Making with Arduino workshop

Class Materials:

https://github.com/bolandrm/arduino_workshop

I’m Ryan Boland

Web Developer @ Tanooki Labs

@bolandrm (github, twitter)

Homework - Install Arduino IDE

Part 1 - Getting started with Arduino & Electronics

Intermission - Build your first circuit and upload a sketch!

Part 2 - More on C and embedded programming (Lots of demos)

Part 3 - Build something!

Safety

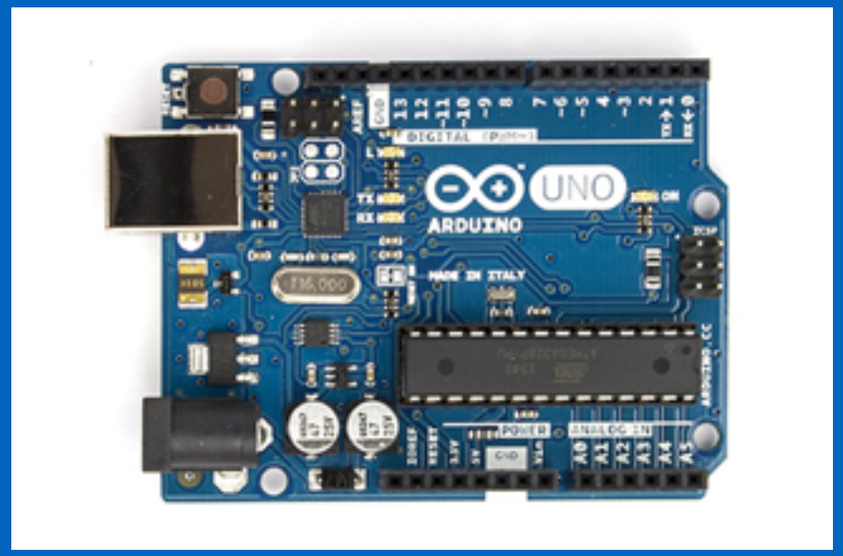
- **Avoid high voltage/current power sources. Household electricity, for example, is extremely dangerous for hobbyists to work with.**
- **Power down circuits before modifying them.**
- **Power down circuits / disconnect batteries when you leave the room.**
- **Take a moment to double check your work before powering it on. Be careful of short-circuits.**

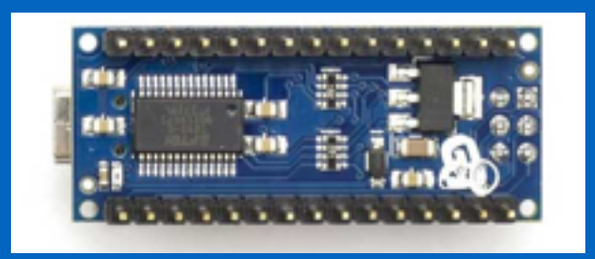
What is Arduino?

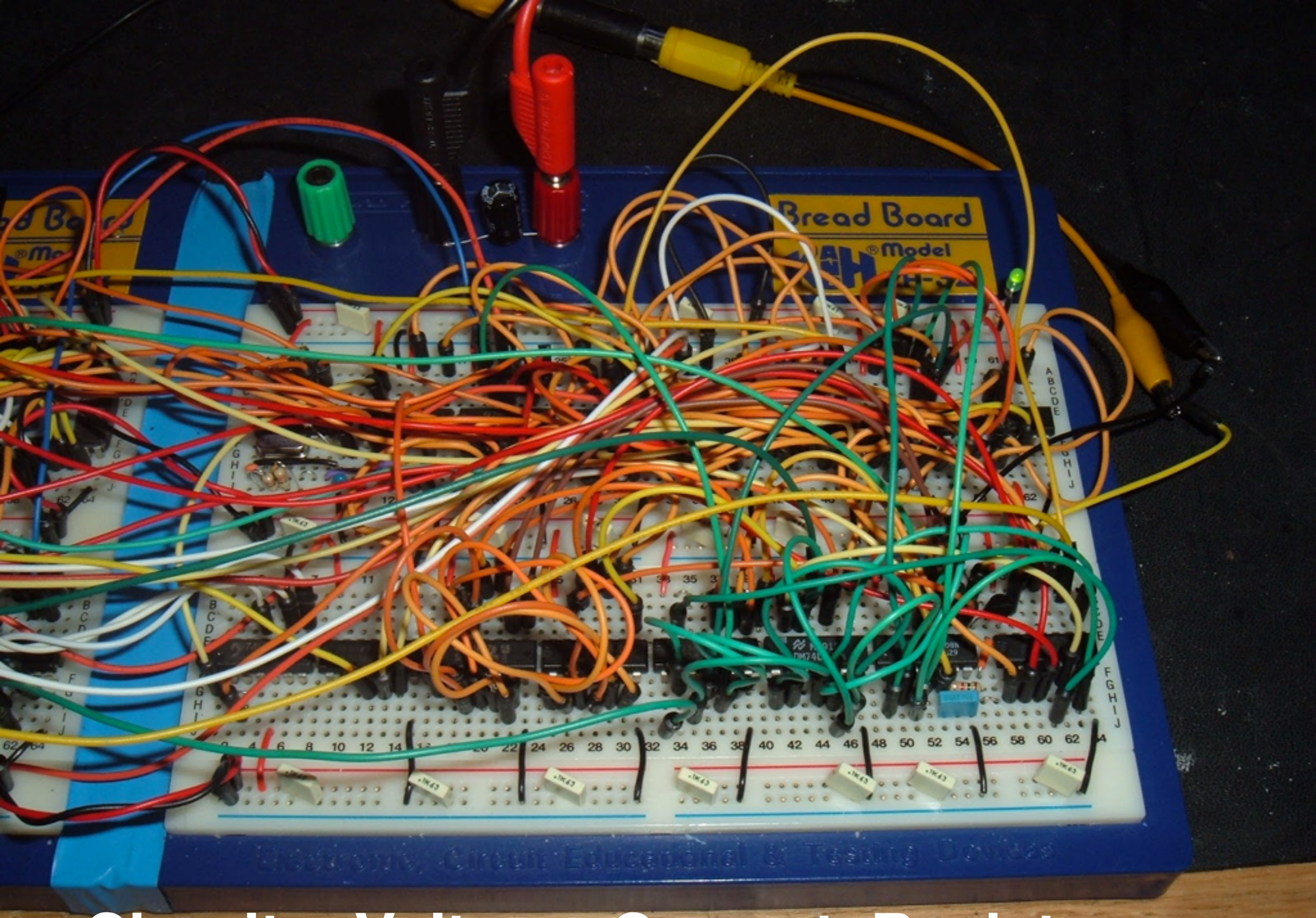
“Arduino is an open-source prototyping platform based on easy-to-use hardware and software.”

- arduino.cc

Hardware and software.



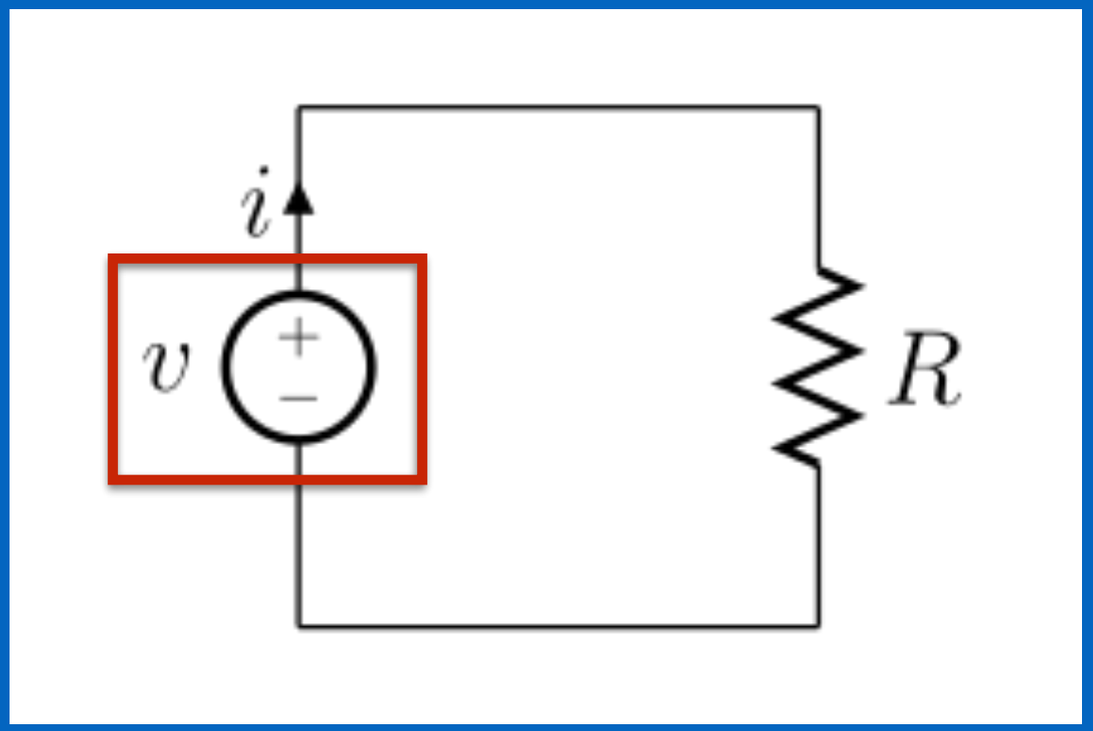




Circuits: Voltage, Current, Resistance

Voltage

Difference in electric potential energy between two points.



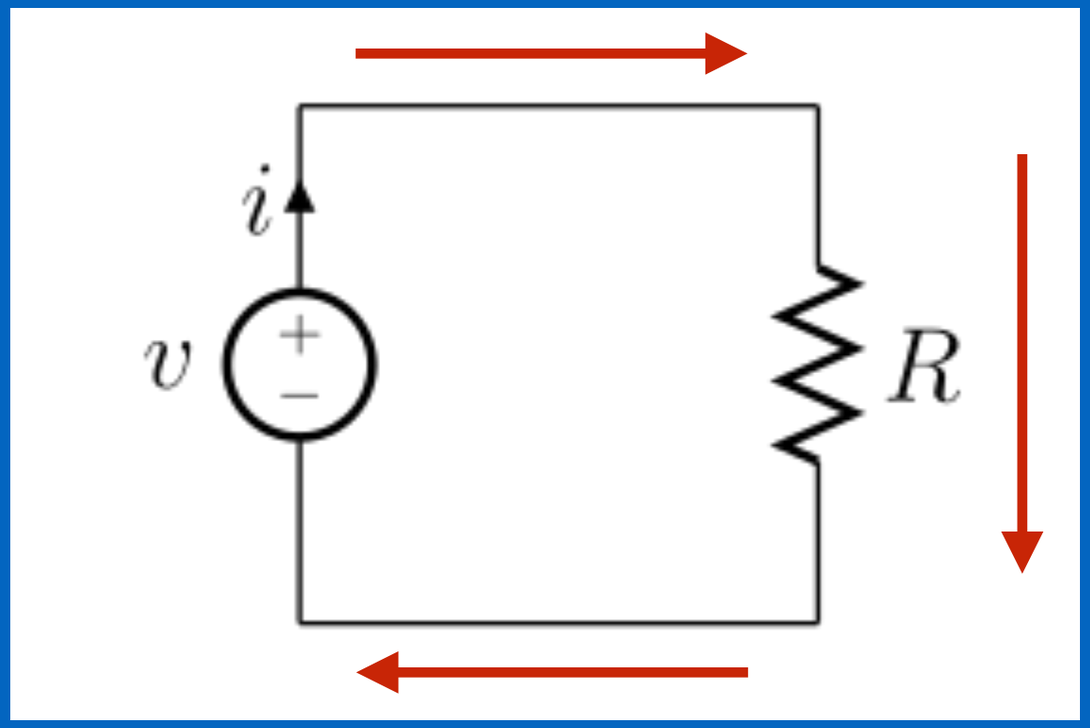
Measured in volts (V)


MADE
IN ITALY
AREF
GND
13
12
~11
~10
~9
8
7
6
5
4
3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
RESET-EN
RESET
ICSP
WWW.ARDUINO.CC
POWER
RESET
3.3V
5V
GND
GND
Vin
ANALOG IN
A0
A1
A2
A3
A4
A5

