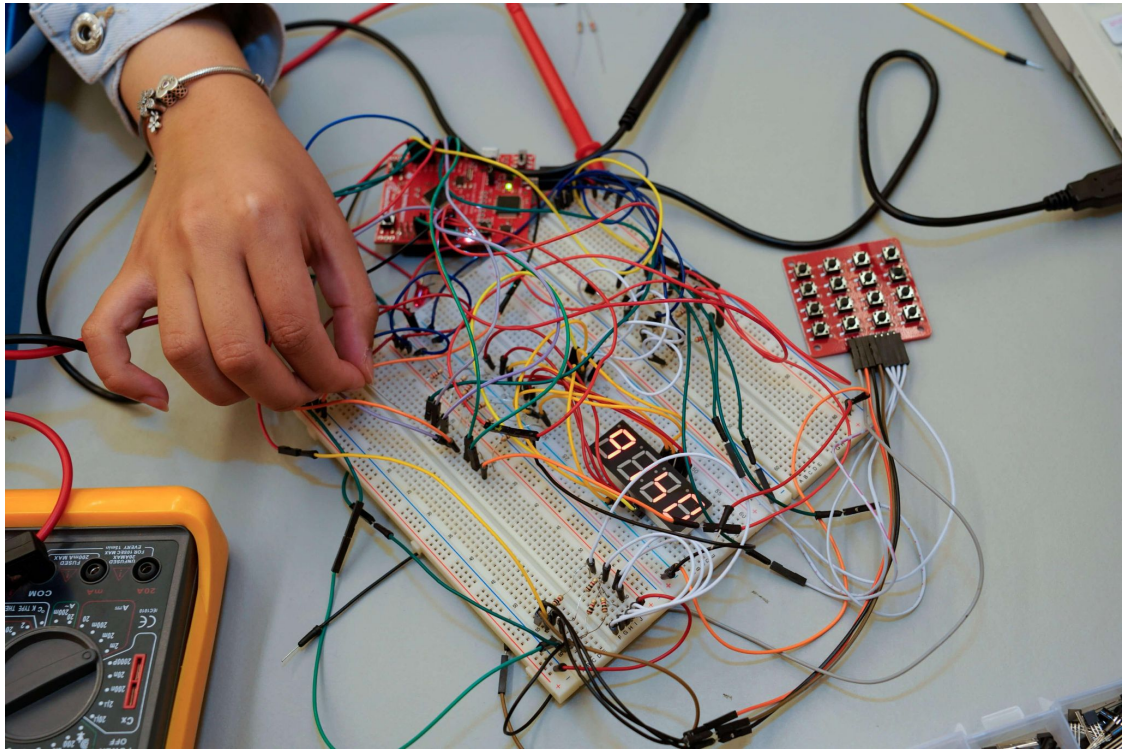


Communications

GEARS 2023

LEEMAN GEOPHYSICAL LLC
CONSULTING & INSTRUMENTATION

Image: American University in Cairo

Two devices can talk to each other using many different protocols

- CAN (controller area network)
- I2C (inter-integrated circuit)
- SPI (Serial Peripheral Interface)
- Serial
- HEART (Highway Addressable Remote Transducer)
- MODBUS
- MANY more!

