HACK DUKE

CODE FOR GOOD

Intro to hardware hacking

August Ning

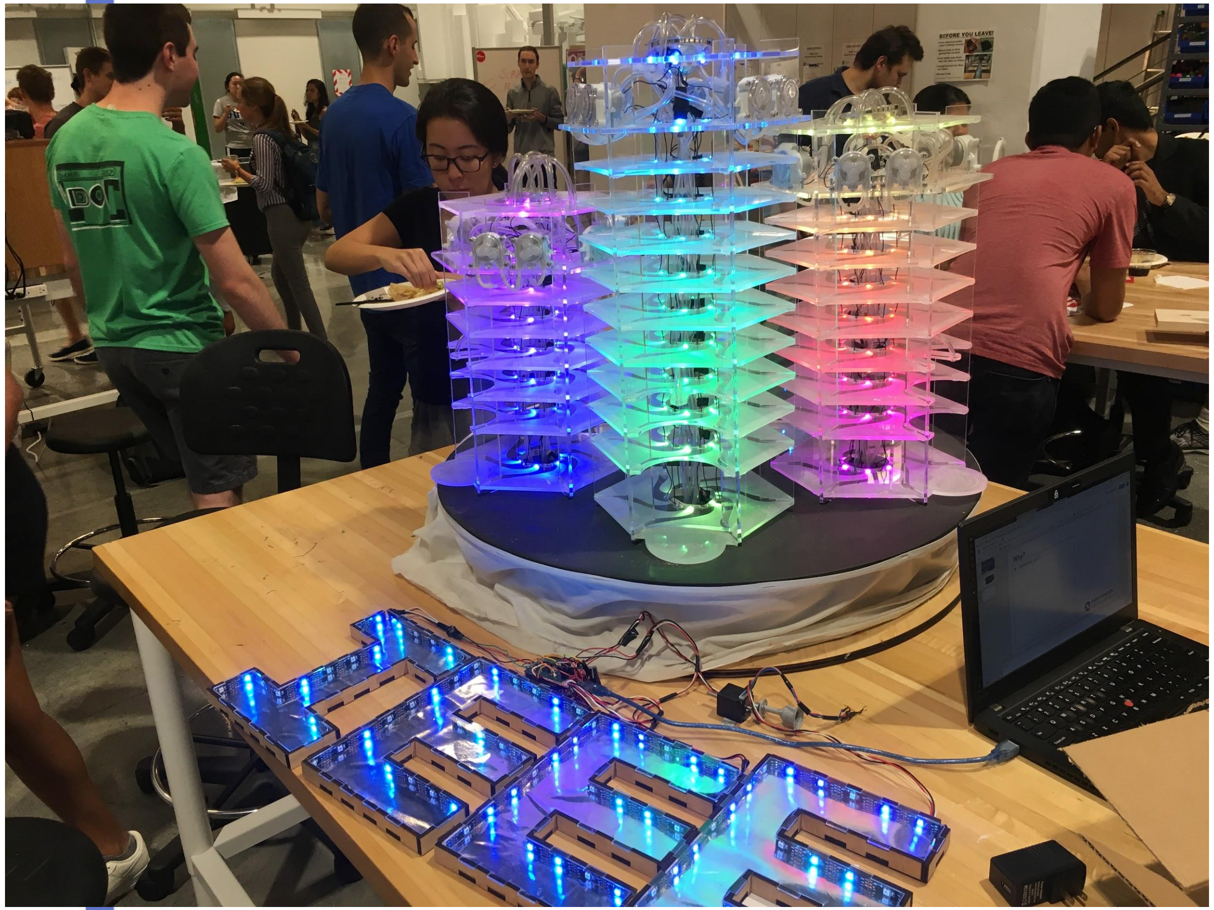
Follow Along!

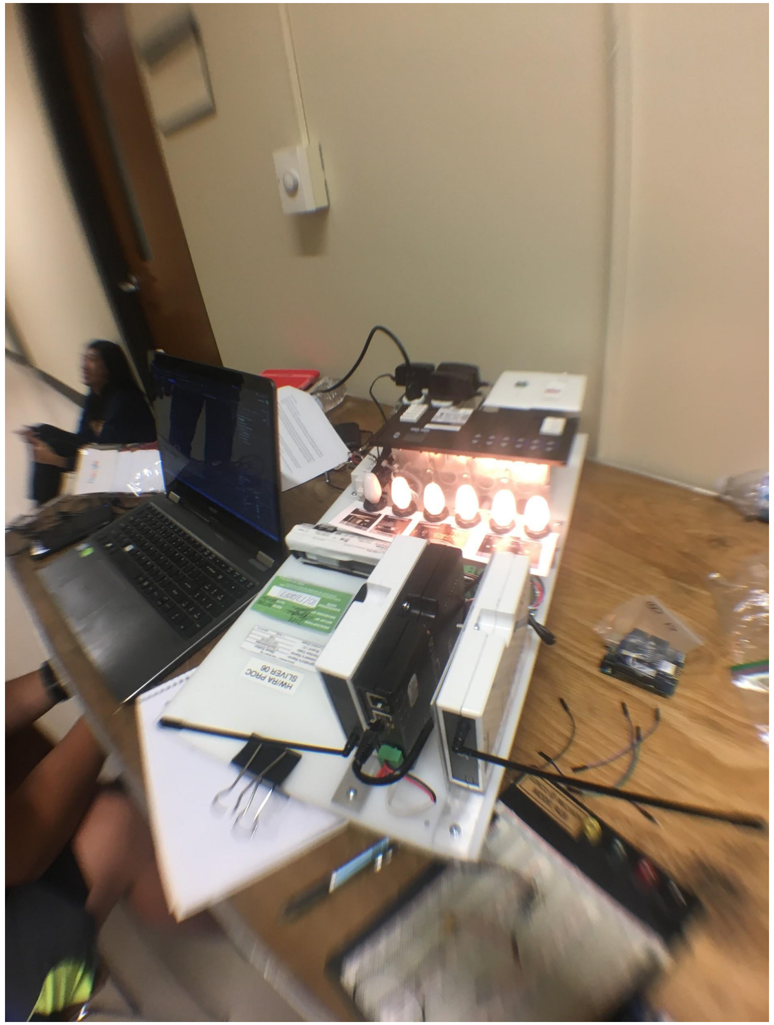
- Slides
- https://bit.ly/2pPaR3a
- Code
- https://bit.ly/2ITl1sm

Why hardware?

- I am a big fan of hardware
- I am here to hope to convince you to use some cool hardware in your hacks this weekend
- MLH has some hardware that you can consider using
- Hardware helps you stand out, and it’s not any harder than writing code

Why Hardware?





MLH Hardware Lab

- Amazon Fire Phone
- **Arduino 101 (Intel) and Base Shield**
- **Particle Photon**
- **Qualcomm DragonBoard 410c**
- **Leap Motion Controller**
- **Muse Headband**
- **Myo Armband**
- **FitBit Iconic**
- Misc Sensors, Soldering kit, Robotics Kit

MLH Hardware Lab

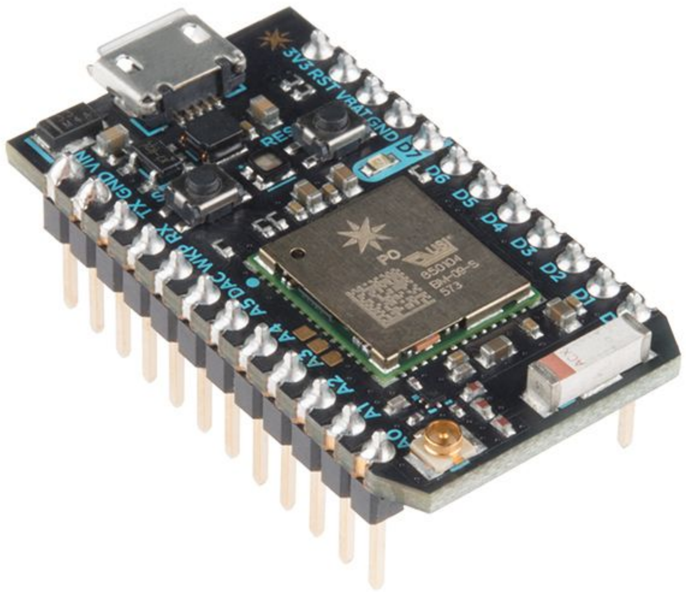
- Amazon Echo
- Google Home Mini
- Samsung Gear VR
- Oculus Rift
- Alienware Laptops