- 0 volts
- Negative
- Ground (GND)
- Common (COM)
- Reference (REF)
- Low

Current

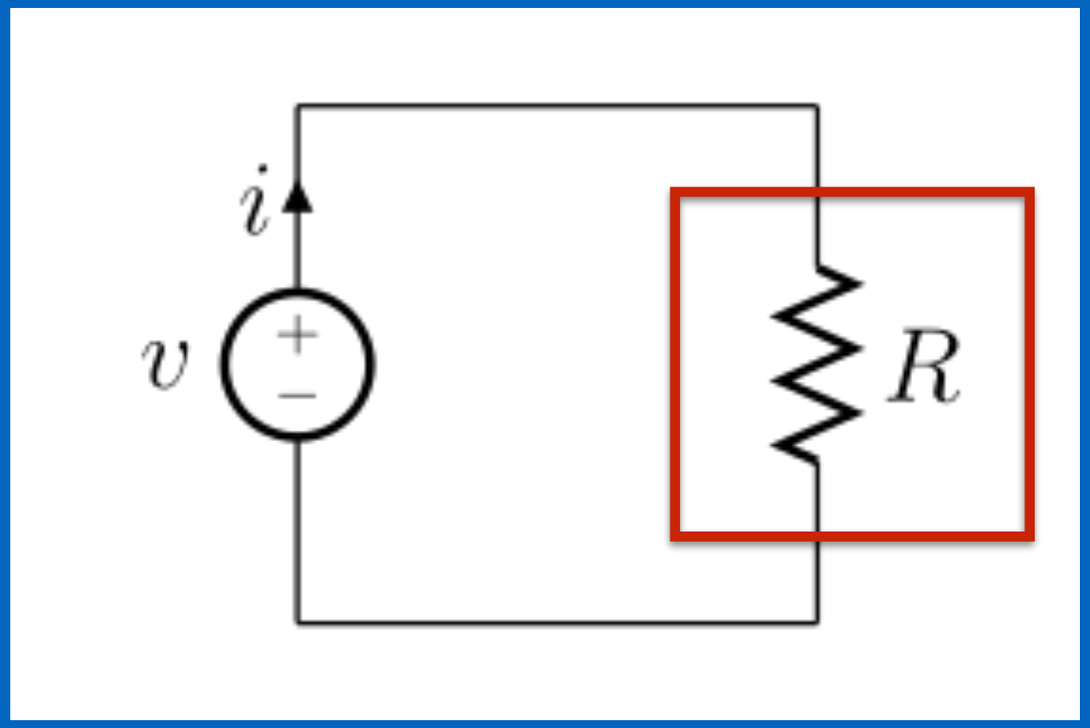
Flow of electric charge.

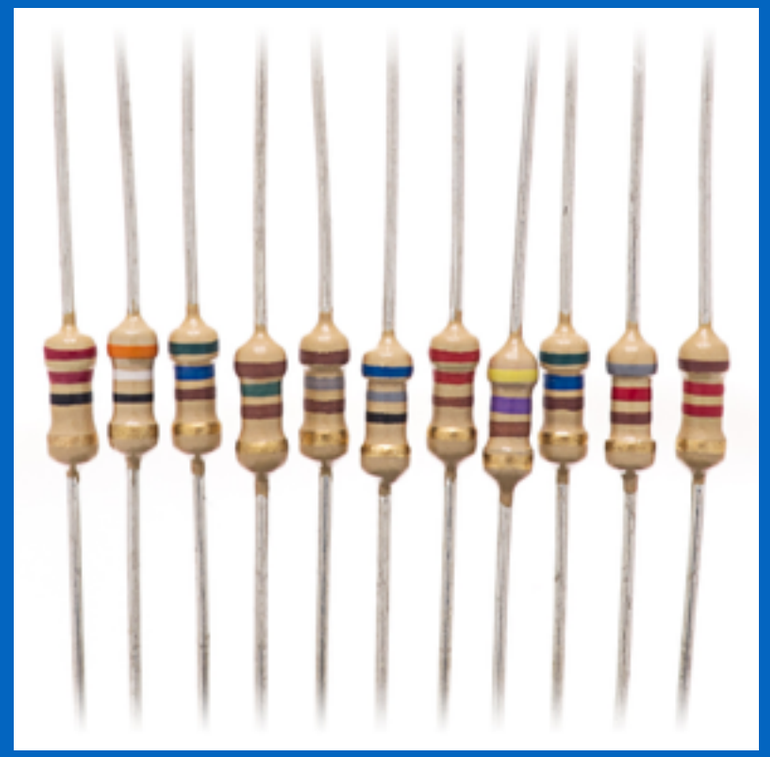


Measured in amperes, or amps (I)

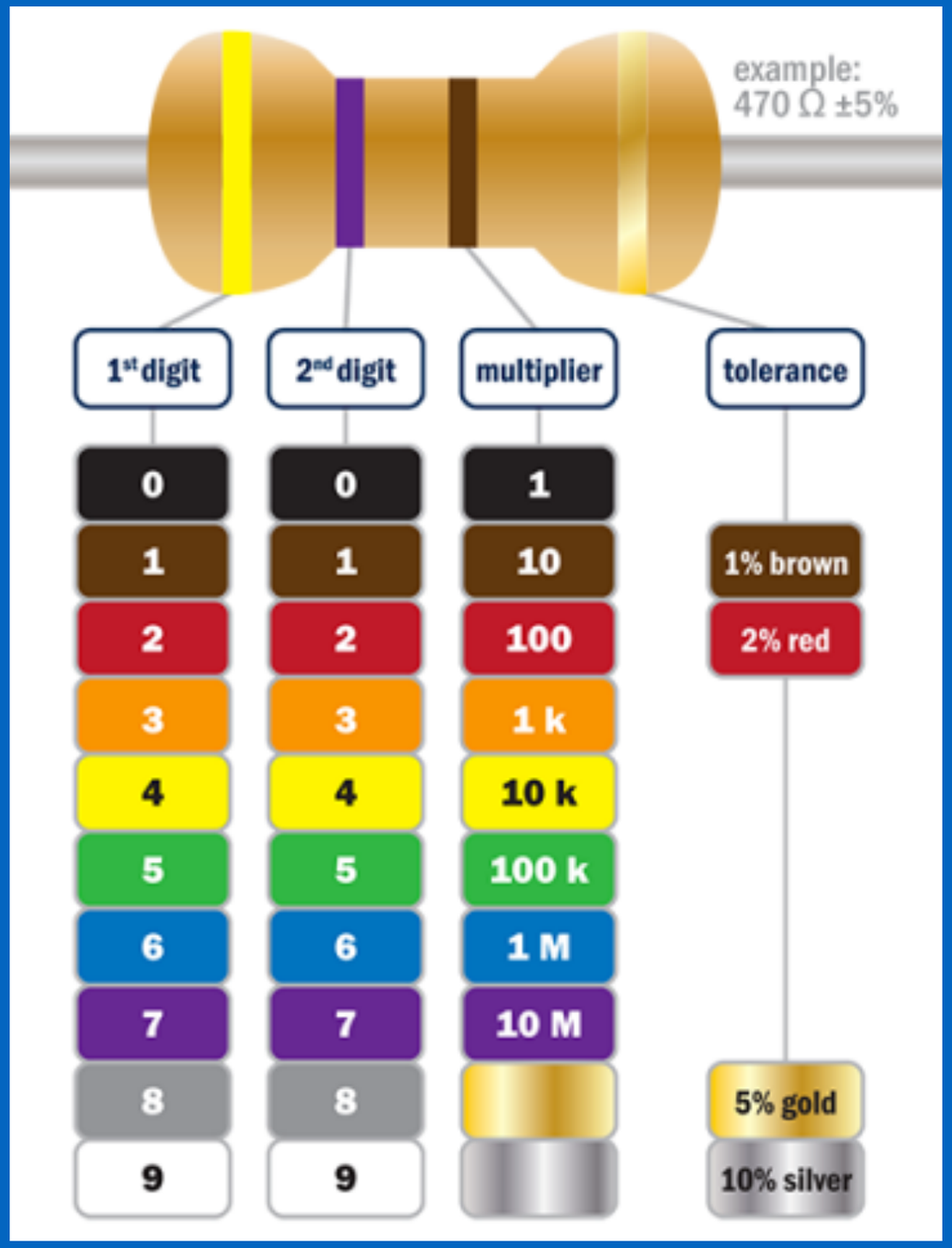
Resistance

Opposition to the passage of electric current





Measured in ohms (Ω)



Resistor Color Codes

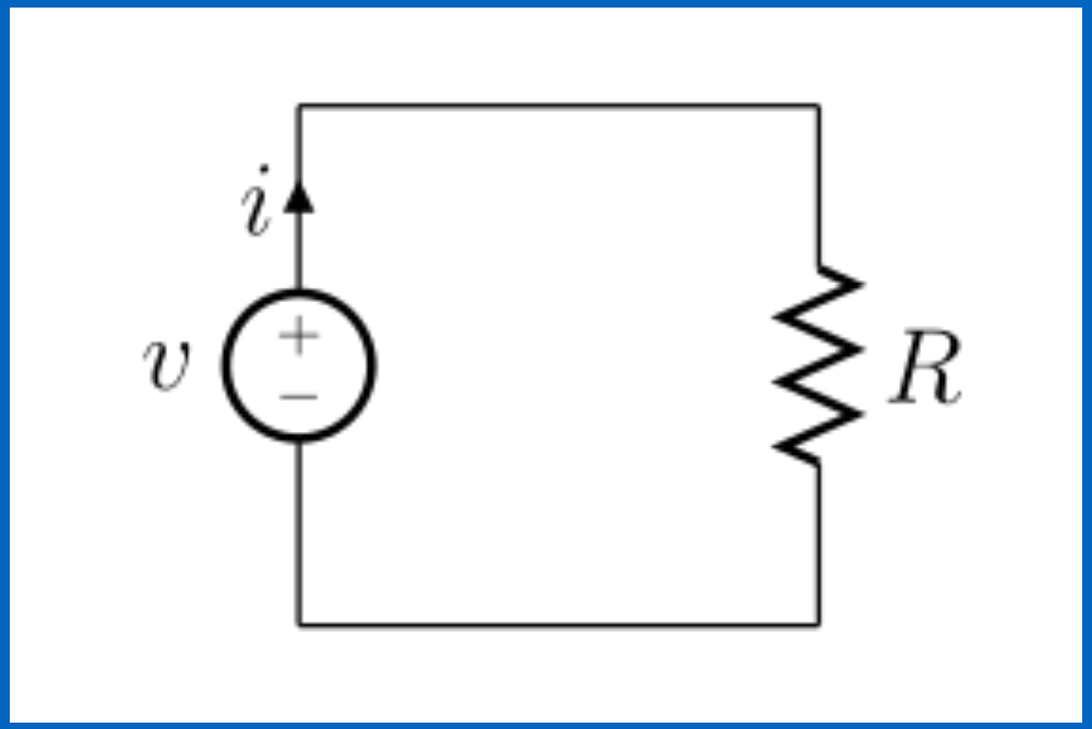
http://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band

All Together!

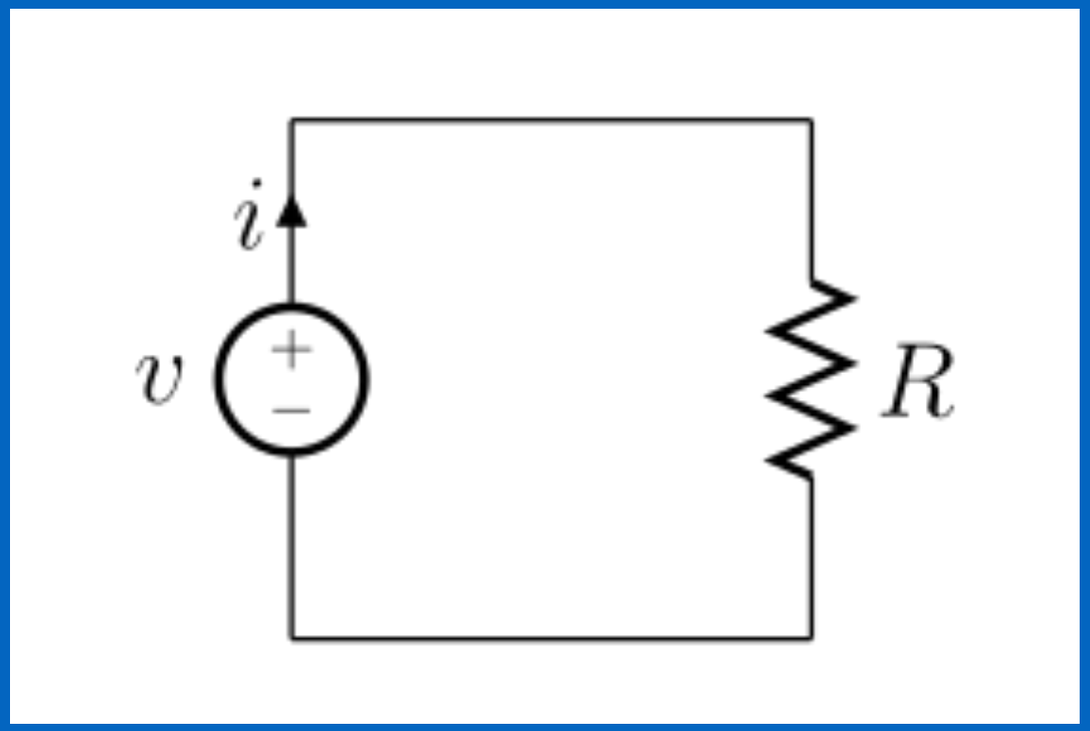
Voltage (V) -> Difference in electric potential energy between two points.