SERIAL

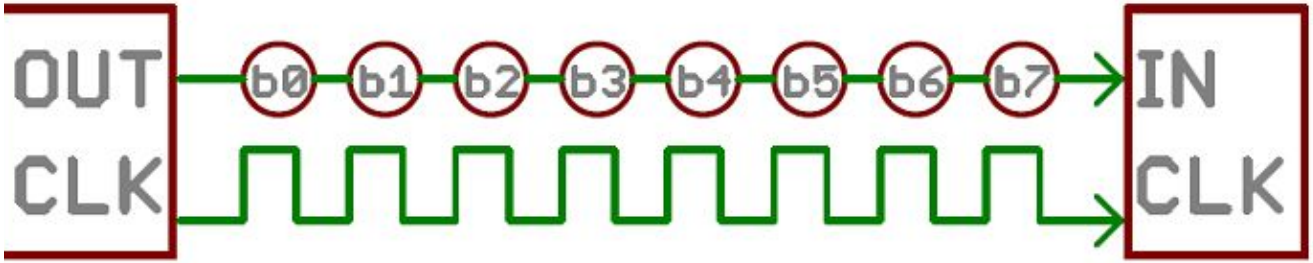
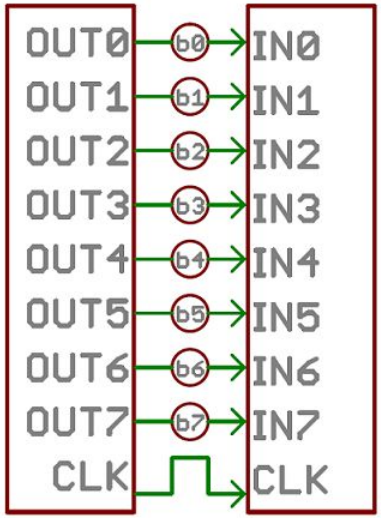


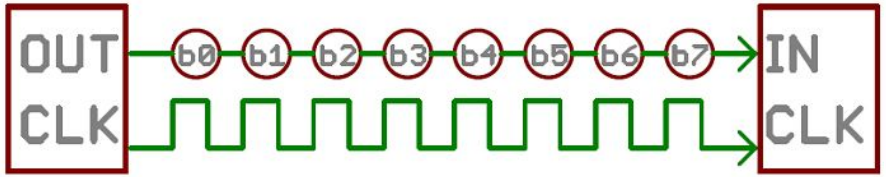
Image: Sparkfun

Serial is probably one of the most common ways to send data between devices

Parallel



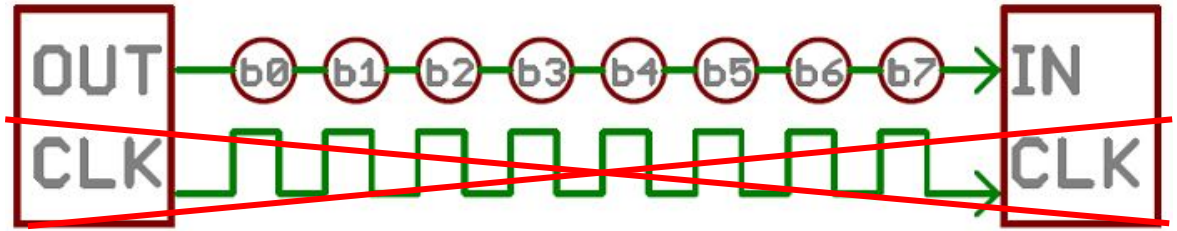
Serial



LEEMAN GEOPHYSICAL LLC
CONSULTING & INSTRUMENTATION

Image: Sparkfun

Asynchronous serial uses a predefined data rate to avoid the need for a clock signal, but this can only be done with a few tricks



- Data Bits
- Synchronization Bits
- Parity Bits
- Baud Rate

Image: Sparkfun

Baud Rate is expressed in bits per second (bps) and is how long each bit is held high or low. They are multiples of 300 bps