Arduino 101

- Based off Arduino, what the tutorial is going to be about!
- Has compatible shield and sensors that are easy to use

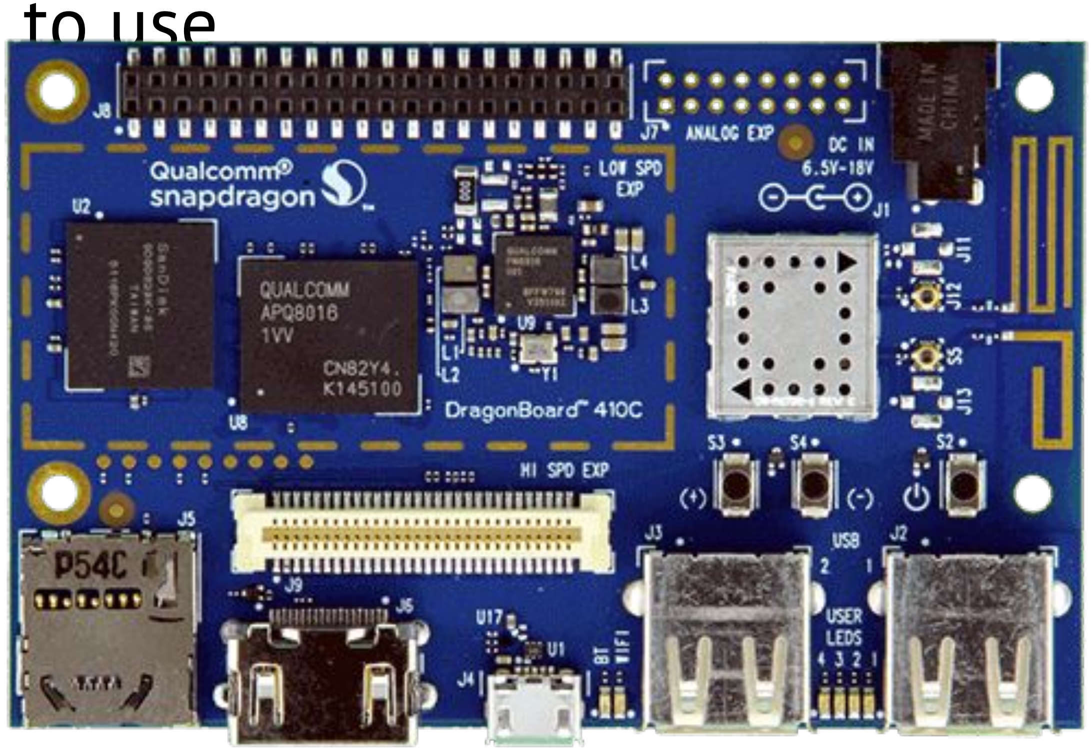
Particle Photon

- Arduino, but with wifi and IoT - uses same coding
- Integrated with Particle app
- Annoying to set up with university wifi



DragonBoard 410c

- Overpowered Raspberry Pi, has Bluetooth, Wifi, GPS
- Actually really hard to use



Leap Motion Controller

- Can track hand motion, and already has a decent SDK

Muse Headband

- Has EEG, accelerometer, gyroscope
- Can request SDK for all platforms

Myo Armband

- Can detect gestures and motions, mapped to computer functions - JS and C++ SDK
- Literally got discontinued yesterday lmao (10/12)

Fitbit Ionic

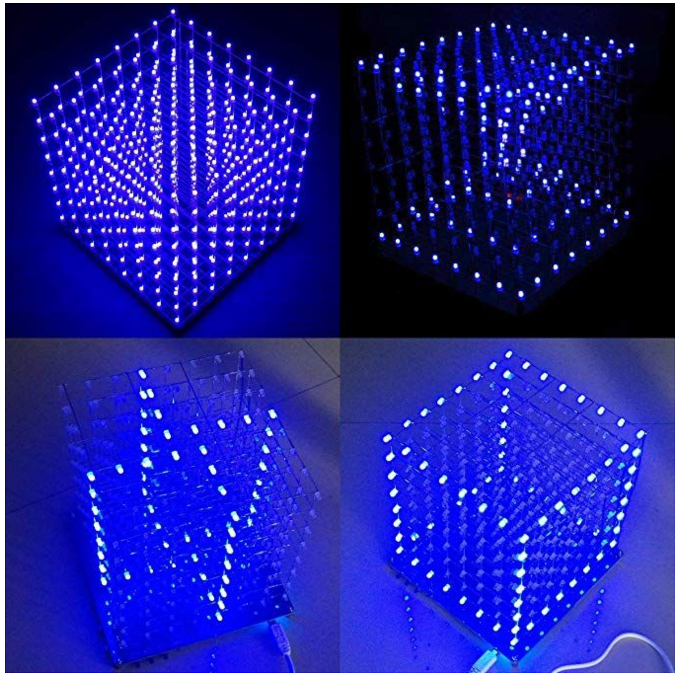
- Health tracker with JS SDK and lots of APIs

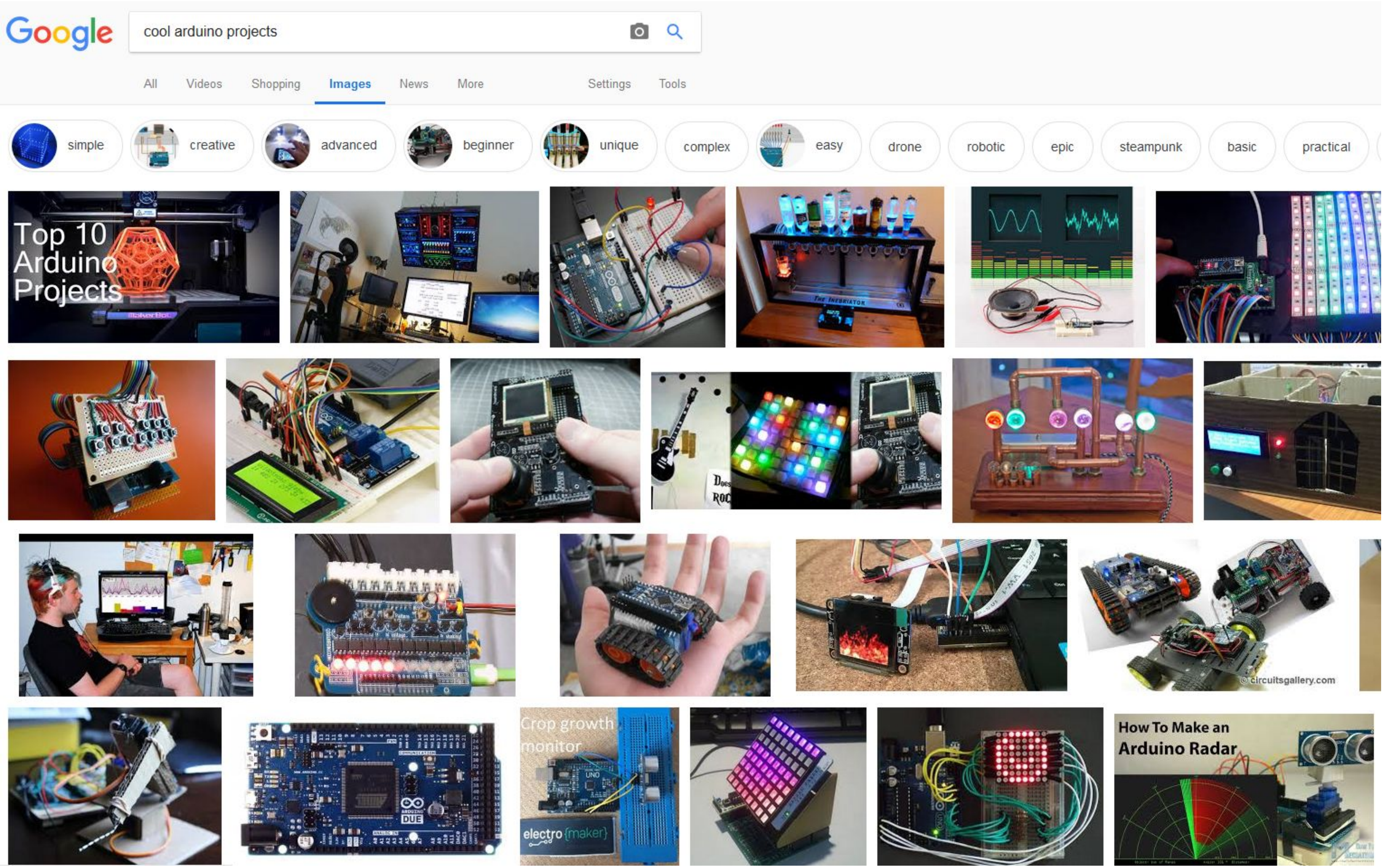
What?