Current (I) - > flow of electric charge

Resistance (Ω) -> Opposition to the passage of electric current



Voltage, Current & Resistance

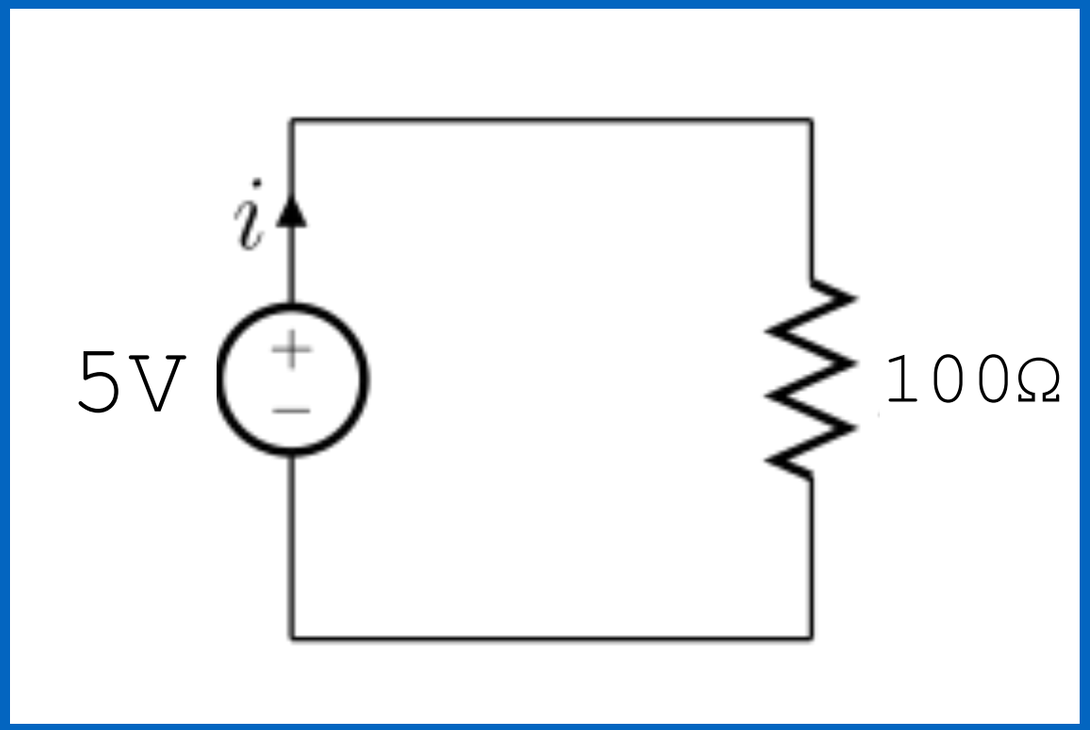


Ohm's Law

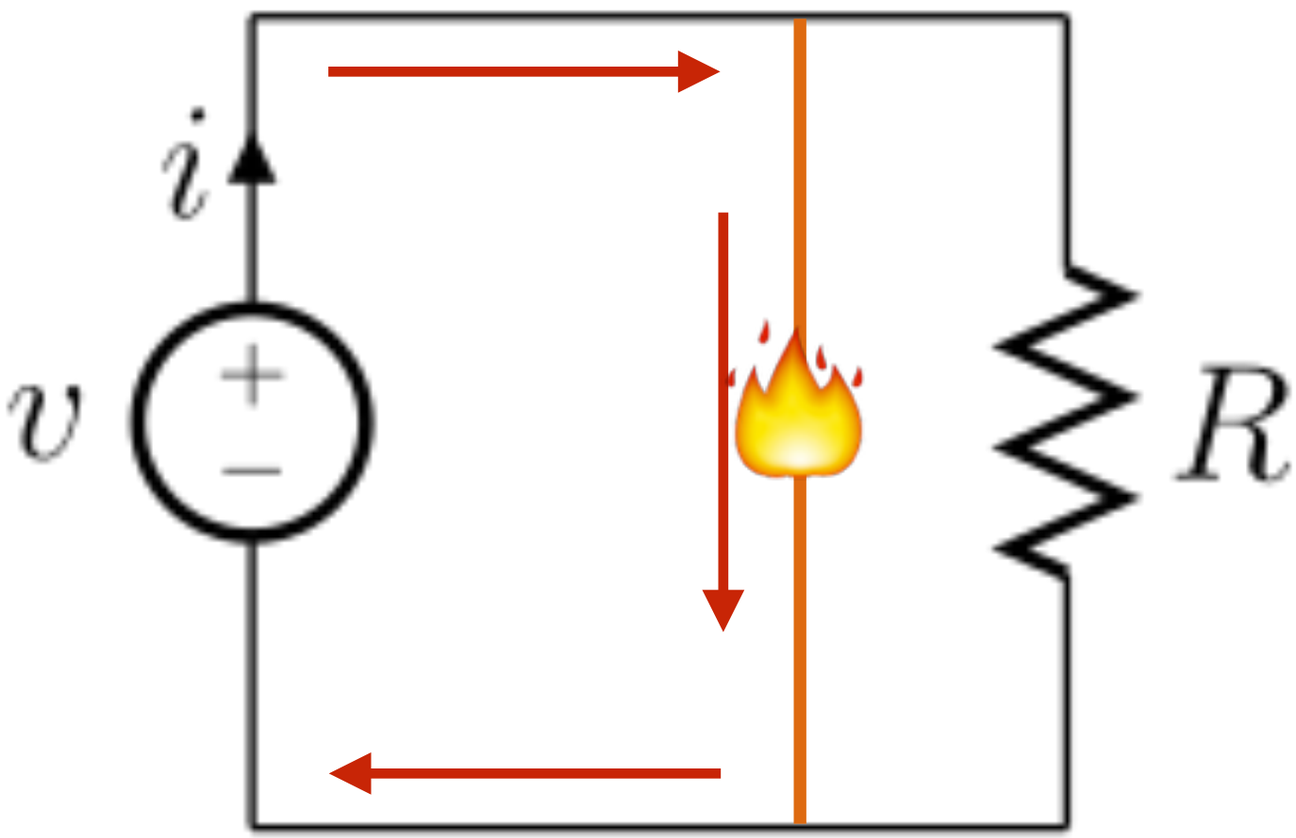
$$V = IR$$

Ohm's Law

$$V = IR$$

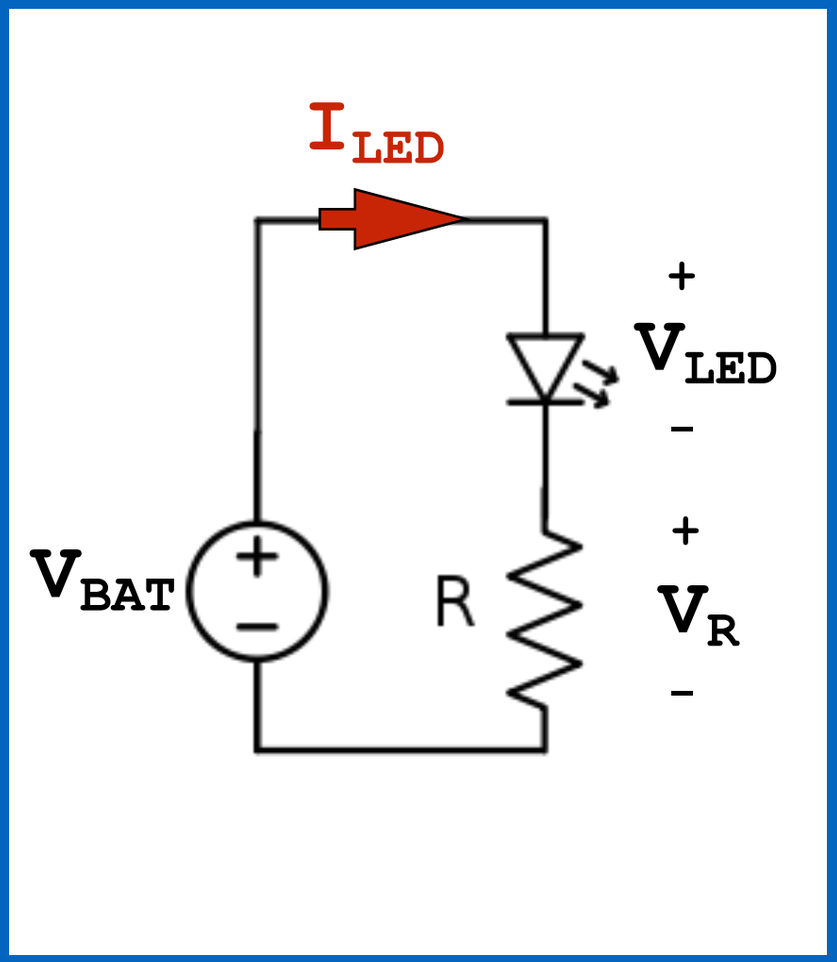


Short Circuits



Path of least resistance!

Ohm's Law Example with LED



LED forward voltage (V_{LED}): 1.7V
LED forward current (I_{LED}): 20mA
Battery voltage (V_{BAT}) = 5V
R = ?

voltage across R:

$V_R = V_{BAT} - V_{LED}$

$V_R = 5V - 1.7V = 3.3V$

Resistance:

$V = IR$

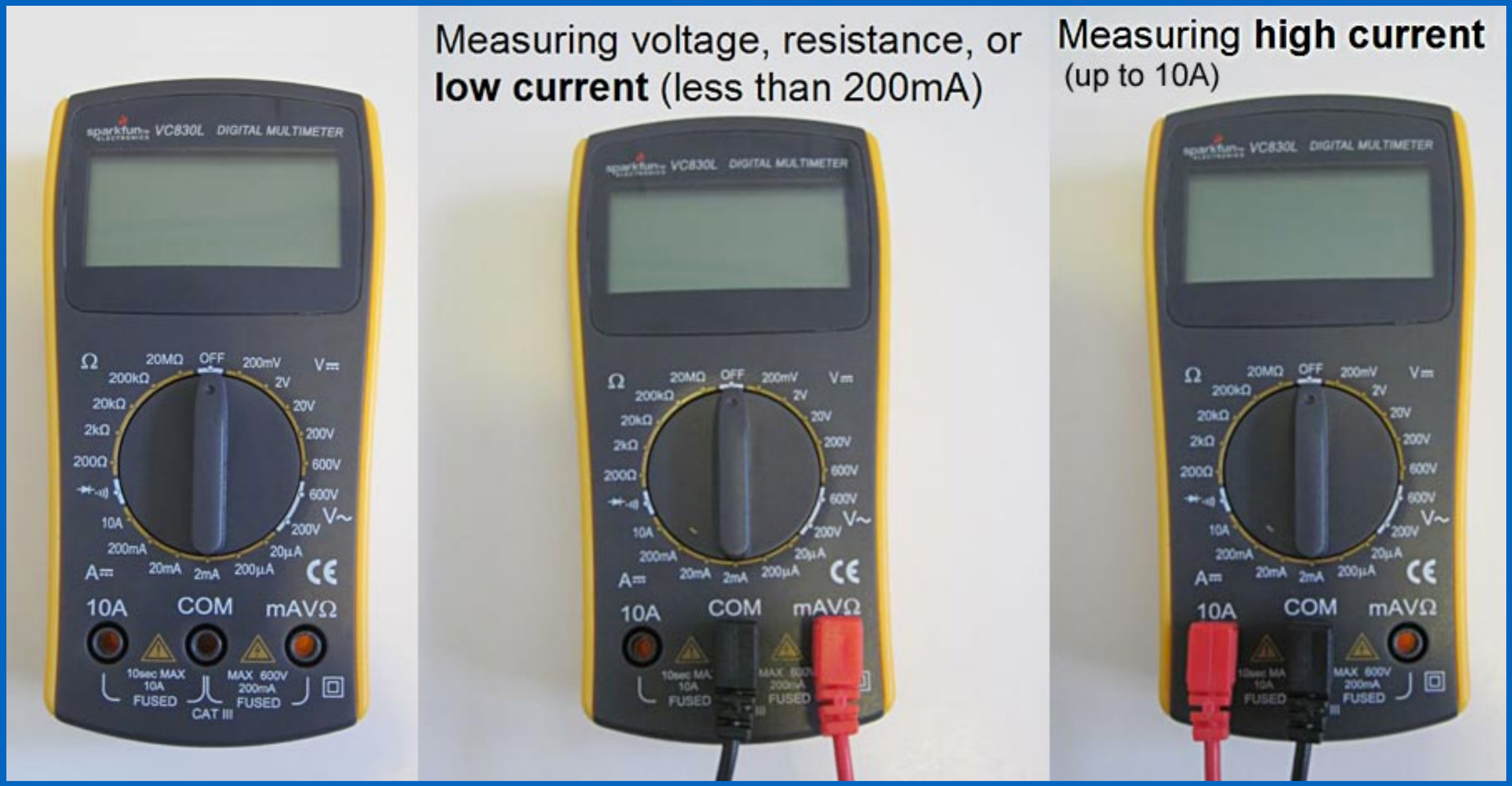
$R = V / I$

$R = V_R / I_{LED}$

$R = 3.3V / 20mA$

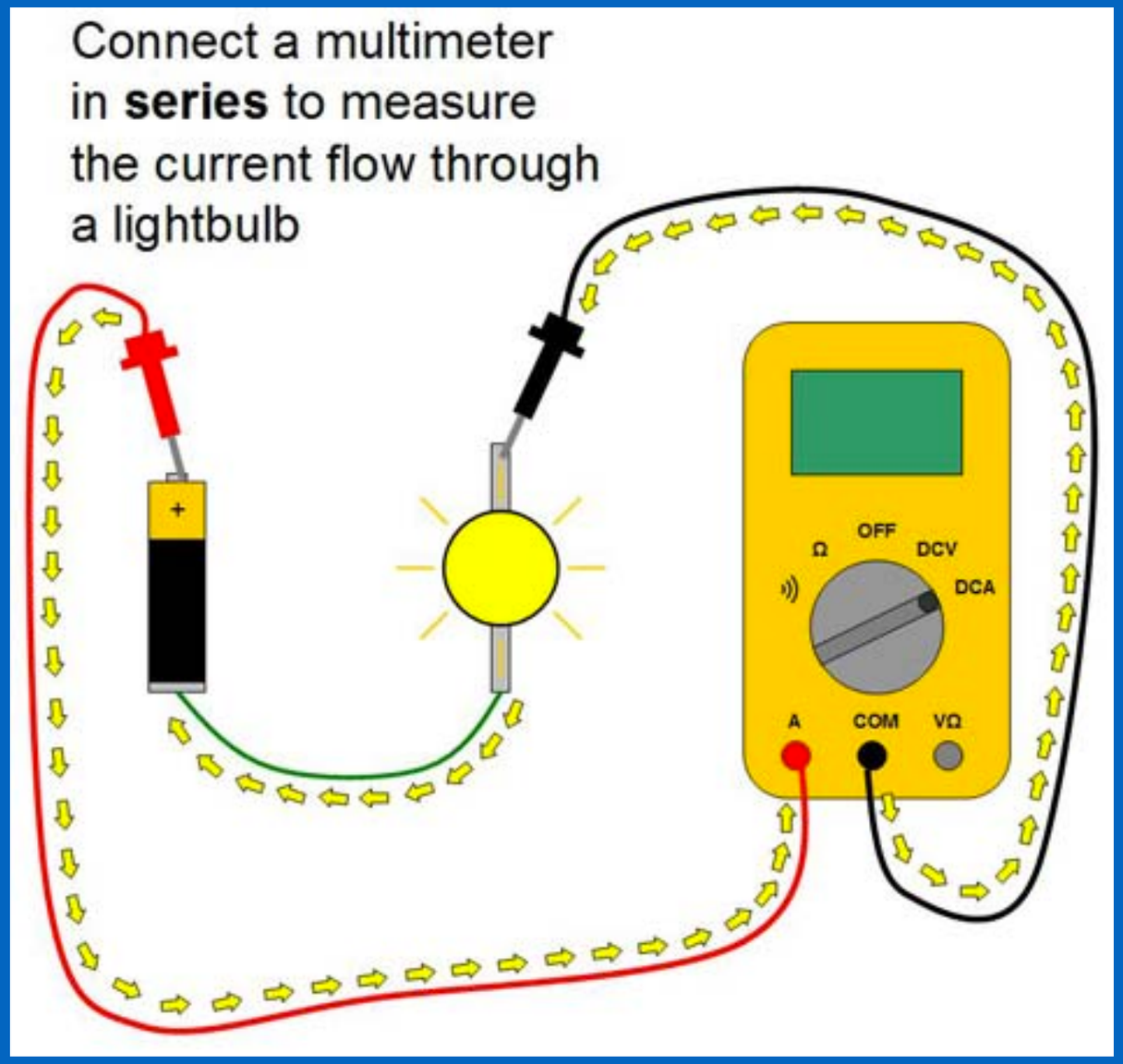
$R = 165\Omega$

Using a Multimeter



images thanks to www.sciencebuddies.org

Measuring Current



Theoretical Values:

V_{BAT} = 5V

I_{LED}: 20mA

R = 165Ω

image thanks to www.sciencebuddies.org

Measuring Voltage

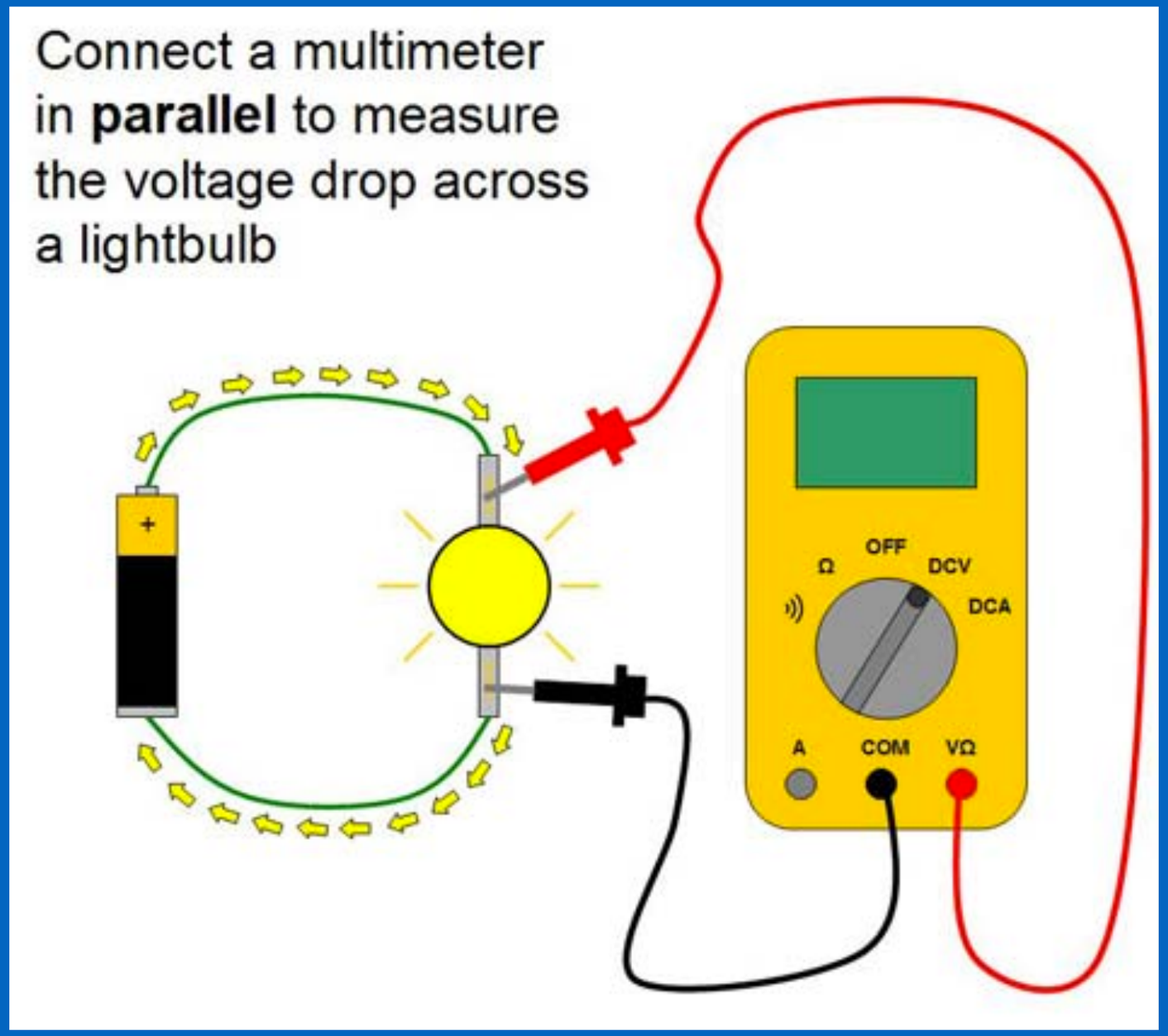
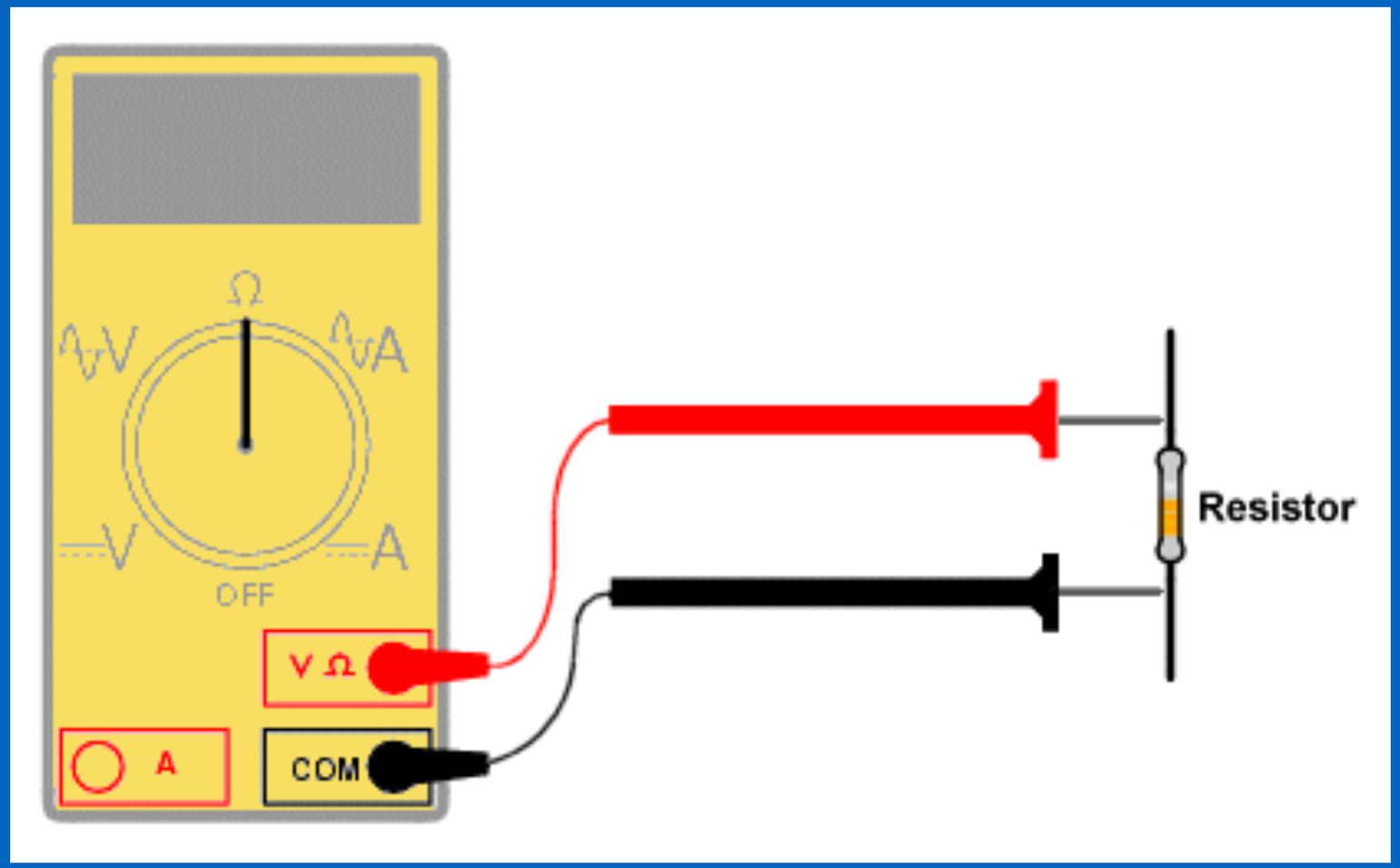