Common Rates

- 1200
- 2400
- 4800
- 9600
- 19200
- 38400
- 57600
- 115200

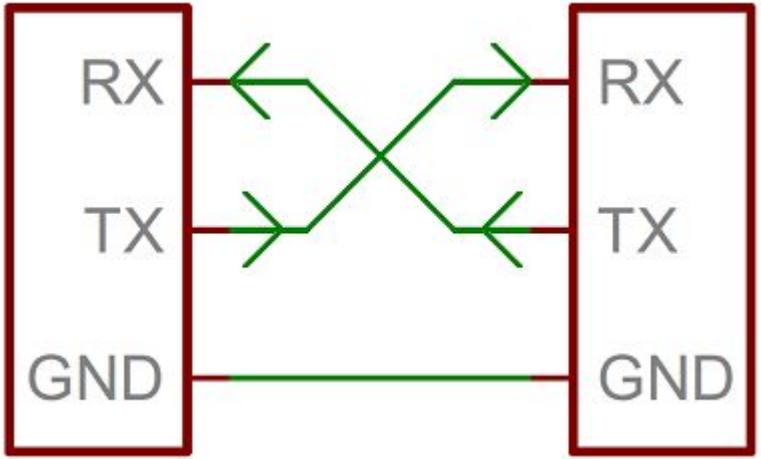
Image: Sparkfun

Data are put into “frames” with the synchronization features, commonly we use the 8N1 format

LEEMAN GEOPHYSICAL LLC
CONSULTING & INSTRUMENTATION

Image: Sparkfun

Transmit and receive are generally labeled with respect to the device itself, but there are better ways!



DTE = Data Terminal Equipment (Controller)

DCE = Data Communications Equipment (Peripheral)

Image: Sparkfun

Serial troubles generally fall into one of a few common issues

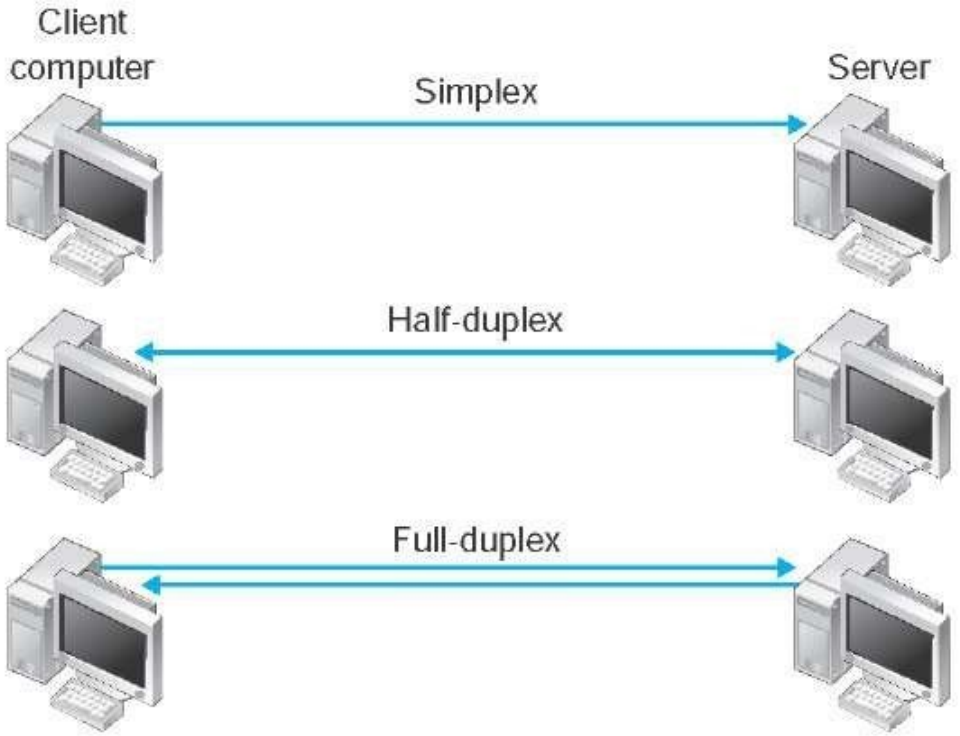
- RX/TX mix up
- Baud Rate Mismatch
- Controller Clock Mismatch
- Bus Contention (only one device per UART!)