- An Arduino is a open source microcontroller board!
  Easy to program
- Many sensors and are made for them

Why?

- Build cool stuff!

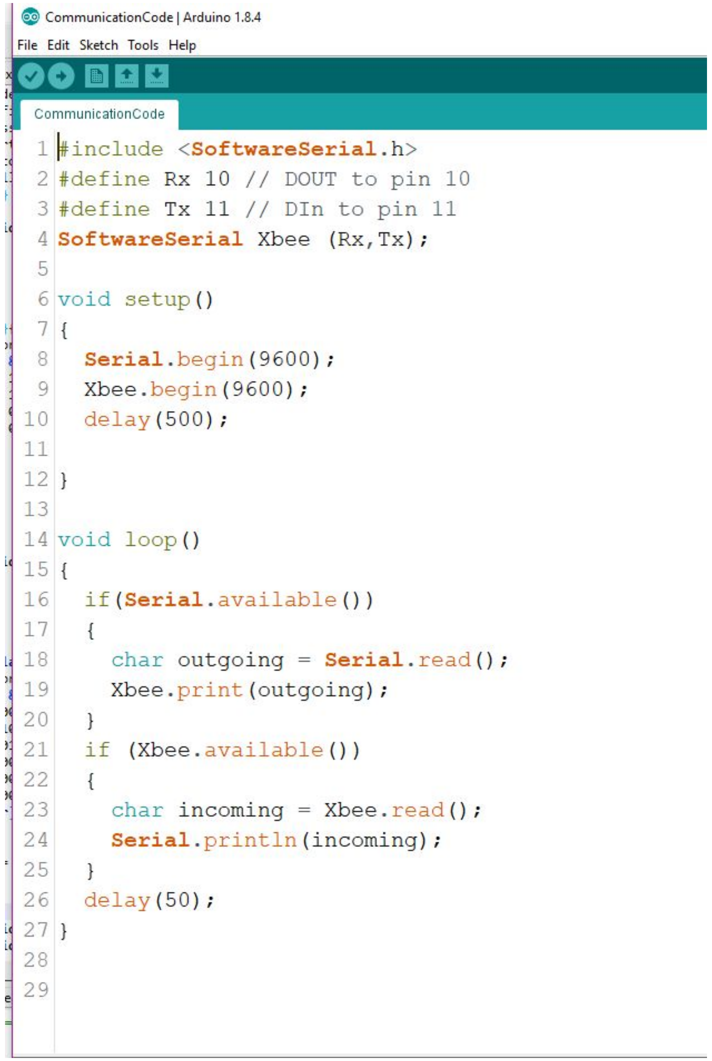




cool arduino projects

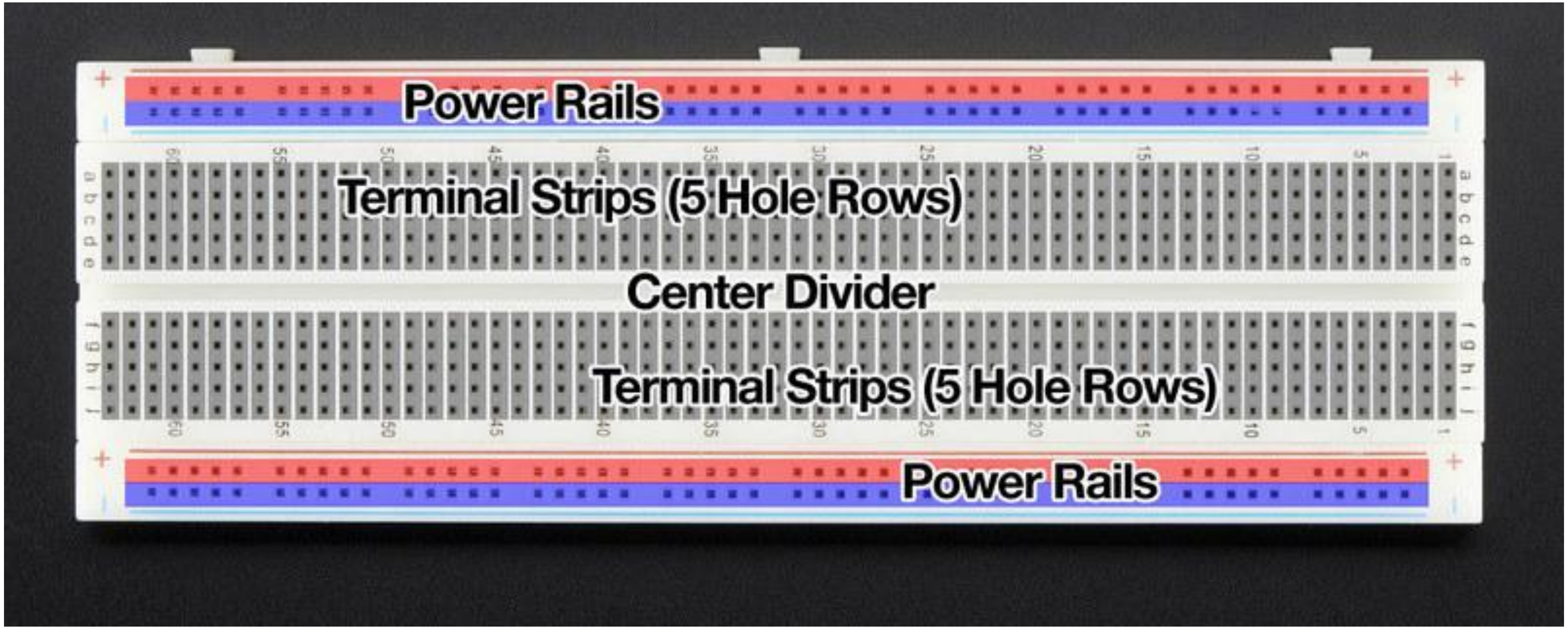
Arduino Programming Language

- Based on C/C++
- Can import libraries to do stuff for you
- Most of the code you need is already on the internet


CommunicationCode | Arduino 1.8.4
File Edit Sketch Tools Help
CommunicationCode

```
#include <SoftwareSerial.h>
#define Rx 10 // DOUT to pin 10
#define Tx 11 // DIn to pin 11
SoftwareSerial Xbee (Rx,Tx);

void setup()
{
  Serial.begin(9600);
  Xbee.begin(9600);
  delay(500);

}

void loop()
{
  if(Serial.available())
  {
    char outgoing = Serial.read();
    Xbee.print(outgoing);
  }
  if (Xbee.available())
  {
    char incoming = Xbee.read();
    Serial.println(incoming);
  }
  delay(50);
}
```