image thanks to www.sciencebuddies.org

Measuring Resistance



Detecting Continuity

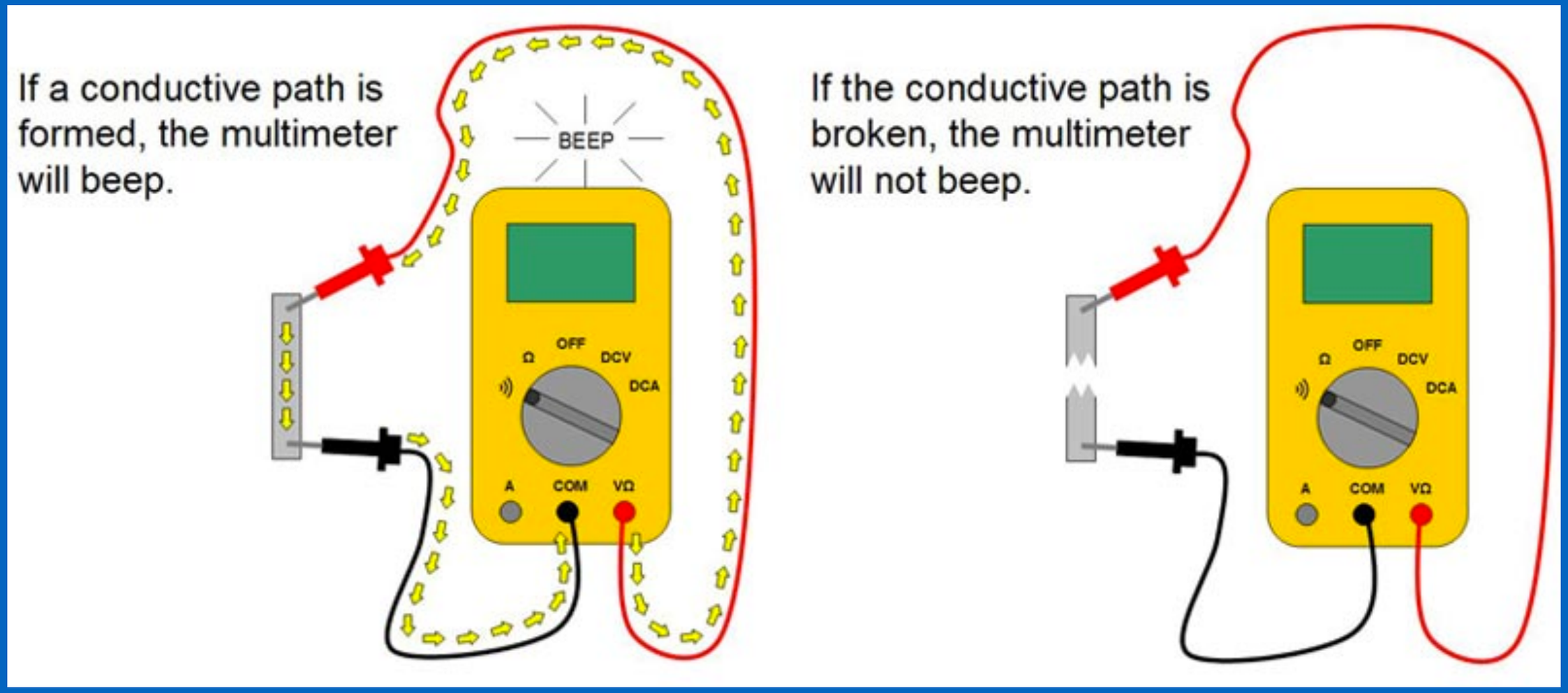


image thanks to www.sciencebuddies.org

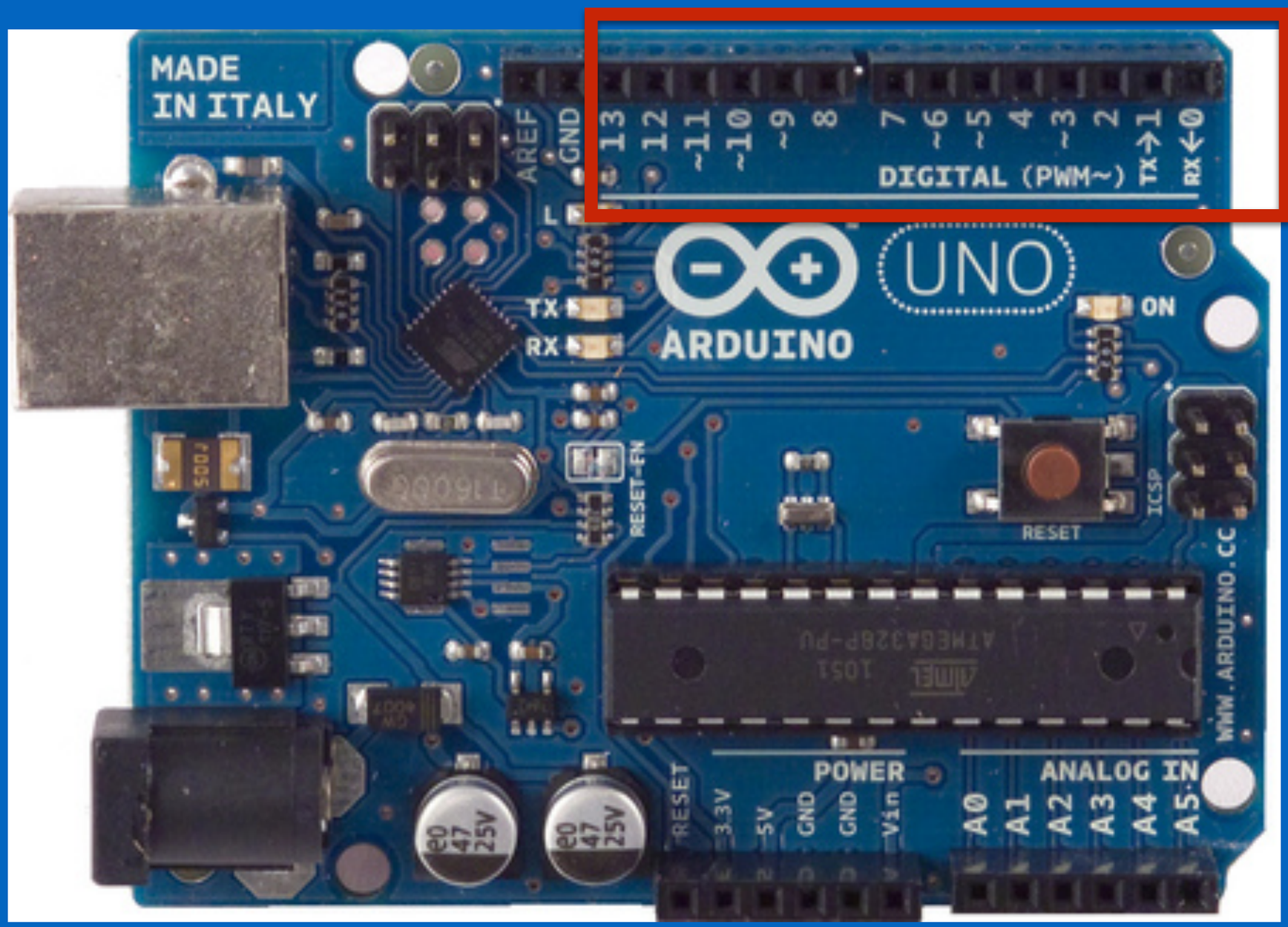
The Arduino IDE

1.6.5

Driver (probably) required!

https://www.arduino.cc/en/Main/Software

Arduino Digital Pins



Arduino Inputs

```
void setup() {
  pinMode(2, INPUT);
}

void loop() {
  if (digitalRead(2) == HIGH) {
    // Do something when Pin is high
  }
}
```

Arduino Inputs

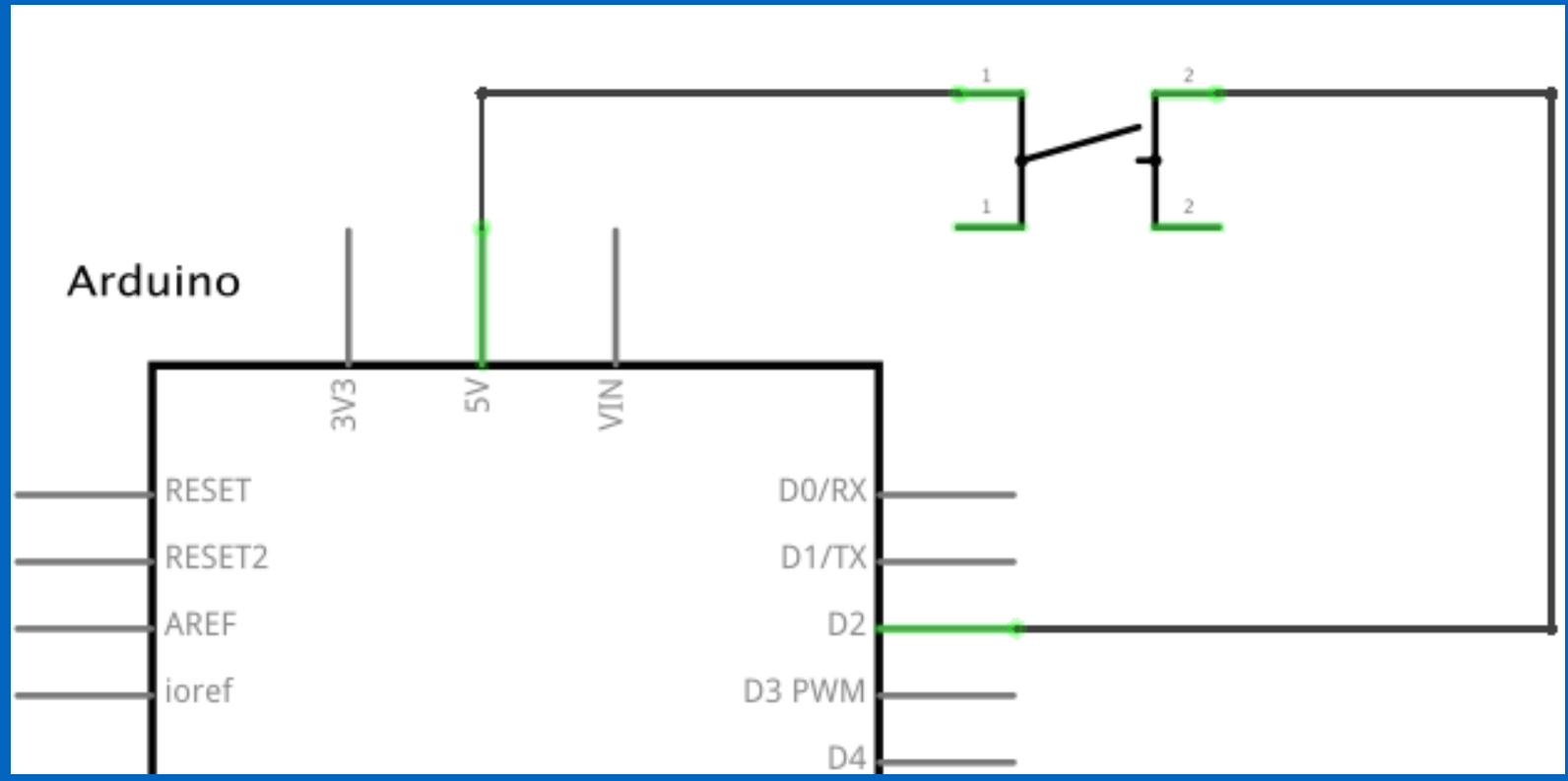
```
void setup() {
 pinMode(2, INPUT);
}

void loop() {
   if (digitalRead(2) == HIGH) {
     // Do something when Pin is high
   }
}
```