The serial protocol can be implemented in hardware a variety of ways

- Logic level (short runs of a few meters)
- RS232 (differential for up to 15 meters)
- RS422/485 (up to 1200 meters)
- Ethernet
- USB

Image: Sparkfun

There are full-duplex and half-duplex devices



LEEMAN GEOPHYSICAL LLC
CONSULTING & INSTRUMENTATION

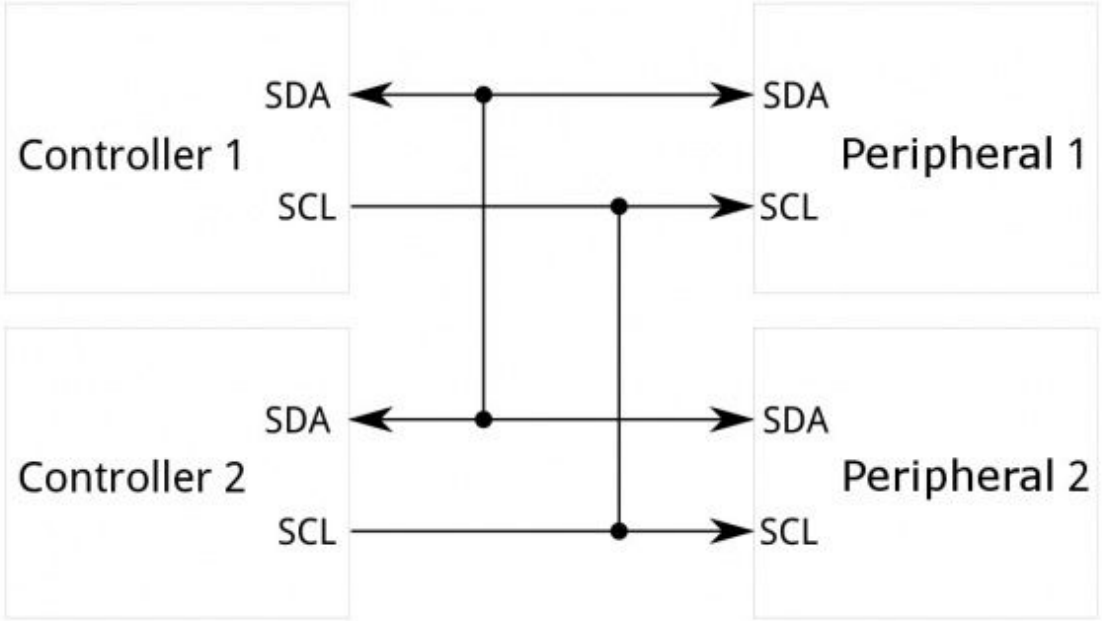
Image: 0x00sec

Serial

- Easy to implement
- Relatively Universal
- Different hardware implementations may work over long distances

- Generally slow rates
- Clocks may need to be adequately accurate
- Pre-agreed upon rates
- Lots of extra framing content

