#9 + HW#10 will be graded
- HW#11 will be coming eventually

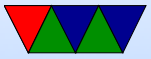


USB 1.1 and 2.0 Signaling

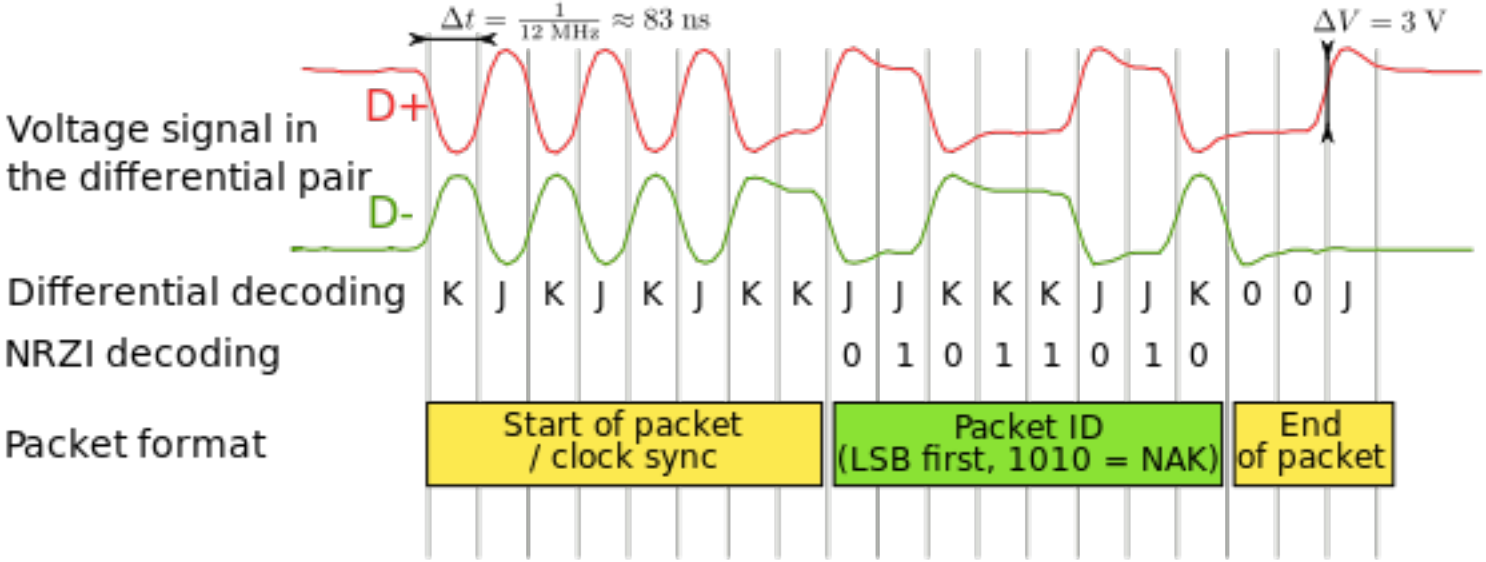
- Differential signaling, twisted pair, 90Ohm impedance
- Low+Full = 0V low, 3.3V high, not terminated
- High = 0V low, 400mV high, terminated with resistor
- Device Detection
  - Host, 15k pulldown pulls data lines to 0 (nothing connected, SE0)
  - USB device pulls line high with 1.5k which overpowers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

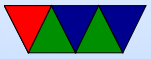


  - Some chargers use special resistors across D lines to indicate power they can draw.
- J and K states.
- NRZI line coding – 0 signaled by J to K (switching state). 1 signaled by leaving as is
- Bit stuffing – after six consecutive 1s must include 0
- starts with 8 bit synch – 00000001 which is KJKJKJKK.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



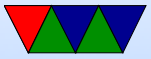
- Example from Wikipedia CC0:





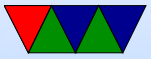
USB 3.0

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

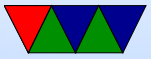


Latency

- 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

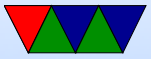


- Old Apple II (1MHz 8-bit) could go from keypress to screen faster than modern keyboards even finish scanning



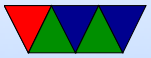
USB Protocol

- Various packets sent
- Checked by CRC



USB Design

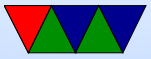
- Each device has endpoint
- isochronous – guaranteed data rate but with some potential data loss (video)
- interrupt – low-latency, like keyboards
- bulk – disk access



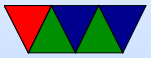
USB Linux

- 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



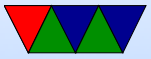
old cameras (not standardized), custom hardware



USB on Rasp-pi

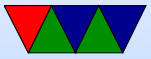
- USB-OTG – on the go. Allows device to act like a host (so can hook up devices as per normal) or as normal USB device. Decides which based on whether A or B cable plugged in, check ID pin (micro/mini have 5th pin)

  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.



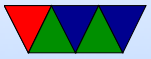
- USB 2.0 (sorta). Cannot supply full power (why? Only 1A power supply typical). Also cannot handle high-bandwidth things like audio cards and USB-cameras well.

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



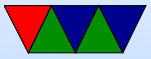
PWM

- Get around the fact that you can't get good timings w/o real-time OS
- Available on GPIO18 (pin 12)
- Can get 1us timing with PWM, 100us with Wiring Pi, probably less with GPIO interface.
- Which would you want for hard vs soft realtime?
- Other things can do? Beaglebone black as full



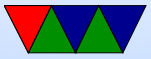
programmable real-time unit (PRU)

200MHz 32-bit processor, own instruction set, can control pins and memory, etc.

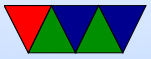


Audio Ports

- In the old days audio used to be just open `/dev/dsp` or `/dev/audio`, then `ioctl()`, `read()`, `write()`
- These days there's ALSA (Advanced Linux Sound Architecture)
  The interface assumes you're using the ALSA library, which is a bit more complicated.
- Pi lacks a microphone input, so if want audio in on your pi probably need a USB adapter.

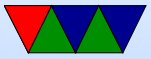


- Also can get audio out over HDMI.
- Pi interface is just a filter on two of the PWM GPIO outputs



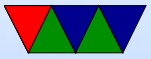
i2s

- PWM audio not that great
- i2s lets you send packets of PWM data directly to a DAC
- At least 3 lines. bit clock, word clock (high=right/low=left stereo), data
- Pi support i2s on header 5

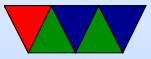


SD/MMC

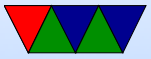
- MultiMediaCard (MMC) 1997
- Secure Digital (SD) is an extension (1999)
- SDSC (standard capacity), SDHC (high capacity), SDXC (extended capacity), SDIO (I/O)
- Standard/Mini/Micro sizes
- SDHC up to 32GB, SDCX up to 2TB



- Support different amounts of sustained I/O. Class rating 2, 4, 6, 10 (MB/s)
- SDIO – can have I/O like GPS, wireless, camera
- Patents. Need license for making.
- SPI bus mode
- One bit mode – separate command and data channels
- Four-bit mode
- 9 pins (8 pins on micro)



- Initially communicate over 1-bit interface to report sizes, config, etc.
- Starts in 3.3V, can switch to 1.8V
- Write protect notch. Ignored on pi?
- DRM built in, on some boards up to 10% of space to handle digital rights
- Can actually fit full Linux ARM server on a wireless SDIO card



- eMMC = like SD card, but soldered onto board