Breadboard

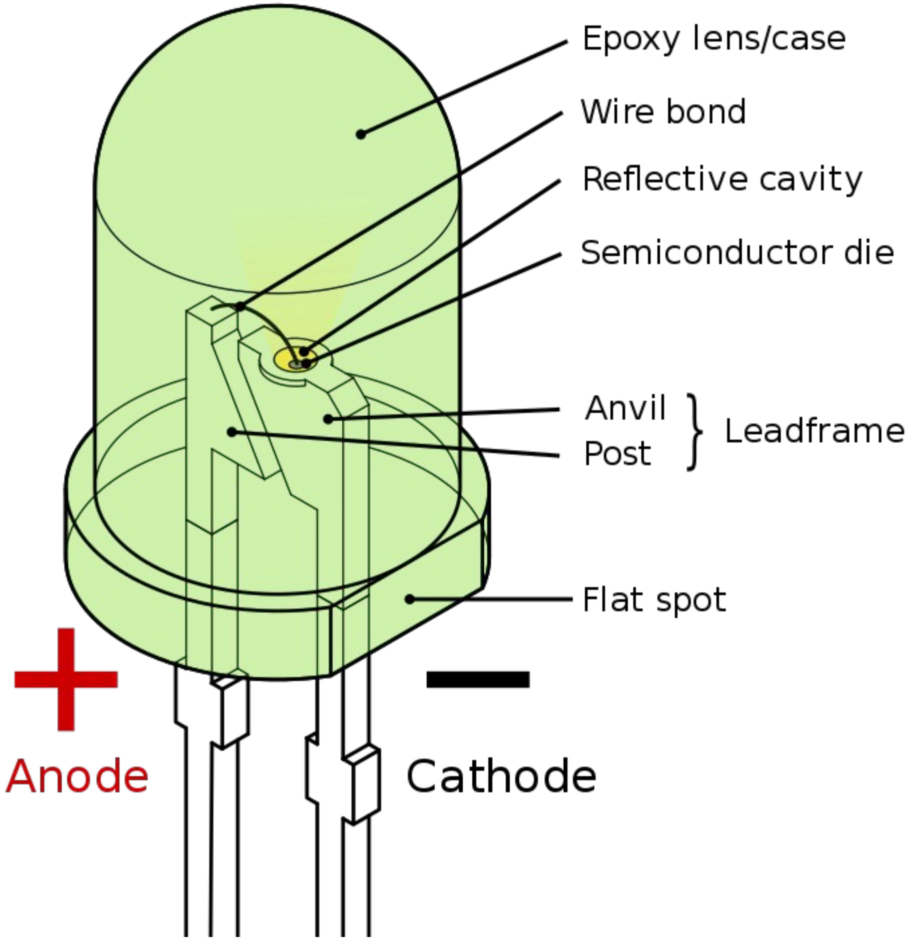

Power Rails
Terminal Strips (5 Hole Rows)
Center Divider
Terminal Strips (5 Hole Rows)
Power Rails

Digital

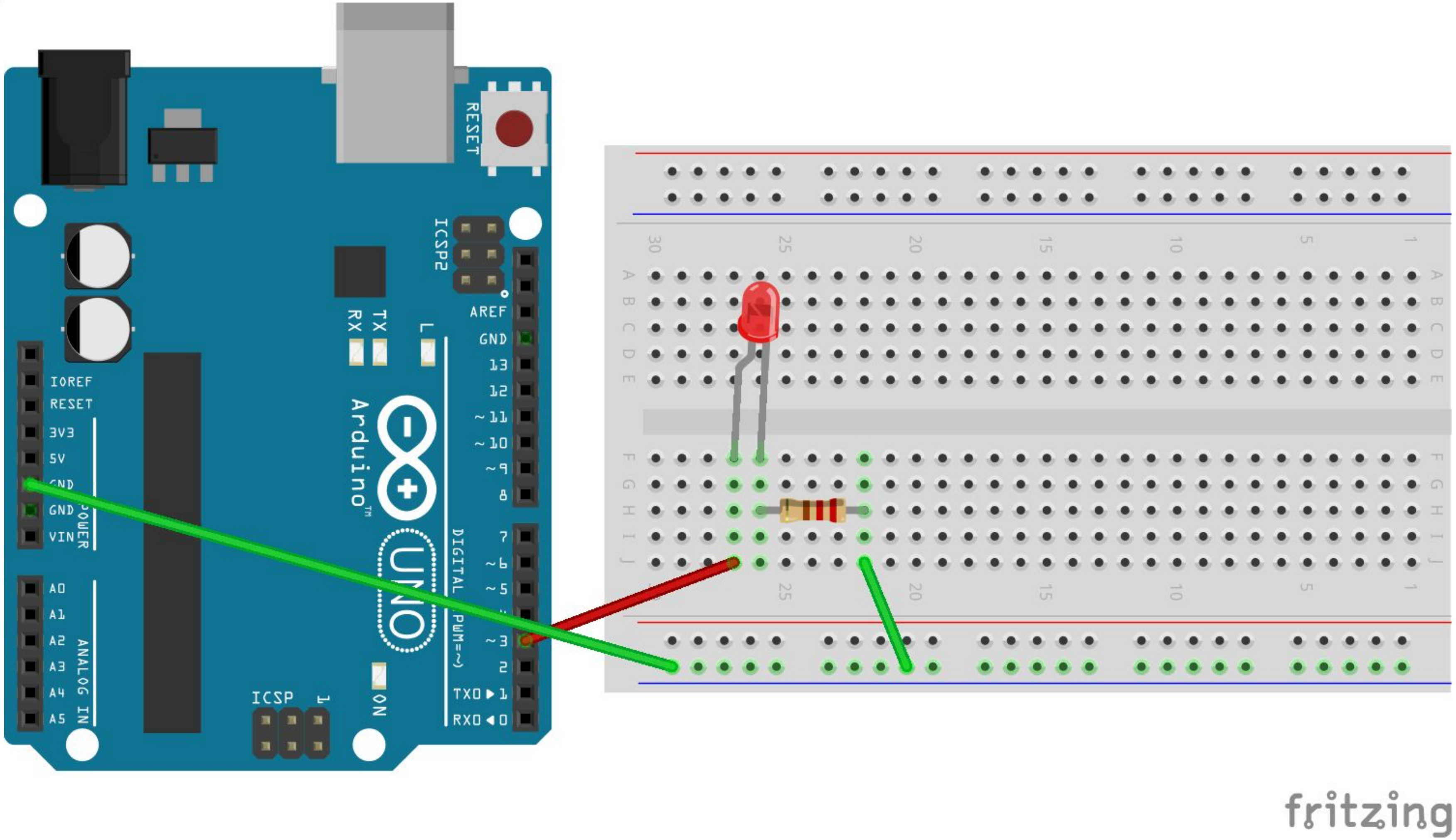
- 0’s and 1’s, HIGHs and LOWs
- Arduino high output is 5 V
  - Enough to power an LED
- Sorta like the “Hello World” of hardware

Blink and LED - Wiring

- Wire anode to any digital PWM pin
- Wire cathode to resistor
- Wire resistor to ground