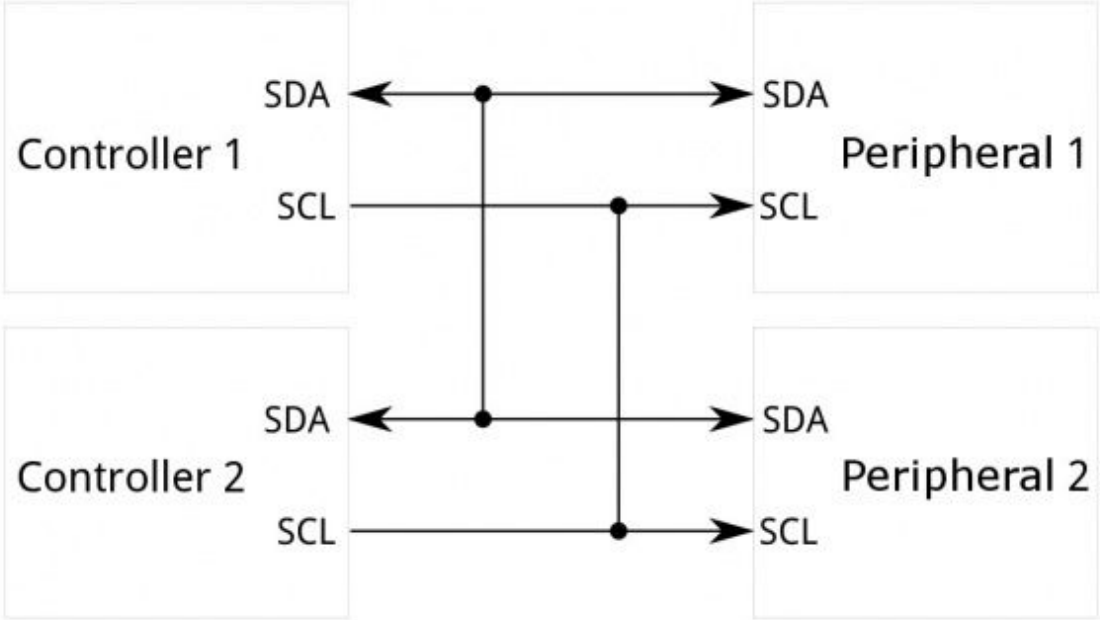
I2C



LEEMAN GEOPHYSICAL LLC
CONSULTING & INSTRUMENTATION

Image: Sparkfun

I2C can allow multiple peripheral devices to communicate with one or more controller devices with only two data wires