Arduino Inputs

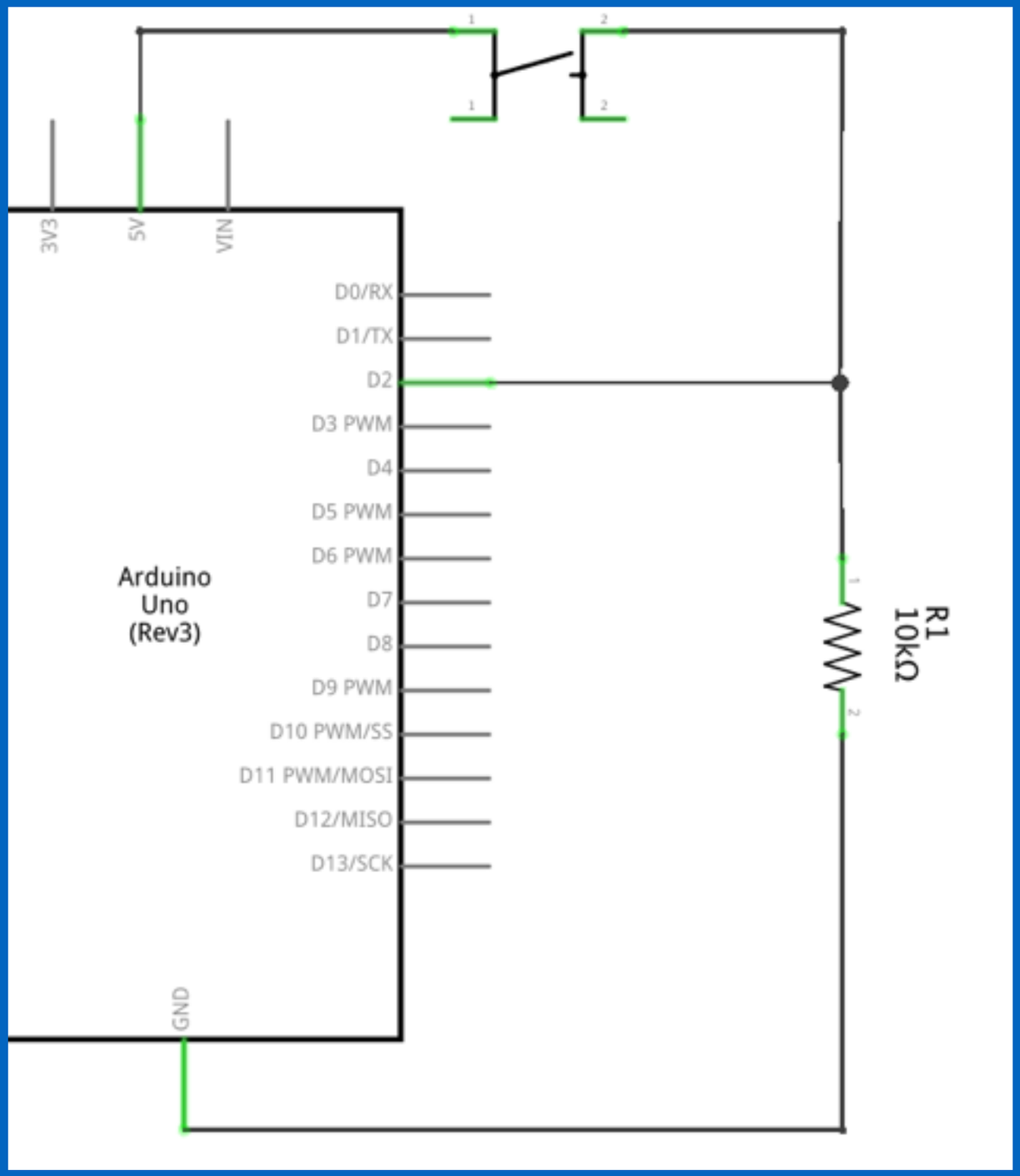
```
void setup() {
  pinMode(2, INPUT);
}

void loop() {
  if (digitalRead(2) == HIGH) {
    // Do something when Pin is high
  }
}
```

Arduino Inputs

push button circuit



Pull Down Resistor



Set a default value!

Arduino Outputs

```
void setup() {
  pinMode(4, OUTPUT);
}

void loop() {
  digitalWrite(4, HIGH);

   // OR
   // digitalWrite(4, LOW);
}
```

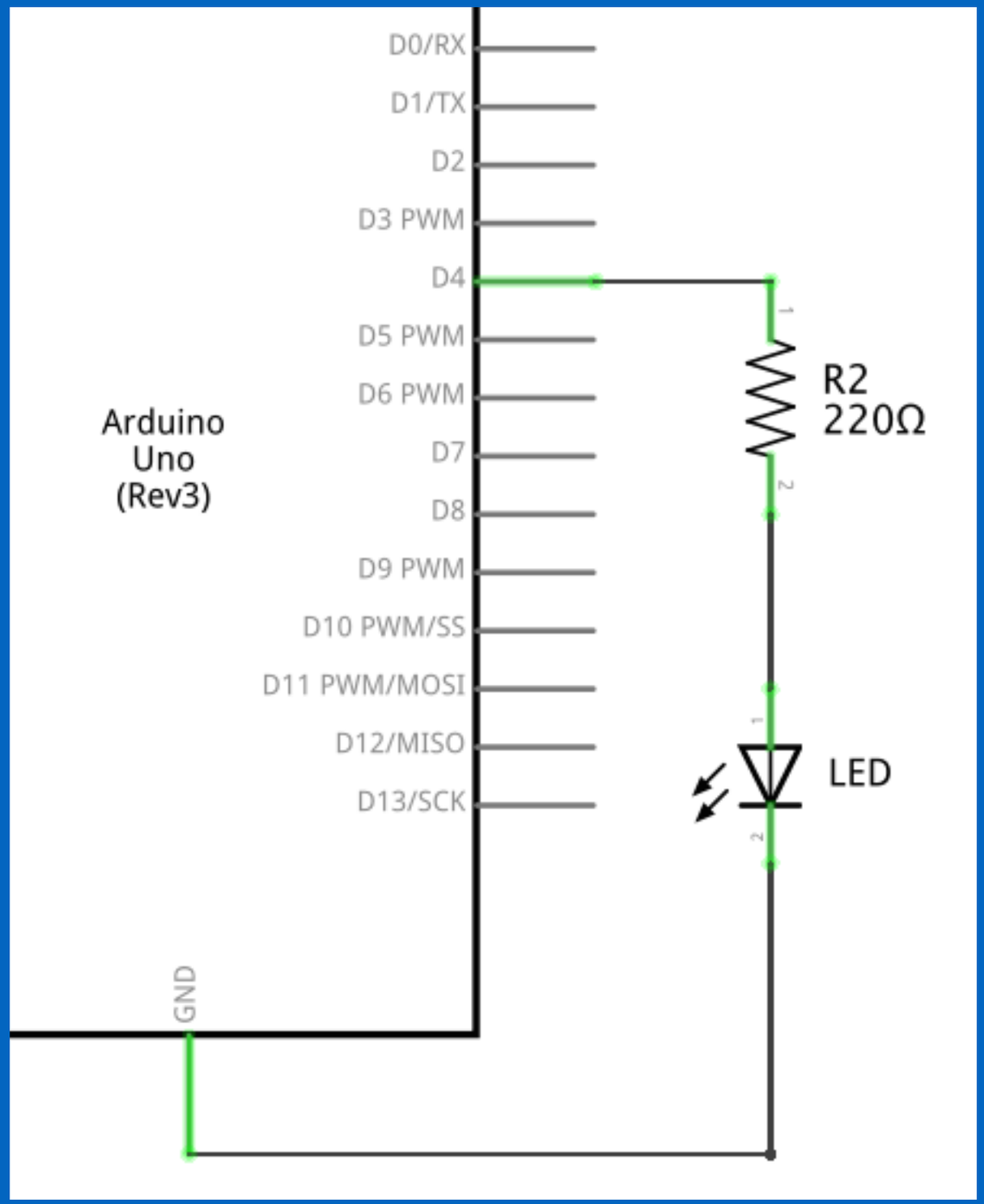
Arduino Outputs

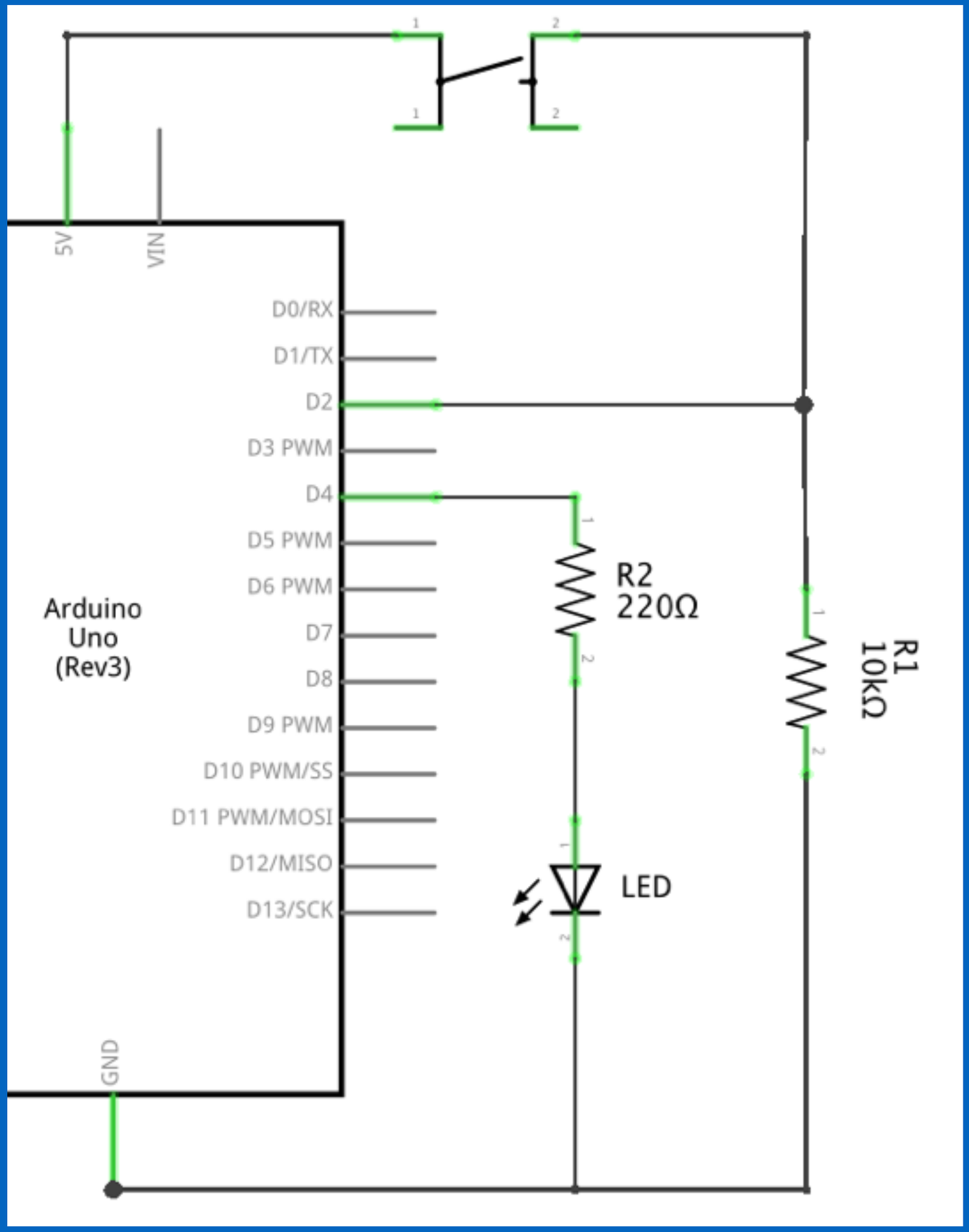
```
void setup() {
  pinMode(4, OUTPUT);
}

void loop() {
  digitalWrite(4, HIGH);

   // OR
   // digitalWrite(4, LOW);
}
```

Arduino Outputs

```
void setup() {
  pinMode(4, OUTPUT);
}

void loop() {
  digitalWrite(4, HIGH);

   // OR
   // digitalWrite(4, LOW);
}
```

Arduino Outputs





LED & Pushbutton

LED & Pushbutton

```
void setup() {
  pinMode(2, INPUT);    // Button
  pinMode(4, OUTPUT);   // LED
}

void loop() {
  if (digitalRead(2) == HIGH) {
    digitalWrite(4, HIGH); // LED ON
  } else {
    digitalWrite(4, LOW);  // LED OFF
  }
}
```

