ECE 471 – Embedded Systems
Lecture 20

Vince Weaver

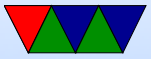
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

15 November 2016

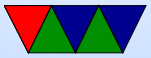
Announcements

- Project ideas! Should have gotten an e-mail
- Grades
- Where was everyone last class



HW#7 Review

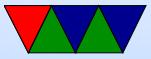
- Code
  - memset *before* you assign values.
  - Error Checking: errors, best if exit. Will take points off if it prints invalid temps even if it doesn't crash.
  - Be sure not to leak file descriptors
  - Why fprintf with %.21f? Oh trying to be 2LF
- Realtime Question:



- Hard – everyone OK
- Soft and Firm:
  Remember, only firm if data useless after deadline missed

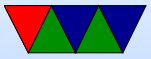
- SPI disadvantage vs i2c – no spec, no errors, more wires, mildly shorter distance, etc
  SPI advantage – can be faster, full duplex, lower power, multiple chip selects. bad chip bring down the bus?

- bitbang takes longer than hardware SPI, and especially if a context switch happens while bitbanging can take



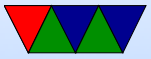
more than 1.2ms. You can bitbang this fast we decided (though maybe not with the sysfs interface)

- /dev/null – throws away the output
- /dev/full – can use to test error handling
- /dev/zero – can be used to make disk images, etc.
- /dev/random – "truly" random. Urandom is pseudo-random
  We had a long discussion about why you need random numbers and where they come from (cryptography)



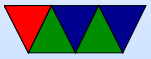
PWM

- 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



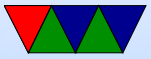
programmable real-time unit (PRU)

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

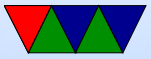


Audio Ports

- 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.

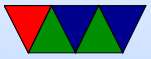


- Also can get audio out over HDMI.
- Pi interface is just a filter on two of the PWM GPIO outputs



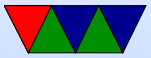
i2s

- 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

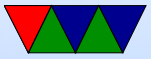


SD/MMC

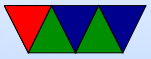
- MultiMediaCard (MMC) 1997
- Secure Digital (SD) is an extension (1999)
- SDSC (standard capacity), SDHC (high capacity), SDXC (extended capacity), SDIO (I/O)
- Standard/Mini/Micro sizes
- SDHC up to 32GB, SDCX up to 2TB



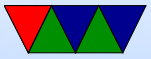
- Support different amounts of sustained I/O. Class rating 2, 4, 6, 10 (MB/s)
- SDIO – can have I/O like GPS, wireless, camera
- Patents. Need license for making.
- SPI bus mode
- One bit mode – separate command and data channels
- Four-bit mode
- 9 pins (8 pins on micro)



- 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

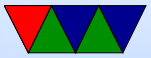


- eMMC = like SD card, but soldered onto board



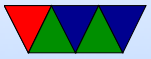
Camera Port

- The SoC has dedicated hardware for driving cameras
- 5megapixel, CSI port (Camera Serial Interface) plus i2c bus to command it.
- Can read data in parallel, directly, without needing USB overhead.
- These chips often used in cell-phones, so makes sense to have support for camera-phone without extra chip being needed.



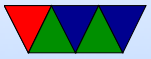
Ethernet

- Old, complicated standard, whole way up to 100GBps
- Modern form is often RJ-45, twisted pairs
- Power over ethernet (no pi support)
- Board has 10/100 Mbpss ethernet port
- Connected to on-board USB hub

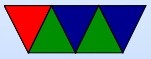


UART – serial port

- Often useful on embedded boards and old systems, might be only way to reliably connect
- RS-232, originally for teletypes
- 3-15V high, -3 to -15V low
- start/stop bits, parity, bit-size
- Hardware vs Software flow control

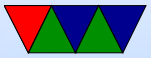


- Speeds 300bps - 115000bps and beyond
- 50feet (15m) w/o special cables
- 3-pin version (transmit, receive ground). Also 5-pin HW flow control (CTS/RTS). Can have 2-pin version if only want to transmit
- These days often hook up USB connector

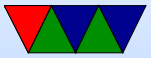


HDMI

- High-Definition Multimedia Interface (2003)
- Compatible with DVI (if no copy protection used)
- Video, audio (up to 8 channels), CEC (consumer electronics control), ethernet
- No support for captions
- DDC – i2c bus, used for EDID (getting device info) and HDCP (copy protection)

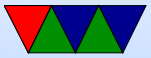


- TDMS – transition minimized differential signaling
  Video, then during scan line breaks, audio, etc
- CEC – control up to 15 devices with one remote control
  (one wire serial bus)
- Various versions, various fees

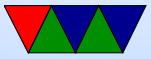


Other video ports

- NTSC
- VGA (analog)
- DVI
- Thunderbolt
- Displayport
- USB?