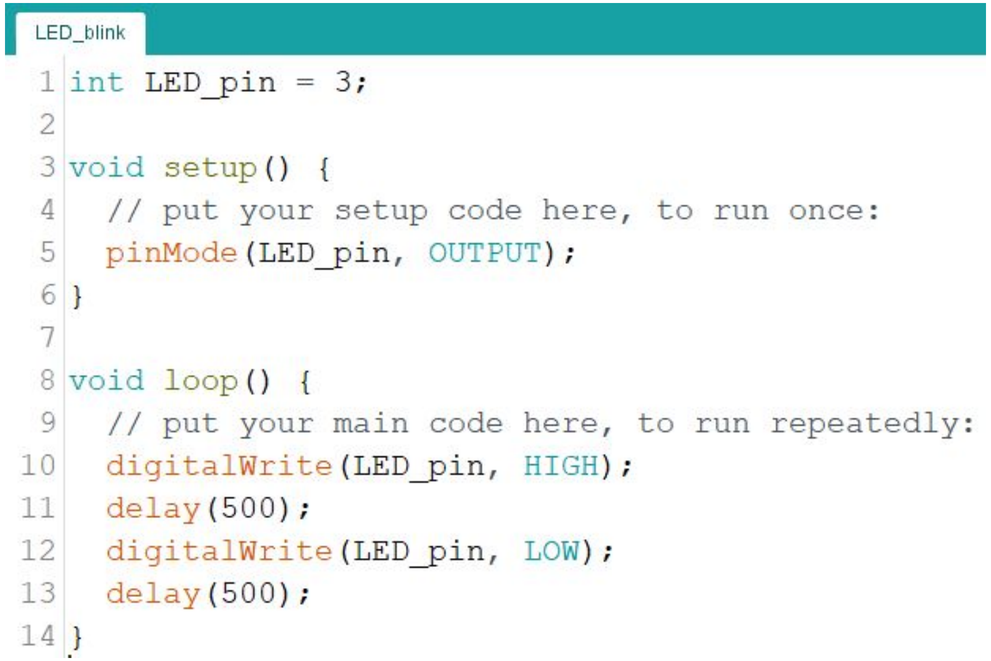


Blink and LED - Wiring



Blink and LED - Coding

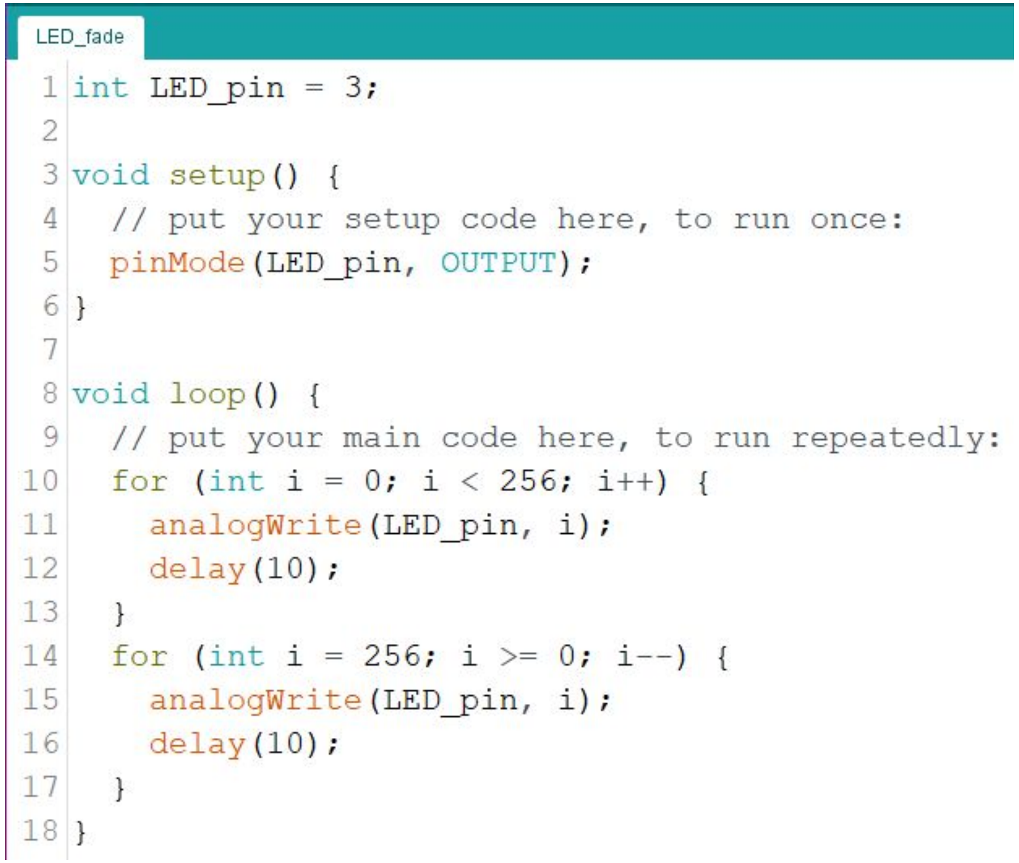
- In setup, declare the pin as an output pin
- In loop, digitalWrite the pin high and low with delays in between
- Compile and push to Arduino



```
int LED_pin = 3;

void setup() {
  // put your setup code here, to run once:
  pinMode(LED_pin, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LED_pin, HIGH);
  delay(500);
  digitalWrite(LED_pin, LOW);
  delay(500);
}
```

Fade an LED - Coding

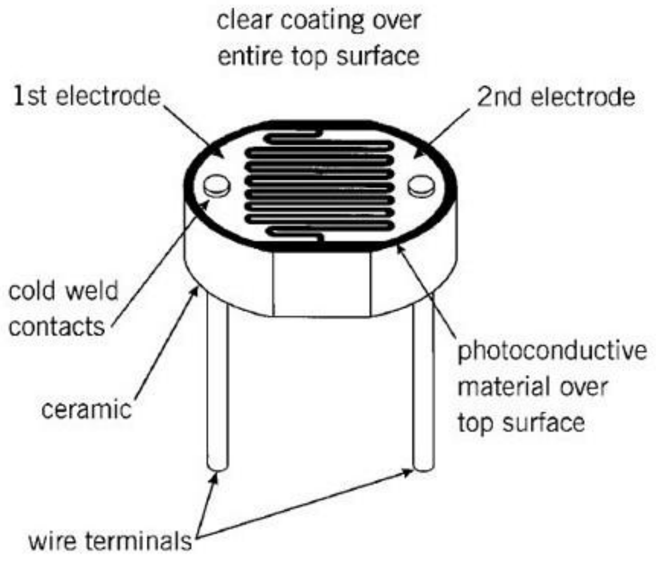
- Write a for loop from 0 to 255
- Increment each time by 1, and delay for a bit
- Reverse the for loop



```
int LED_pin = 3;

void setup() {
  // put your setup code here, to run once:
  pinMode(LED_pin, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  for (int i = 0; i < 256; i++) {
    analogWrite(LED_pin, i);
    delay(10);
  }
  for (int i = 256; i >= 0; i--) {
    analogWrite(LED_pin, i);
    delay(10);
  }
}
```