LEEMAN GEOPHYSICAL LLC
CONSULTING & INSTRUMENTATION

Image: Sparkfun

The SDA and SCL lines require pull ups as the devices can only pull a line low to prevent conflicts.

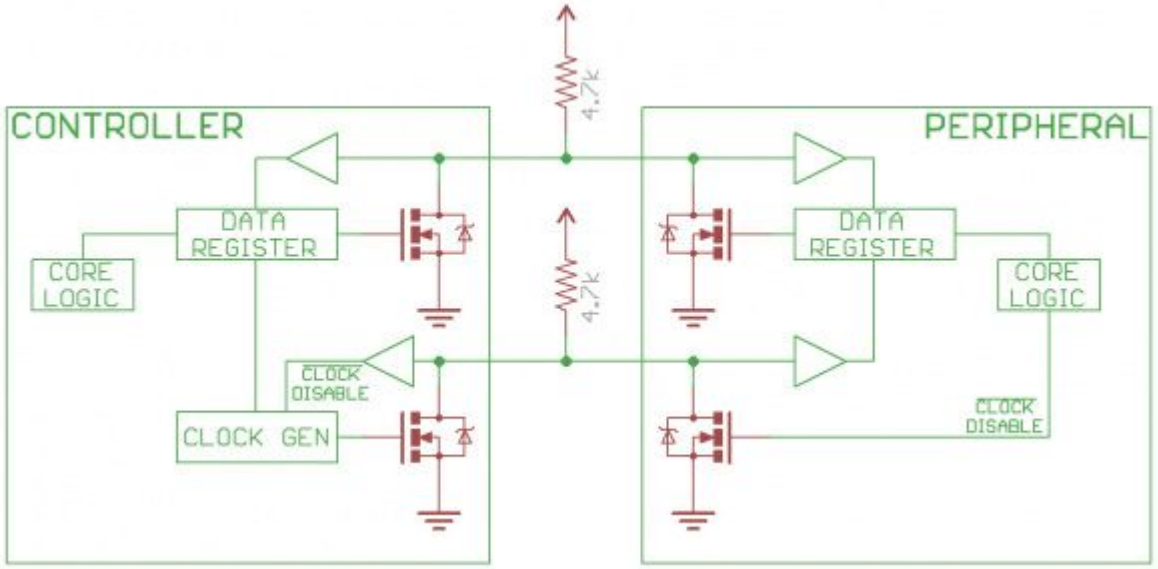
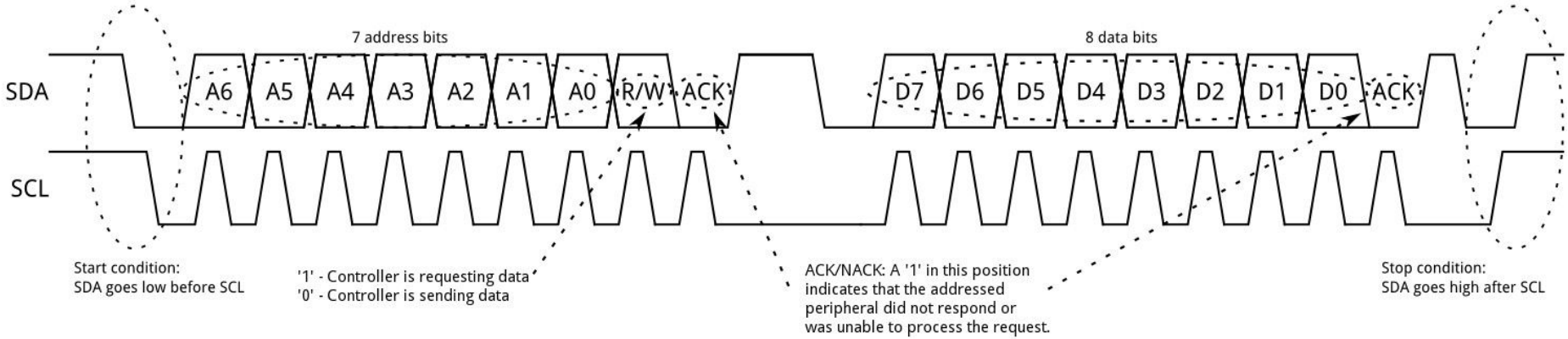