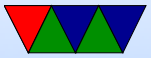


Other Busses not found until RPI-3



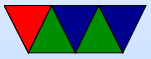
Wireless

- Wireless ethernet
- 2.4GHz or 5GHz

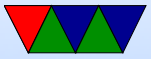


Bluetooth

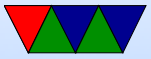
- Basic unit: piconet, master node and up to seven *active* slave nodes within 10m
- Many can exist in an area, and can be connected by a bridge. Connected piconets are called a scatternet
- There can also be up to 255 “parked” nodes in a picnoet
- When parked, can only respond to activation on beacon
- Hold and siff?
- Slaves designed to be cheap, so dumb. Master is smart and runs them. slave/slave communication not possible



- Master broadcasts clock 312.5us. Master transmits in even, slave in odd.
- Radio layer – 2.4GHz, 10 meters. 79 channels of 1MHz.
- pairing
- Bluetooth V1.1 has 13 different application protocols.
- Bluetooth 4.0 (Bluetooth Low Energy) (2010)
  - 25Mbps/200 feet
  - Entirely new stack, designed for low power rapid setup links
  - Not backwards compatible, but same frequency range
  - New profiles

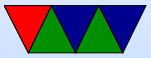


- Linux interface: depends on type. Filetransfer/obex. Audio (looks like an audio driver) network device, serial device



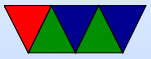
CANbus

- Automotive. Introduced by BOSCH, 1983
- One of OBD-II protocols
- differential, 2 wires, 1MBps important things like engine control
- single wire, slower cheaper, hvac, radio, airbags

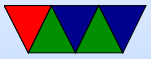


CANbus Protocol

- id, length code, up to 8 bytes of data id (usually 11 or 29 bits) type and who is sending it. Also priority (lower is higher) length is 4 bits. some always send 8 and pad with zeros
- Type is inferred from id. Can be things like engine RPM, etc
- DBC database has the ids and values. ASCII text database, hard to get legally.

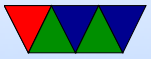


- Dominant/Recessive. Message with lowest ID wins arbitration.
- CAN-FD – extended version with larger sizes



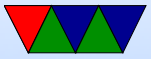
CANbus Linux

- Can4linux – `open("/dev/can0"); read(); write();` External project?
- SocketCAN – contributed by Volkswagen. In kernel. Uses socket interface. /Documentation/networking/can.txt



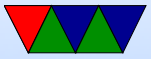
CANbus on Pi

- No

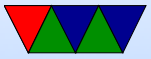


ISA Bus

- Introduced with IBM-PC in 1981
- 8-bit (4.77MHz) then 16-bit (8MHz)
- +/-5V, +/-12V, 8 data, 20 address, DMA, IRQ
- Replaced by VLB (more pins, extra header), EISA (double pins in same connector), MCA micro-channel (different proprietary from IBM)

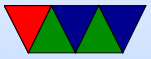


- Not enumerable at first, set jumpers. Later “Plug-n-Play”



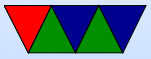
LPC Bus

- Low-pin-count bus
- Intel, 1998, try to get rid of ISA
- Things like PS/2, Serial ports, floppy, etc.
  Still used for TPM Trusted Computing nonsense
- Replace 16-bit 8.33MHz parallel bus with 4-bit wide 33.3MHz bus. Only 7 wires. Easier to route than 72



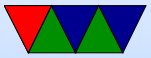
“Conventional” PCI Bus

- Peripheral Component Interconnect
- Enumerable
- 1993, intel
- 62-pins, parallel, 133MB/s
- Extended with 32 or 64-bit versions, 33 or 66MHz, 3.3 or 5V. All slight differences in connectors to support all that.



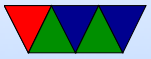
- AGP (Accelerated Graphics Card) for graphics cards. 1997. Direct connect to CPU (not shared), multiple channels, faster clock

- PCI-X 1998, extension to 133MHz. Not to be confused with PCI-Express (PCIe)



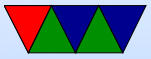
PCI protocol

- 256B Config space, mapped into CPU address. Small area system can probe, used to setup larger mappings
- Can have on-board ROM that can be executed. Problem when using on non-x86 systems (emulators needed? special [expensive] PowerPC versions?)
- Latency timers keep bus-master from hogging bus
- 4 interrupt lines, can be shared. Level rather than edge-triggered interrupts make sharing easier

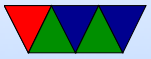


PCIe

- PCI-express, 2003
- Serial, replaced point-to-point with lanes, packet-based x1, x2, x4, x8 x16, x32
- Compatible with PCI at software level
- Differential Signaling
- External – Thunderbolt

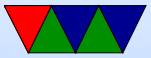


- Serial better due to timing skew
- New x86 audrino quark has PCIe



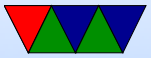
PCIexpress Mini

- PCIe x1, USB, SMBus, etc
- Smaller card



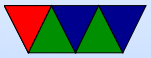
PCMCIA Bus

- Personal Computer Memory Control International Association
- 16-bit
- Cardbus, 32-bit
- Mostly replaced these days



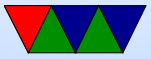
PC/104 Bus

- Stackable small x86 boards usually
- Run ISA or PCI signals up vertically



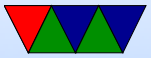
VME Bus

- m68k bus but generic enough
- Still found in some embedded systems



Other

- SATA, eSATA, PATA, SCSI (disk drives)
- Firewire
- RapidIO
- Quickpath QPI
- Hypertransport
- Thunderbolt (requested)



- List of competing busses at end of USB wiki article