Analog

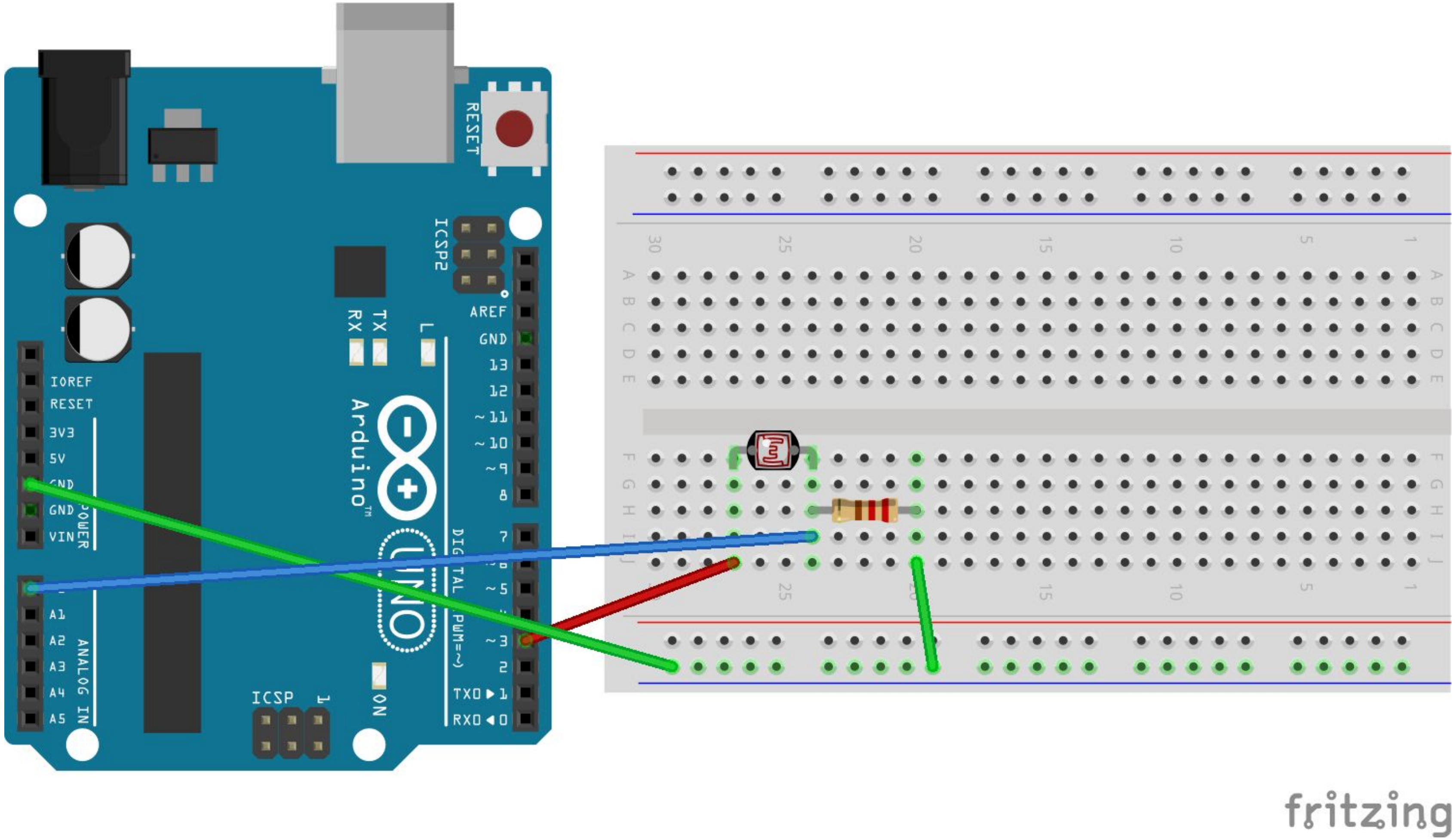
- Dynamic, can map 0 to 5 V input to 0 to 1023
- Useful for reading in voltage values
- Can use photoresistors to show variable voltage divider


clear coating over entire top surface
1st electrode
2nd electrode
cold weld contacts
ceramic
photoconductive material over top surface
wire terminals

Photoresistor Voltage Divider - Wiring

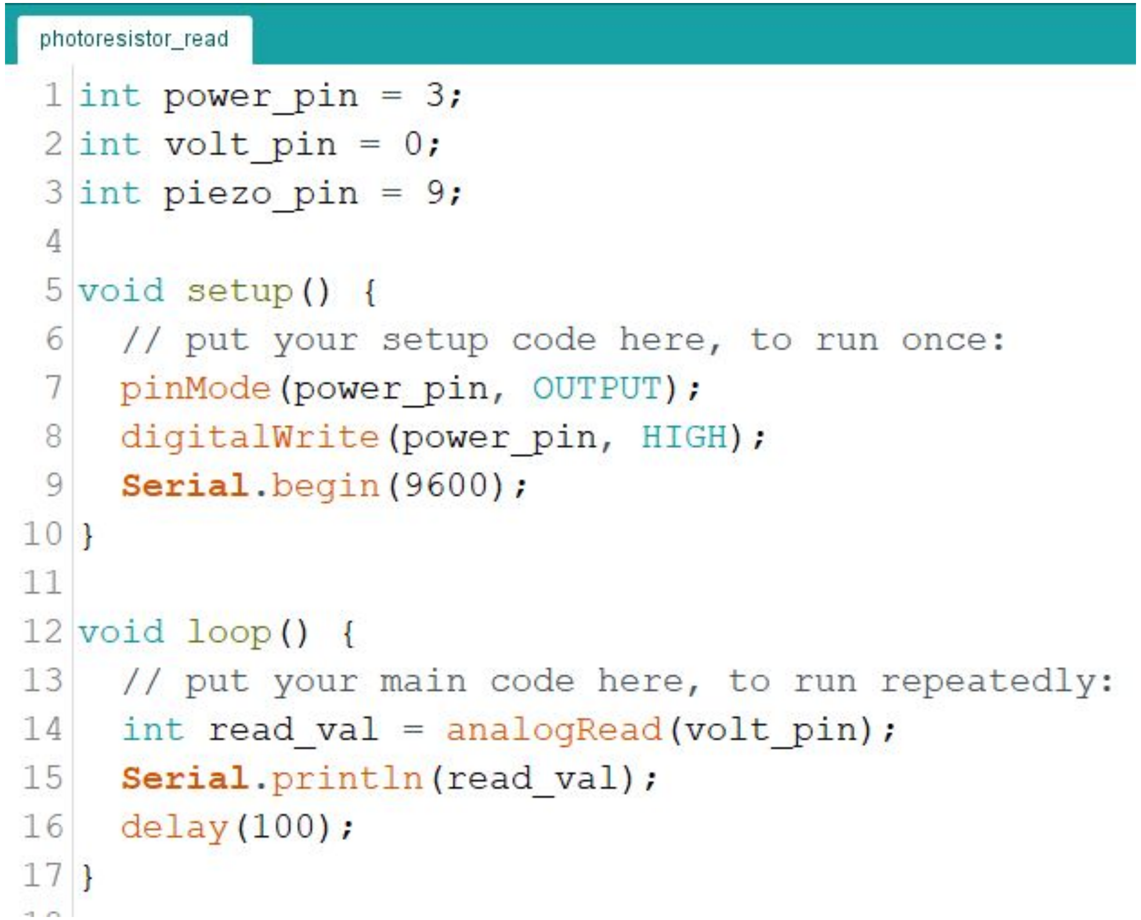
- Can put wire on to 5 V
- Put photoresistor in series with 220 Ohm resistor
- 220 Ohm resistor to ground
- Wire analog port to node between photoresistor and 220 Ohm resistor

Photoresistor Voltage Divider - Wiring



Photoresistor Voltage Divider - Coding

- Start serial monitor in setup
- Poll every few seconds on the analog port with analogRead
- Vary the light on the photoresistor

