ECE 471 – Embedded Systems
Lecture 14

Vince Weaver

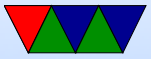
http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

20 October 2015

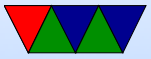
Announcements

- Any HW questions?
- Midterm review
- Project

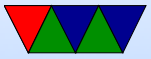


Midterm Review

- Very much like questions on HW.
- Definition of embedded system
  Say if a given system is an embedded system.
- Benefits of using an operating system / downsides
- Code density. ARM THUMB
- Real time

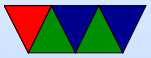


- i2c / gpio – characteristics. Also understand code like that from HW. Do not expect you to have entire protocol memorized, but should know it's open collector, etc.



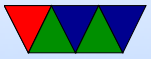
Pointer Review

- char buff1;
- char *buff2;
  buff2=&buff1;
  buff2=malloc(10*sizeof(char));
- char buff3[10];
- char **buff4;
- read(fd,&buff1,1);
- read(fd,buff2,10);
- read(fd,buff3,10);



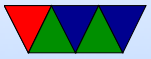
System Busses

- 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.

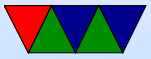


Parallel vs Serial Busses

- 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.

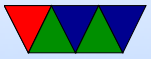


- People (especially HPC) still grumble about speed of PCIe



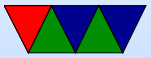
SPI bus

- Serial Peripheral Interface Bus
- Synchronous full-duplex serial bus named/formalized by Motorola. No real standard.
- What does synchronous mean? (Separate clock line)
- What does full-duplex mean? (Transmit and receive at same time)



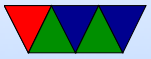
What used for?

- LCD displays
- Optional interface to SD cards
- LED strips
- JTAG

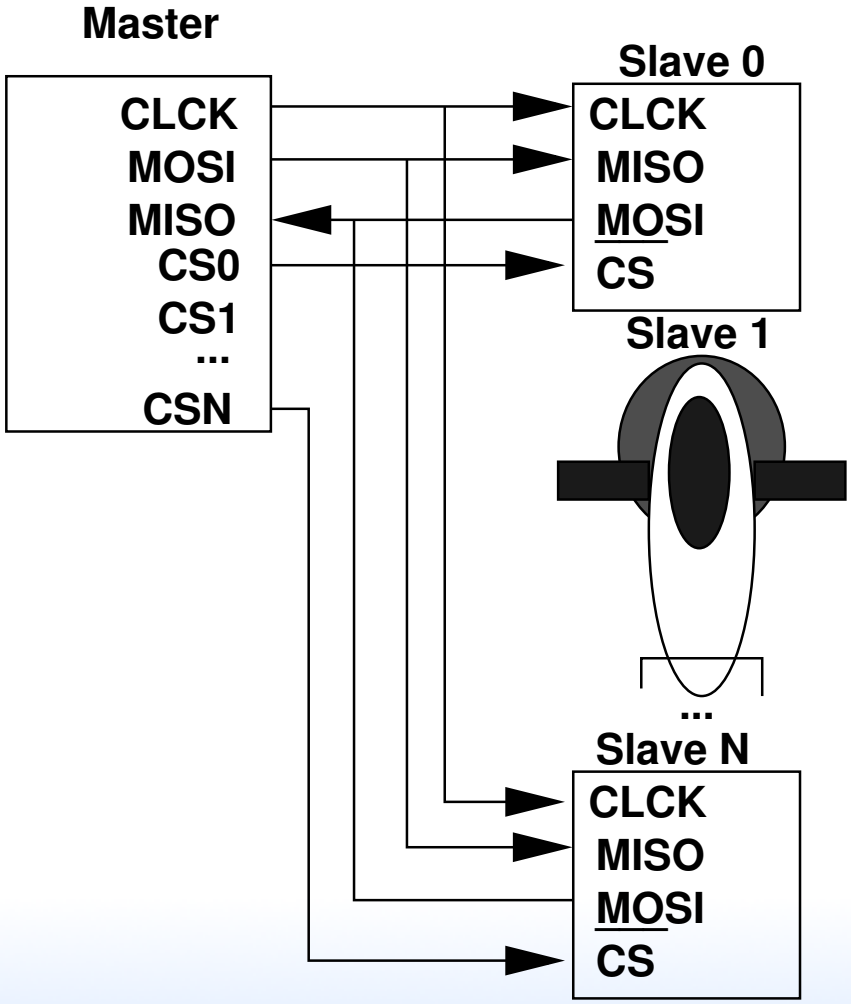


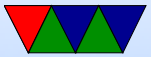
Hardware Setup

- Master/slave with multiple slave select lines
- 4-wire bus
- SCLK – serial clock (output from master)
- MOSI – master out, slave in
- MISO – master in, slave out
  Must be high impedance if more than one slave



