ECE 471 – Embedded Systems
Lecture 30

Vince Weaver

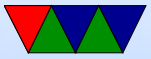
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

28 November 2018

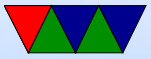
Announcements

- HW#10 was posted
- Feel free to return borrowed hardware.



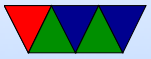
PWM Linux

- Available on GPIO18 (pin 12) (also GPIO19 on newer)
- Assuming 4.9 kernel or later (sets up proper clock) just edit config.txt and add `dtoverlay=pwm`
- reboot. lsmod should show `pwm_bcm2835` loaded
- `https://www.kernel.org/doc/Documentation/pwm.txt`
- sysfs interface
  - `cat /sys/class/pwm/pwmchip0/npwm`
  - `echo "0" > /sys/class/pwm/pwmchip0/export`
  - `echo "1000" > /sys/class/pwm/pwmchip0/pwm0/period`



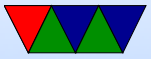
- `echo "500" > /sys/class/pwm/pwmchip0/pwm0/duty_cycle`
- `echo "1" > /sys/class/pwm/pwmchip0/pwm0/enable`

- Feel free to use wiring-pi or similar



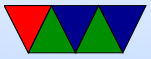
USB

- 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`



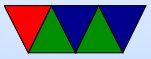
USB 1.0

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

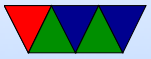


USB Physical Layer

- 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.

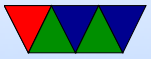


- Enumeration, vendor and device
- Connectors. A/B. Designed so only one end goes to host. Micro, mini.



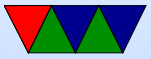
USB 2.0

- 2000
- High Speed 480Mbit/s



USB More Recent

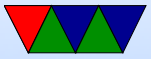
- USB 3.0 – 2008 – SuperSpeed 5GBit/s (though hard to hit that) Full Duplex (earlier half duplex)
- USB 3.1 – 2014 – SuperSpeed+ 10Gbit/s
- Backwards compatible, has 5 extra pins next to standard micro with GND, SSTX+/- and SSRX+/- (full duplex)
- USB-C – 2014
  24-pin: 4 power/ground pairs, two differential non-super-speed pairs, four pairs of high-speed data bus, two



sideband pins, two pins for cable orientation

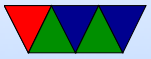
cables can be USB2, USB3, USB3.1, up to 5A(20V=100W) but 3A more common

wrong pullup can cause cable that damages hardware

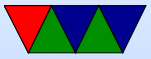


USB 1.1 and 2.0 Signaling

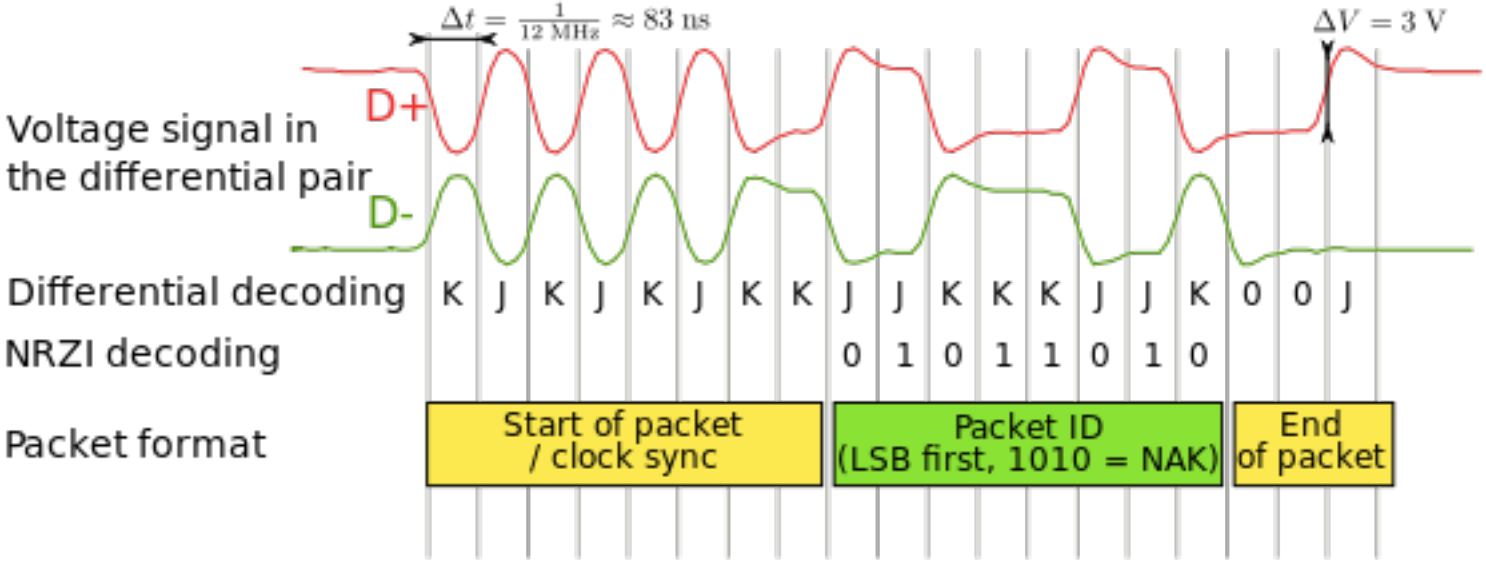
- Differential signaling, twisted pair, 90Ohm impedance
- Low+Full = 0V low, 3.3V high, not terminated
- High = 0V low, 400mV high, terminated with resistor
- Device Detection
  - Host, 15k pulldown pulls data lines to 0 (nothing connected, SE0)
  - USB device pulls line high with 1.5k which overpowers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

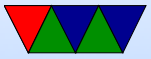


  - Some chargers use special resistors across D lines to indicate power they can draw.
- J and K states.
- NRZI line coding – 0 signaled by J to K (switching state). 1 signaled by leaving as is
- Bit stuffing – after six consecutive 1s must include 0
- starts with 8 bit synch – 00000001 which is KJKJKJKK.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



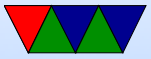
- Example from Wikipedia CC0:





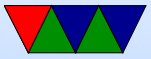
USB 3.0

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

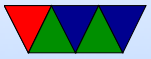


Latency

- 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

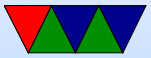


- Old Apple II (1MHz 8-bit) could go from keypress to screen faster than modern keyboards even finish scanning



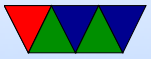
USB Protocol

- Various packets sent
  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

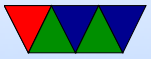
- Each device has endpoint
- isochronous – guaranteed data rate but with some potential data loss (video)
- interrupt – low-latency, like keyboards
- bulk – disk access



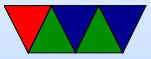
USB Linux

- 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



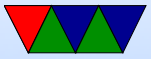
old cameras (not standardized), custom hardware



USB on Rasp-pi

- 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