LED & Pushbutton

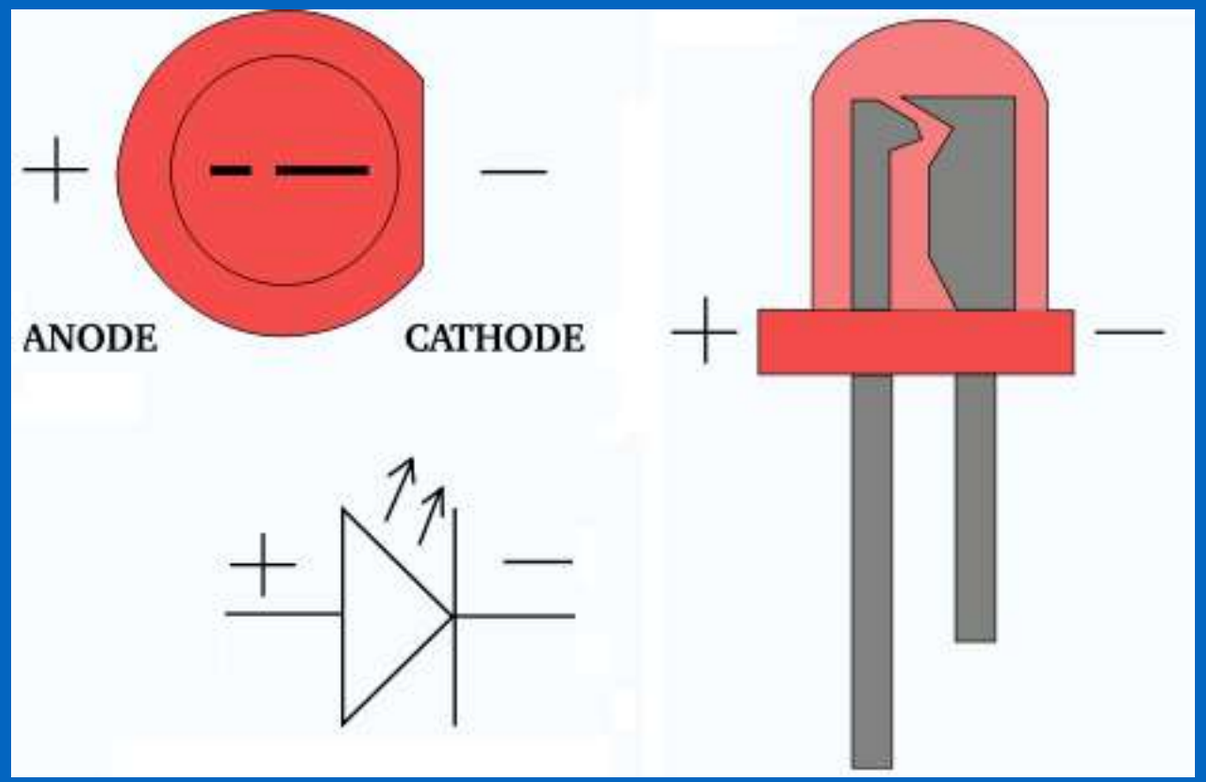
```
void setup() {
  pinMode(2, INPUT);    // Button
  pinMode(4, OUTPUT);   // LED
}

void loop() {
  if (digitalRead(2) == HIGH) {
    digitalWrite(4, HIGH); // LED ON
  } else {
    digitalWrite(4, LOW);  // LED OFF
  }
}
```

LED & Pushbutton

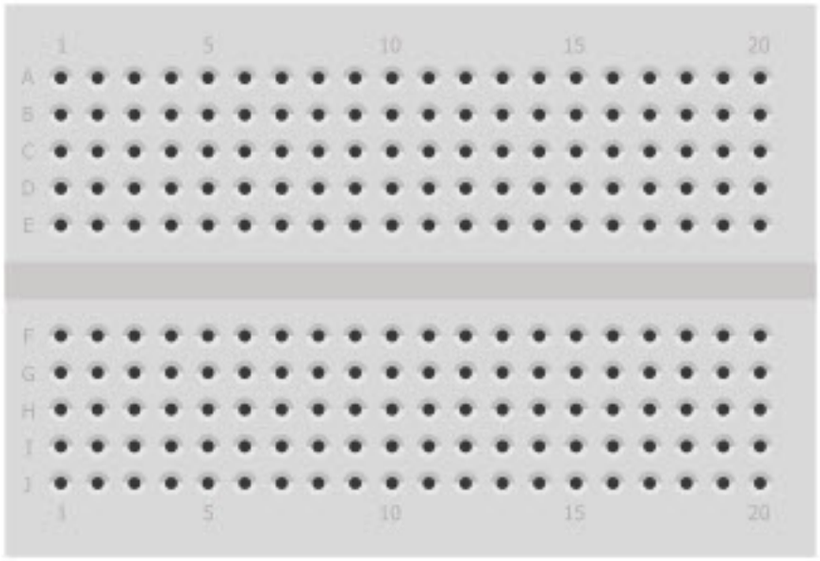
```
void setup() {
  pinMode(2, INPUT);    // Button
  pinMode(4, OUTPUT);   // LED
}

void loop() {
  if (digitalRead(2) == HIGH) {
    digitalWrite(4, HIGH); // LED ON
  } else {
    digitalWrite(4, LOW);  // LED OFF
  }
}
```

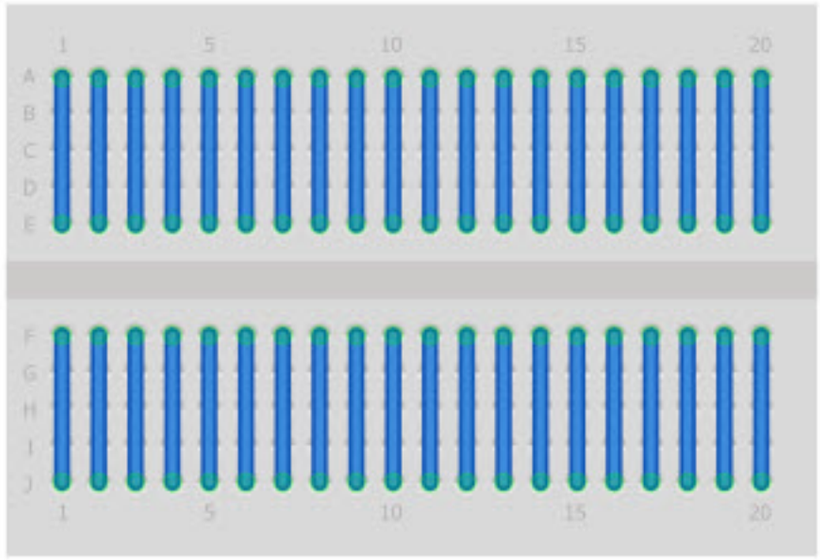
Reminders!

LEDs are polarized!

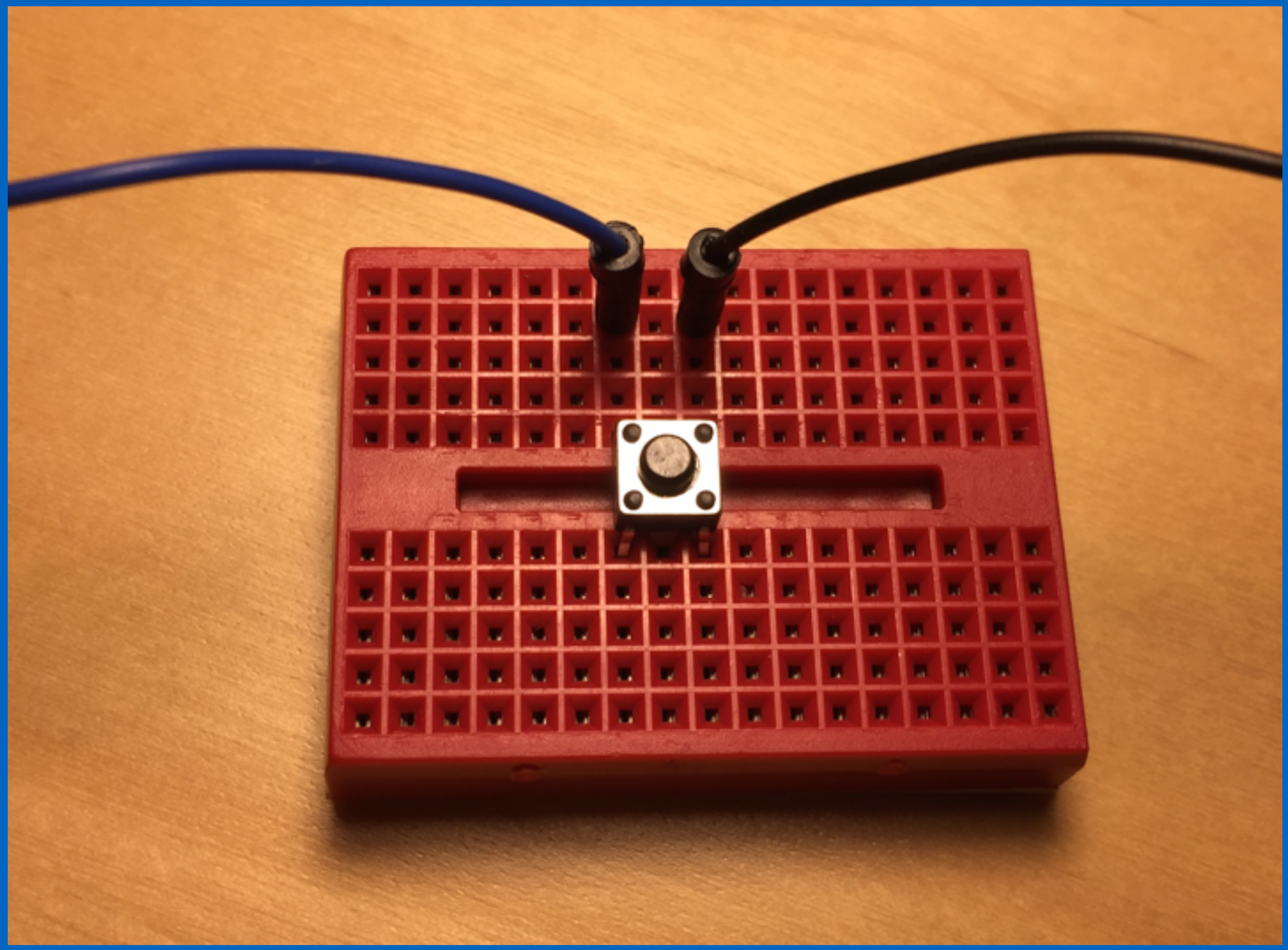


Breadboard Rows are Connected Electrically





Connect the button like this!



Intermission!

- Download materials from Github
- Install Arduino IDE (Version 1.6.5)
- Install CH340 driver
- Open the blink sketch, and modify it so that the LED blinks 10 times faster
- Build LED pushbutton circuit, upload the code and try it out
- Try out a multimeter

Done? Try out the other buzzer events example.

Force Sensitive Resistor

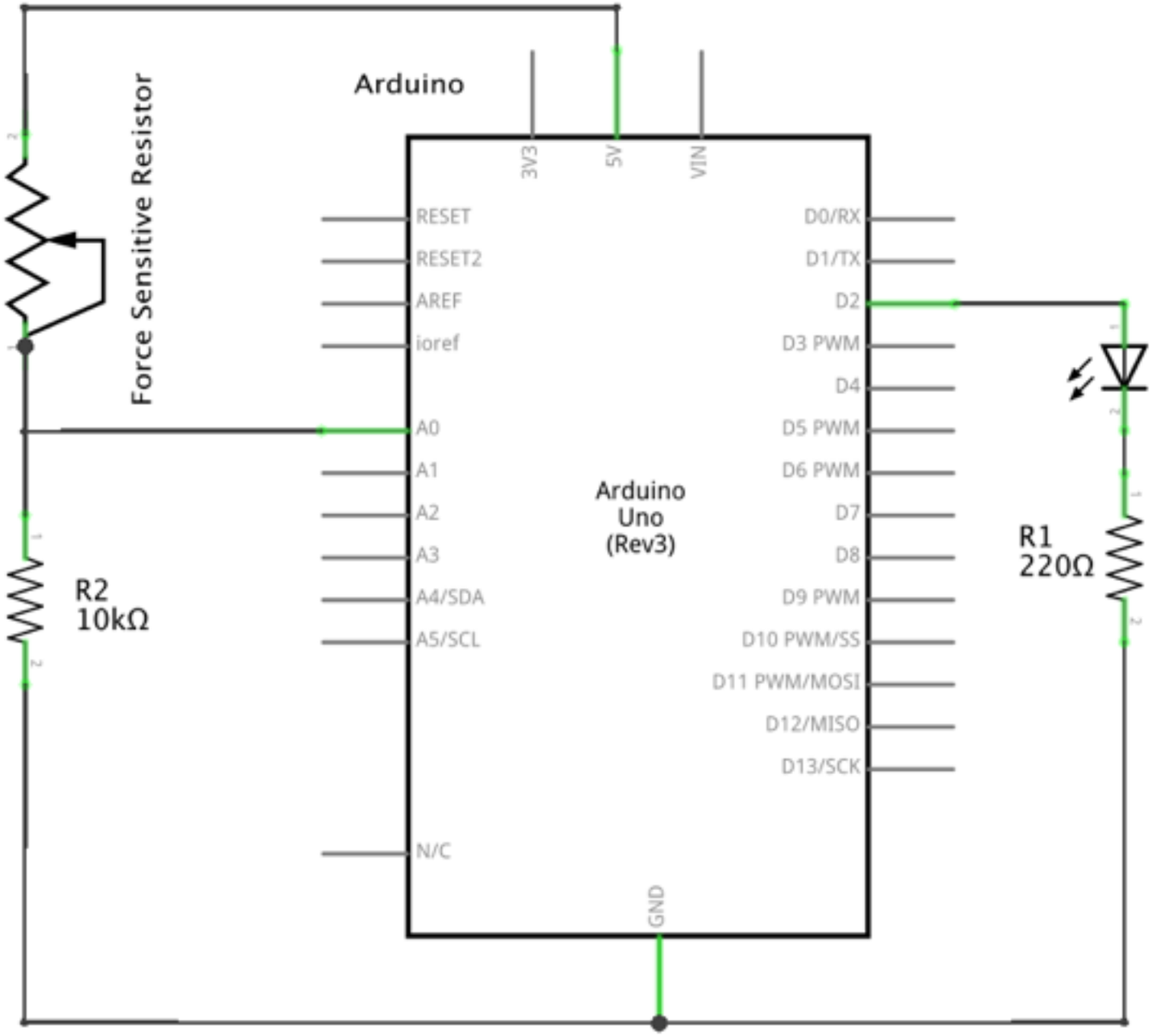
Analog vs Digital

Digital: Finite number of possible values. (commonly, 0V **or** 5V).