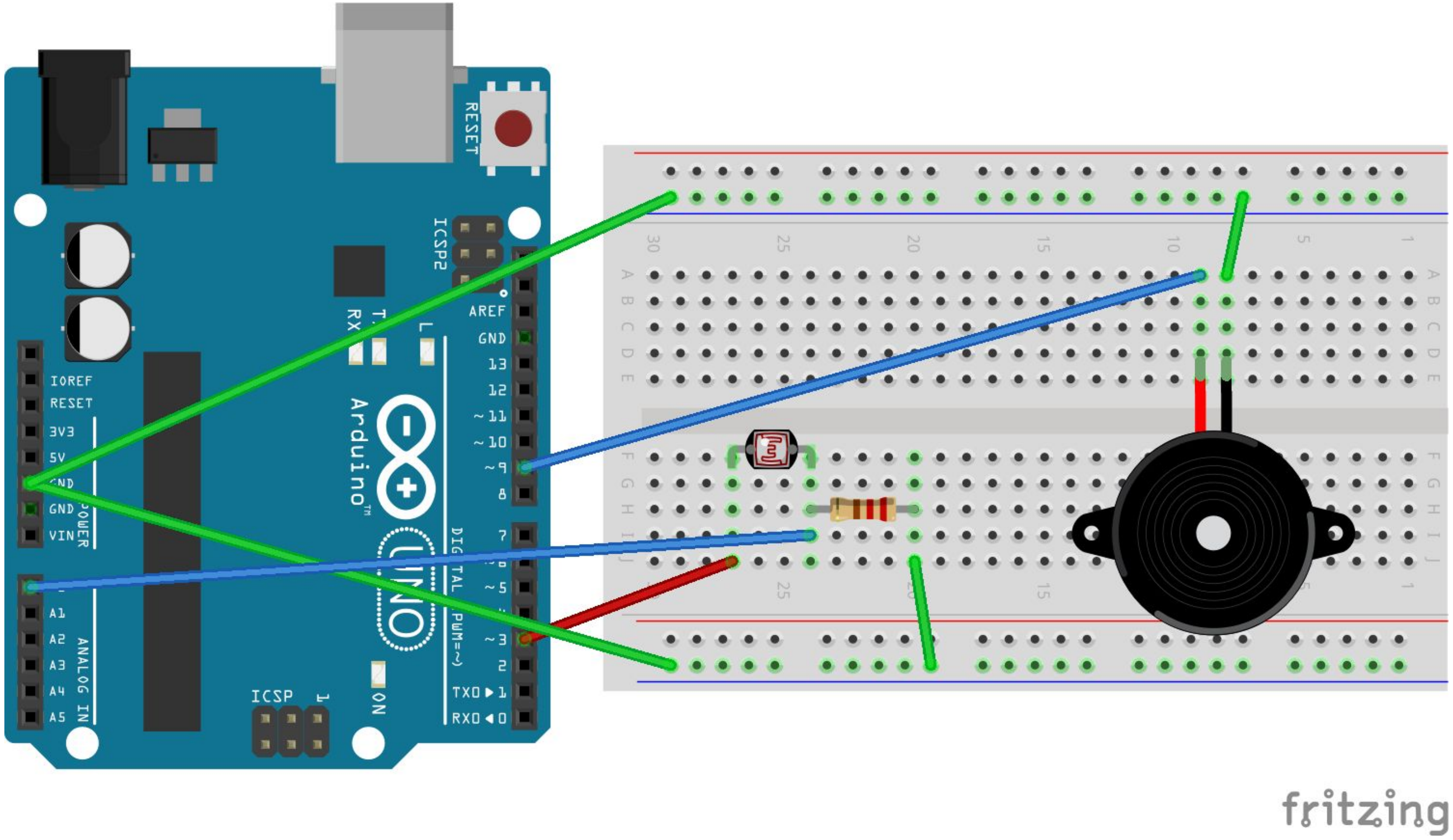


```
int power_pin = 3;
int volt_pin = 0;
int piezo_pin = 9;

void setup() {
  // put your setup code here, to run once:
  pinMode(power_pin, OUTPUT);
  digitalWrite(power_pin, HIGH);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  int read_val = analogRead(volt_pin);
  Serial.println(read_val);
  delay(100);
}
```

Jank Theremin - Wiring

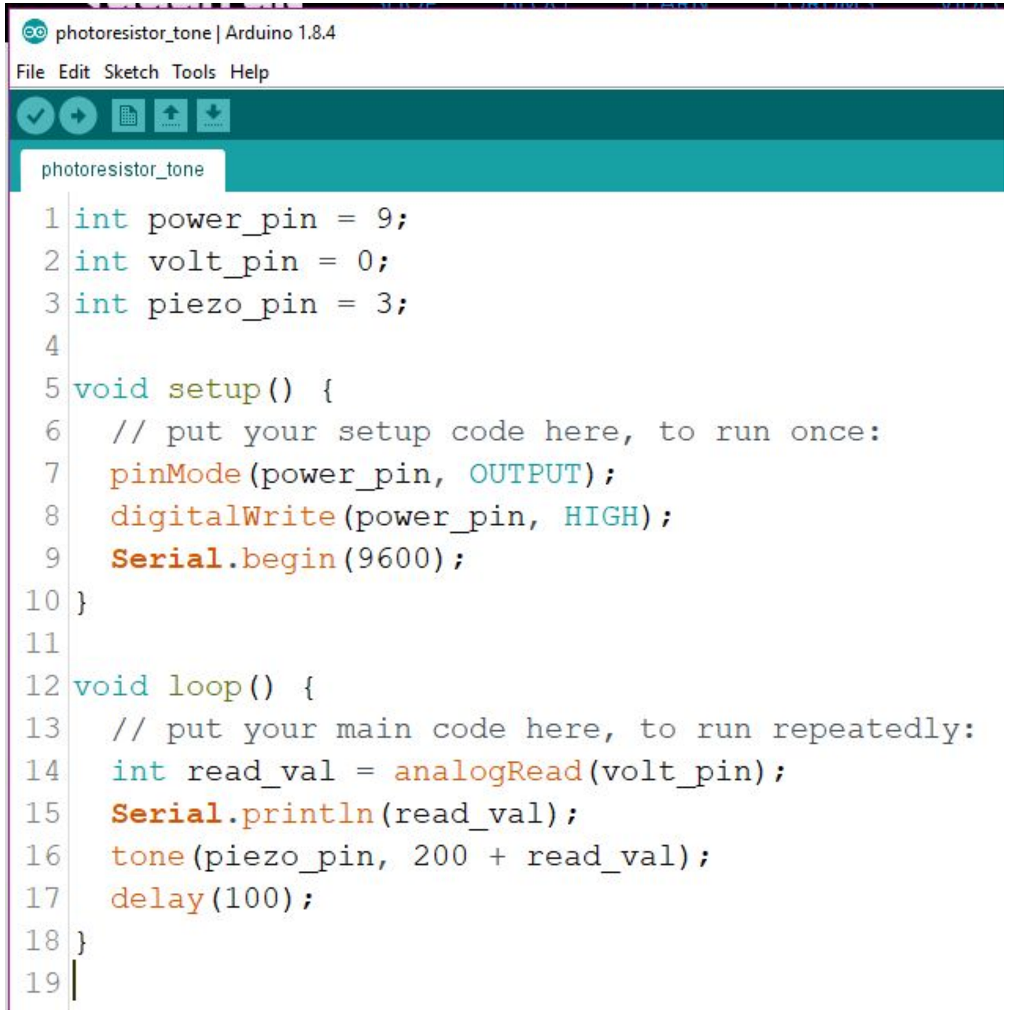
- Wire up a piezobuzzer + up to a PWM digital pin
- Wire other side to -

Jank Theremin - Wiring



Jank Theremin - Coding

- Adding to your photoresistor code, save the analogRead to a int
- Call the tone command on the piezobuzzer pin with the value of the analogRead


photoresistor_tone | Arduino 1.8.4
File Edit Sketch Tools Help
photoresistor_tone

```
int power_pin = 9;
int volt_pin = 0;
int piezo_pin = 3;

void setup() {
  // put your setup code here, to run once:
  pinMode(power_pin, OUTPUT);
  digitalWrite(power_pin, HIGH);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  int read_val = analogRead(volt_pin);
  Serial.println(read_val);
  tone(piezo_pin, 200 + read_val);
  delay(100);
}

```

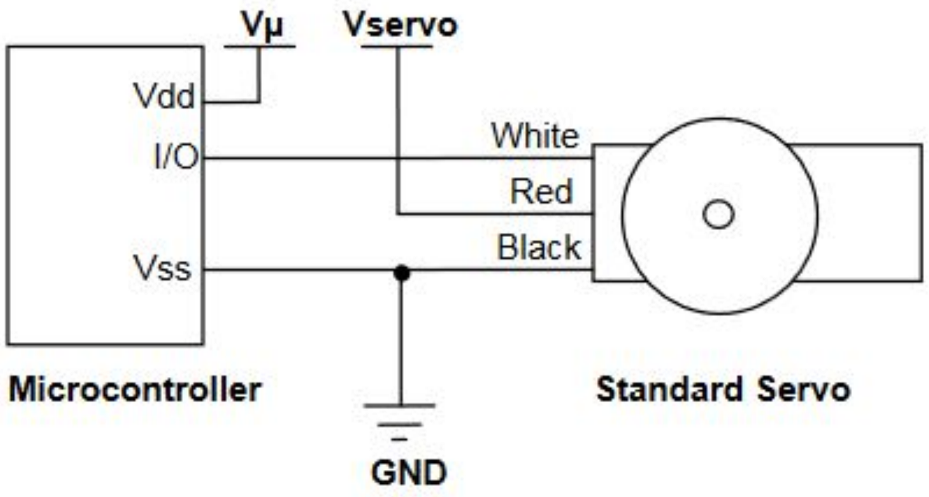
Libraries

- Provide extra functionality for use in sketches, e.g. working with hardware or manipulating data
- Servo is standard Arduino Library
- Lots of sensors that you use have pre written libraries

