

# ECE 471 – Embedded Systems

## Lecture 29

Vince Weaver

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

`vincent.weaver@maine.edu`

26 November 2018

# Announcements

- HW#9 will be graded
- HW#10 will be posted soon, a reading assignment
- Project status report due Friday. Not long:
  - A one-line statement of your project topic
  - A short summary of the progress you've made so far
  - List any parts you need that you don't have yet
  - List if you're willing to present early (Friday the 7th, Monday 10th or Wednesday 12th vs Friday the 14th) (there will be some bonus for presenting early)



# Hand Back and Go Over Midterm

- Average was an 88



# Wii Nunchuck

- Fairly easy to interface
- Put onto i2c bus. Device 0x52
- Send handshake to initialize. Use longer one (0xf0/0x55/0xfb/0x00) not the simpler one you might find(0x40/0x00). This works on generic nunchucks and possibly also disables encryption of results.
- To get values, send 0x00, usleep a certain amount, and read 6 bytes. This includes joy-x, joy-y, accelerometer



x/y/z and c and z button data. More info can be found online.

byte0 = joy-x, byte1 = joy-y, byte2 = top8 acc x, byte3 = top8 acc y, byte4 = top8 acc z, byte 5 is bottom 2 z,y,x then button c and z (inverted?)



# Linux and Keyboard

- Old ps/2 keyboard just a matrix of keys, controlled by a small embedded processor.  
Communication via a serial bus. Returns “keycodes” when keypress and release and a few others.
- Many modern keyboards are USB, which requires full USB stack. To get around needing this overhead (for BIOS etc) support bit-bang mode. OS usually has abstraction layer that supports USB keyboards same as old-style



- Linux assumes “CANONICAL” input mode, i.e. like a teletype. One line at a time, blocking input, wait until enter pressed.
- You can set non-CANONICAL mode, async input, and VMIN of 1 to get reasonable input for a game. Arrow keys are reported as escape sequences ( ESCAPE-[A for up, for example).
- Even lower-level you can access “RAW” mode which gives raw keycode events, etc.
- See the `tcgetattr()` and `tcsetattr()` system calls
- There are libraries like `ncurses` that abstract this a bit.



Also GUI and game libraries (SDL).





# Faking Linux Input Events

- How to insert input events into Linux, i.e. have a software program fake keyboard/mouse/joystick events.
- Linux supports a "uinput" kernel driver that lets you create user input.
- There is some info on a library that makes this easier here: <http://tjjr.fi/sw/libsuinput/>
- It has examples for keyboard and mouse. Joystick should be possible but there's no sample code provided.
- Python wrappers seem to exist too.



# System Busses

- Older busses often exposed CPU pins directly to connector: Apple II, S-100, ISA
- This was not sustainable, if only because number of CPU pins grew rapidly. Also speed issues.



# Parallel vs Serial Busses

- Originally most busses were Parallel. More bits at a time means higher bandwidth. IDE, Parallel Port, 32-bit PCI, 64-bit PCI
- Problems with parallel: keeping signals in sync. As busses go faster, skew comes into things. Wire length matters. Power issues with driving wide busses.
- Newer busses are serial: SATA, PCIe, USB, Firewire, etc. Also advantage of having fewer wires to route.
- HPC users grumble about speed of PCIe