Image: Sparkfun

Data flows both ways on data wires with the device controlling the bus at that time controlling the clock wire



LEEMAN GEOPHYSICAL LLC
CONSULTING & INSTRUMENTATION

Image: Sparkfun

I2C (Inter-Integrated Circuit)

- Two wires only
- Devices are addressed
- No predetermined clock rate
- Simple to make in hardware
- Multiple peripherals and controllers

- Slower than other busses (simplex)
- Address conflicts limit number of devices

SPI

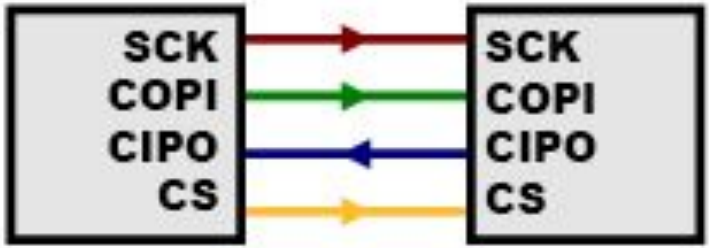


Image: Sparkfun

SPI (Serial Peripheral Interface) is a full duplex, synchronous method for short range communication between chip-scale devices

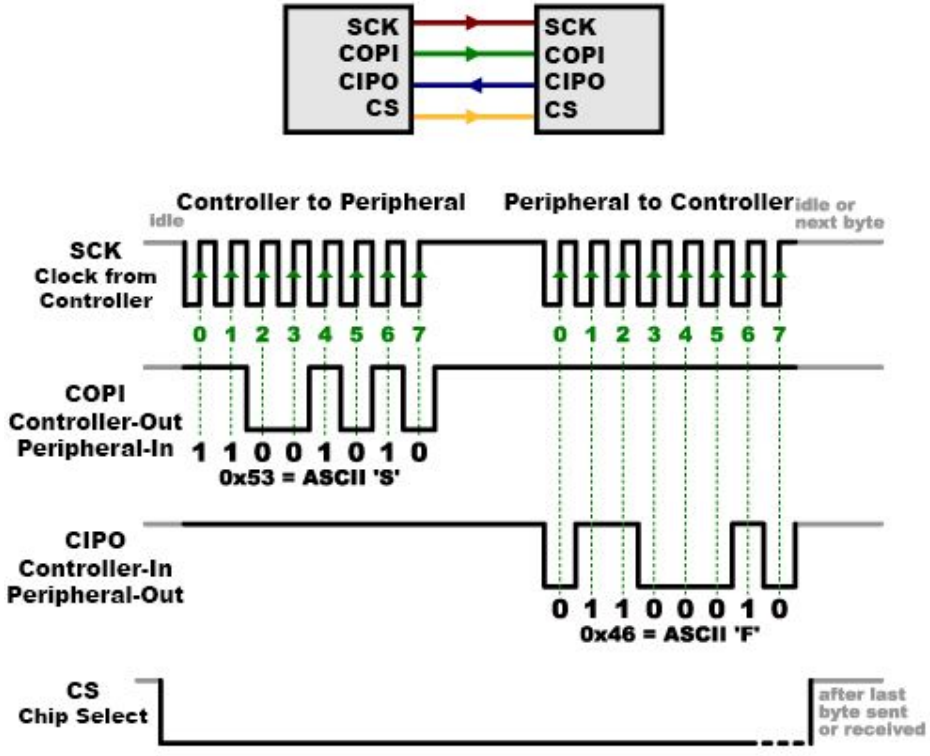
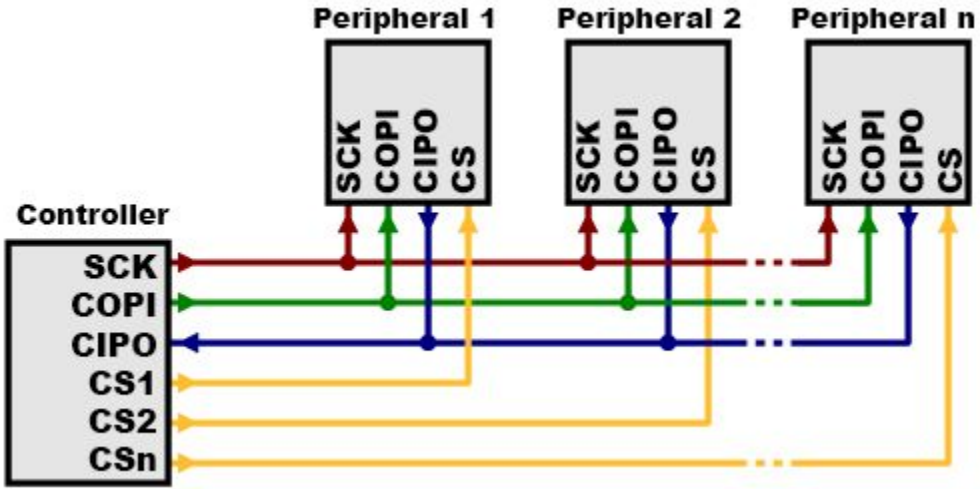


Image: Sparkfun

Using a chip select line (CS) we can talk to many identical devices - no need to worry about addresses!



LEEMAN GEOPHYSICAL LLC
CONSULTING & INSTRUMENTATION

Image: SparkFun

SPI (Serial Peripheral Interface)

- Fast!
- Simple to implement in hardware
- Multiple peripherals
- No address conflicts

- Lots of wires!
- Well defined packets required
- All communications must go through the controller

Before starting wiring, it is important to:

- List all of the system inputs and outputs
- Draw up the schematic and/or block diagram
- Carefully check the specifications of each component
- Think through the power system
- Think about any noise considerations

Common Gotchas

- Incorrect logic levels
- Incorrect power
- Missing pull-up/downs
- Too long of wires
- No power
- Missing Grounds
- Things plugged into Arduino pins 0/1 (computer serial)