Servo - Wiring

- Put white wire on PWM digital port

Quick-Start Circuit



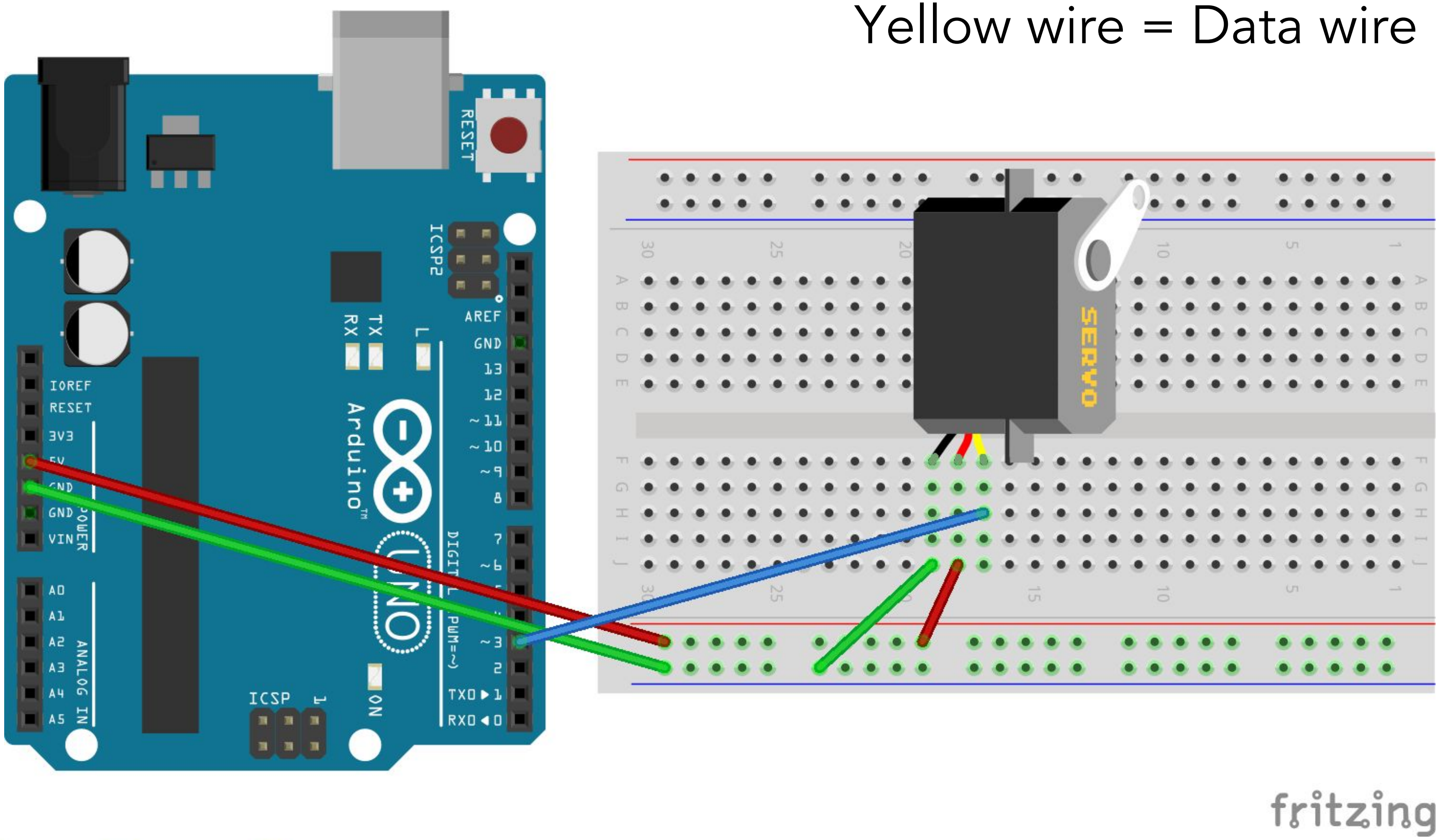
Vµ = microcontroller voltage supply

Vservo = 4 to 6 VDC, regulated or battery

I/O = PWM TTL or CMOS output signal from microcontroller: 3.3 to 5 V, not to exceed Vservo + 0.2 V

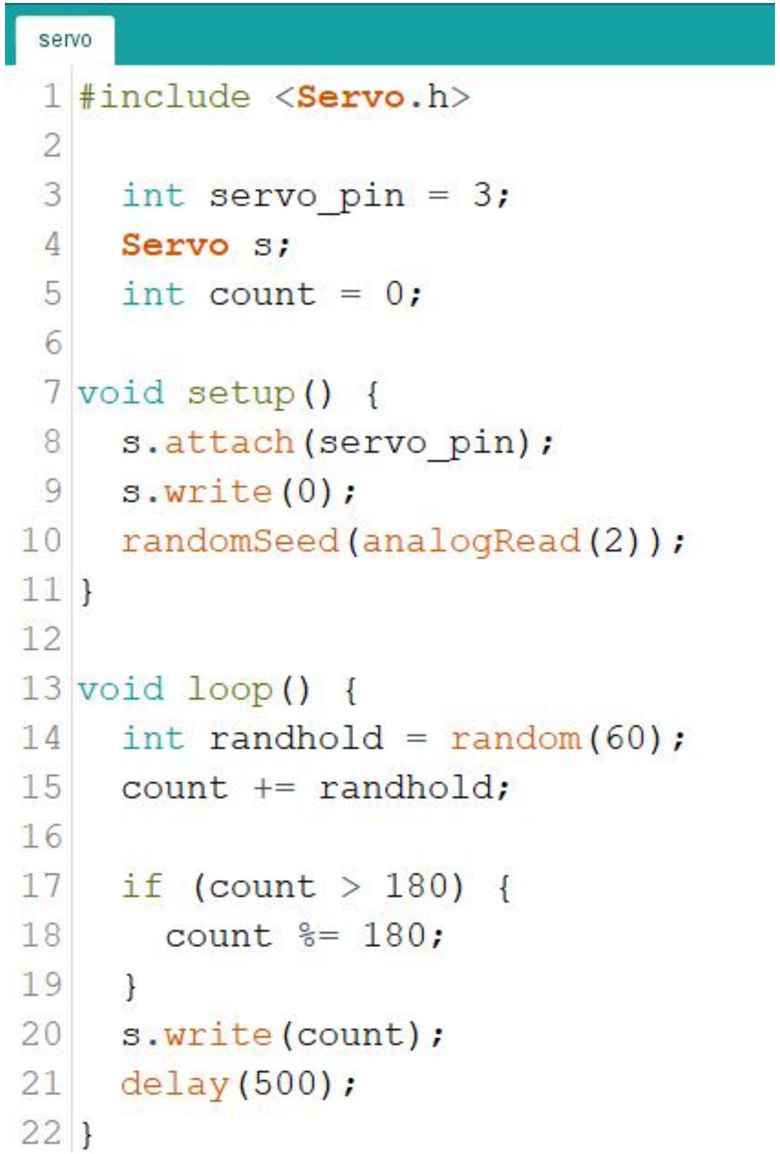
Servo - Wiring

Yellow wire = Data wire



Servo - Coding

- Remember to include the Servo.h
- Servo can set itself to any angle between 0 and 180
- This set the servo at some random position every 500 ms



```
#include <Servo.h>

  int servo_pin = 3;
  Servo s;
  int count = 0;

void setup() {
  s.attach(servo_pin);
  s.write(0);
  randomSeed(analogRead(2));
}

void loop() {
  int randhold = random(60);
  count += randhold;

  if (count > 180) {
    count %= 180;
  }
  s.write(count);
  delay(500);
}
```

What’s Next?

- Combine all the code you’ve done together for fun
- Make functions to help with repetitive code
- You now know enough Arduino to do any project you want to do
- You can probably google most questions that you have
- Make cool projects!