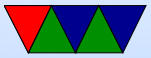
- CS0, CS1, etc – slave chip selects





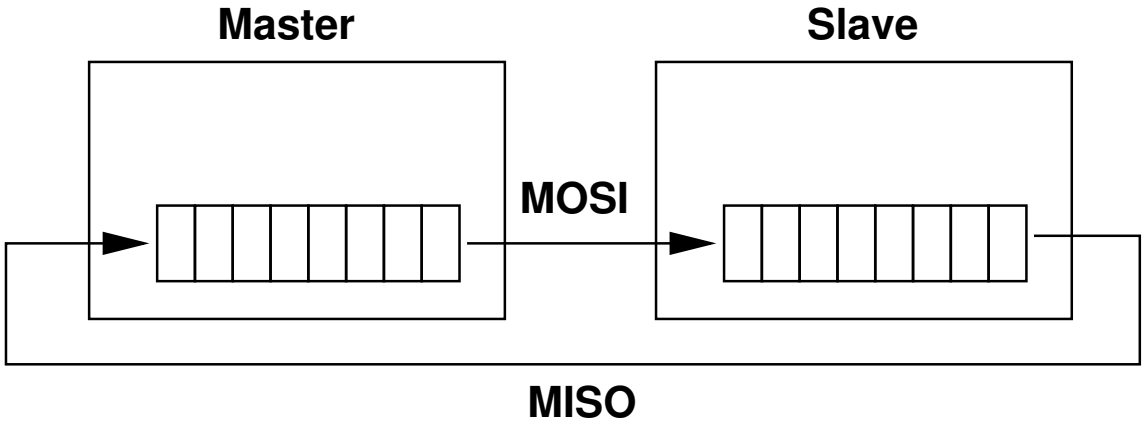
SPI protocol

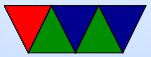
- Pulls chip-select of desired slave low
- Master starts clock
  No set speed, just what the slave can handle.
  Up to a few MHz
- Must both Send *and* receive (at same time over MISO/MOSI wires)
  Doesn't have to be useful data, but must be done both ways



- Master transmits data bits as long as it has it. When done turns off clock and maybe deselects slave.

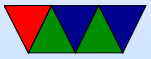
- It's basically just a shift register in the master and slave, and you rotate through enough bits to swap the values in each, then both sides can read out the transfer.



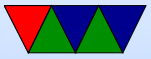


Clock Polarity/Phase

- Many have adopted Freescale's terminology
- CPOL=0 – base clock is zero
  - CPHA=0 – data captured on rising edge
  - CPHA=1 – data captured on falling edge
- CPOL=1 – base clock is one
  - CPHA=0 – data captured on falling edge
  - CPHA=1 – data captured on rising edge

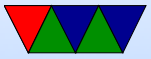


- Also given as “mode” numbers, 0 - 3. CPOL/CPHA. This can vary by manufacturer. Check your data sheet!



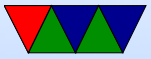
Connection

- “independent” – One slave per select line
- “daisy-chain” – MISO to MOSI, like long chain of shift registers, only need one SS line.



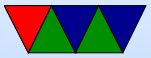
Interrupts

- Possible... think touch screens and such. Not officially specified



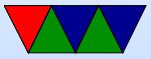
Errors

- No way to indicate errors
- Some chips will ignore if invalid data sent (wrong number of bits) some not

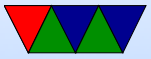


SPI advantages

- Full-duplex
- fast (no set speed limit)
- arbitrary message size in bits
- low power (no pullup resistors)
- Can be implemented with minimal hardware (just a 74HC495 shift register)

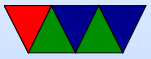


- no arbitration
- no unique ids
- unidirectional signals
- clock provided by master (no oscillator needed in slaves)

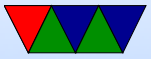


SPI disadvantages

- more pins (4 plus ground plus power plus one more each slave)
- short distances
- no flow control
- no error reporting
- no standard

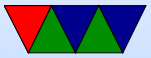


SPI vs i2c



SPI bus on Raspberry Pi

- SPI1 is on the header
- Pin 23 – SCLK
- Pin 19 – MOSI
- Pin 21 – MISO
- Pin 24 – CE0
- Pin 26 – CE1



- Unlike some boards, no nIRQ (SPI interrupt) pin