Analog: Infinite number of possible values (commonly ranges between 0V and 5V)

Force Sensitive Resistor

The Circuit - Resistive Voltage Divider



FSR Resistance:
2.5kΩ - 1MΩ

Voltage on pin A0:
0.05-4V

Use variables for your pin numbers.

```
int fsrPin = A0;
int ledPin = 3;

int fsrMin = 0;
int fsrMax = 818;

void setup() {
  // We'll send debugging info via the Serial monitor
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  pinMode(fsrPin, INPUT);
}

void loop() {
  int fsrReading = analogRead(fsrPin);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading); // the raw analog reading

  fsrReading = constrain(fsrReading, fsrMin, fsrMax);

  int outputValue = map(fsrReading, fsrMin, fsrMax, 0, 255);
  analogWrite(ledPin, outputValue);

  delay(200);
}
```

(or macros)

```
#define FSR_PIN  A0
#define LED_PIN  3

int fsrMin = 0;
int fsrMax = 818;

void setup() {
  // We'll send debugging info via the Serial monitor
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  pinMode(FSR_PIN, INPUT);
}

void loop() {
  int fsrReading = analogRead(FSR_PIN);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading); // the raw analog reading

  fsrReading = constrain(fsrReading, fsrMin, fsrMax);

  int outputValue = map(fsrReading, fsrMin, fsrMax, 0, 255);
  analogWrite(LED_PIN, outputValue);

  delay(200);
}
```

serial monitoring

```
#define FSR_PIN  A0
#define LED_PIN  3

int fsrMin = 0;
int fsrMax = 818;

void setup() {
  // We'll send debugging info via the Serial monitor
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  pinMode(FSR_PIN, INPUT);
}

void loop() {
  int fsrReading = analogRead(FSR_PIN);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading); // the raw analog reading

  fsrReading = constrain(fsrReading, fsrMin, fsrMax);

  int outputValue = map(fsrReading, fsrMin, fsrMax, 0, 255);
  analogWrite(LED_PIN, outputValue);

  delay(200);
}
```

analog inputs

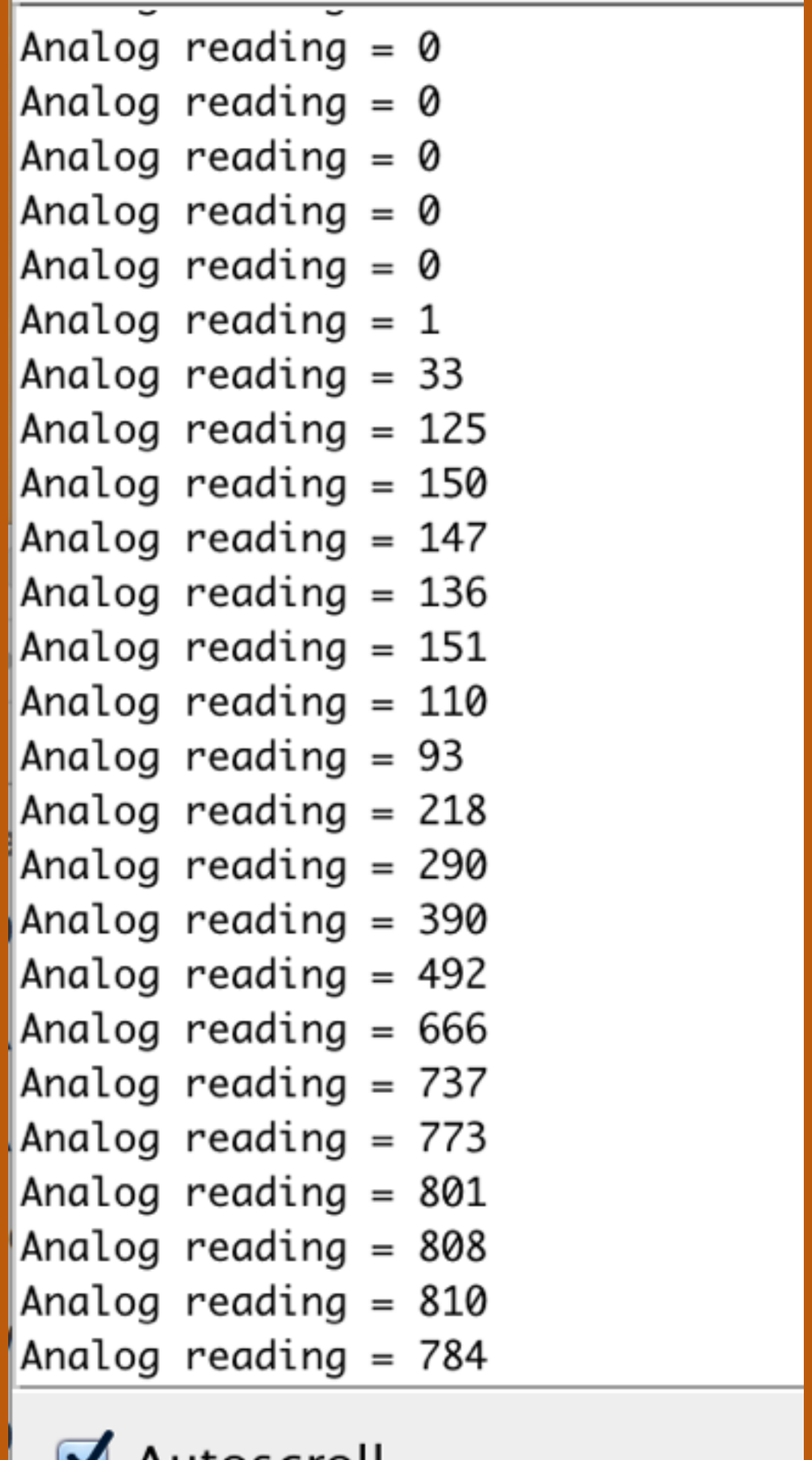
```
#define FSR_PIN  A0
#define LED_PIN  3

int fsrMin = 0;
int fsrMax = 818;

void setup() {
  // We'll send debugging info via the Serial monitor
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  pinMode(FSR_PIN, INPUT);
}

void loop() {
  int fsrReading = analogRead(FSR_PIN);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading); // the raw analog reading

  fsrReading = constrain(fsrReading, fsrMin, fsrMax);

  int outputValue = map(fsrReading, fsrMin, fsrMax, 0, 255);
  analogWrite(LED_PIN, outputValue);

  delay(200);
}
```

analog inputs

```
#define FSR_PIN  A0
#define LED_PIN  3

int fsrMin = 0;
int fsrMax = 818;

void setup() {
  // We'll send debugging info via the Serial monitor
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  pinMode(FSR_PIN, INPUT);
}

void loop() {
  int fsrReading = analogRead(FSR_PIN);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading); // the raw analog reading

  fsrReading = constrain(fsrReading, fsrMin, fsrMax);

  int outputValue = map(fsrReading, fsrMin, fsrMax, 0, 255);
  analogWrite(LED_PIN, outputValue);

  delay(200);
}
```

analog inputs

```
#define FSR_PIN  A0
#define LED_PIN  3

int fsrMin = 0;
int fsrMax = 818;

void setup() {
  // We'll send debugging info via the Serial monitor
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  pinMode(FSR_PIN, INPUT);
}

void loop() {
  int fsrReading = analogRead(FSR_PIN);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading); // the raw analog reading

  fsrReading = constrain(fsrReading, fsrMin, fsrMax);

  int outputValue = map(fsrReading, fsrMin, fsrMax, 0, 255);
  analogWrite(LED_PIN, outputValue);

  delay(200);
}
```



FSR Readings

Analog pin range:
0 - 1023 (0V - 5V)

Range for FSR:
0 - 818 (0V - 4V)

set min & max

```
#define FSR_PIN  A0
#define LED_PIN  3

int fsrMin = 0;
int fsrMax = 818;

void setup() {
  // We'll send debugging info via the Serial monitor
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  pinMode(FSR_PIN, INPUT);
}

void loop() {
  int fsrReading = analogRead(FSR_PIN);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading); // the raw analog reading

  fsrReading = constrain(fsrReading, fsrMin, fsrMax);

  int outputValue = map(fsrReading, fsrMin, fsrMax, 0, 255);
  analogWrite(LED_PIN, outputValue);

  delay(200);
}
```

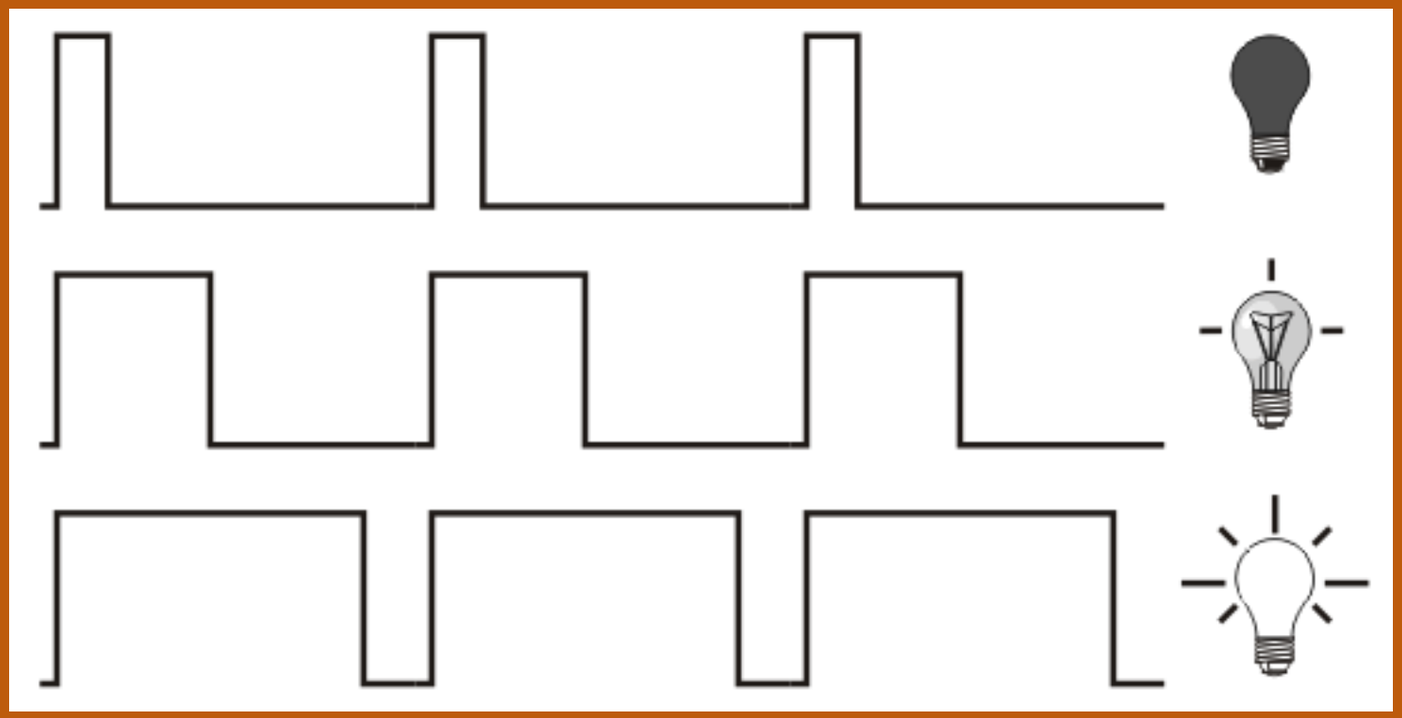
contrain

```
#define FSR_PIN  A0
#define LED_PIN  3

int fsrMin = 0;
int fsrMax = 818;

void setup() {
  // We'll send debugging info via the Serial monitor
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  pinMode(FSR_PIN, INPUT);
}

void loop() {
  int fsrReading = analogRead(FSR_PIN);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading); // the raw analog reading

  fsrReading = constrain(fsrReading, fsrMin, fsrMax);

  int outputValue = map(fsrReading, fsrMin, fsrMax, 0, 255);
  analogWrite(LED_PIN, outputValue);

  delay(200);
}
```

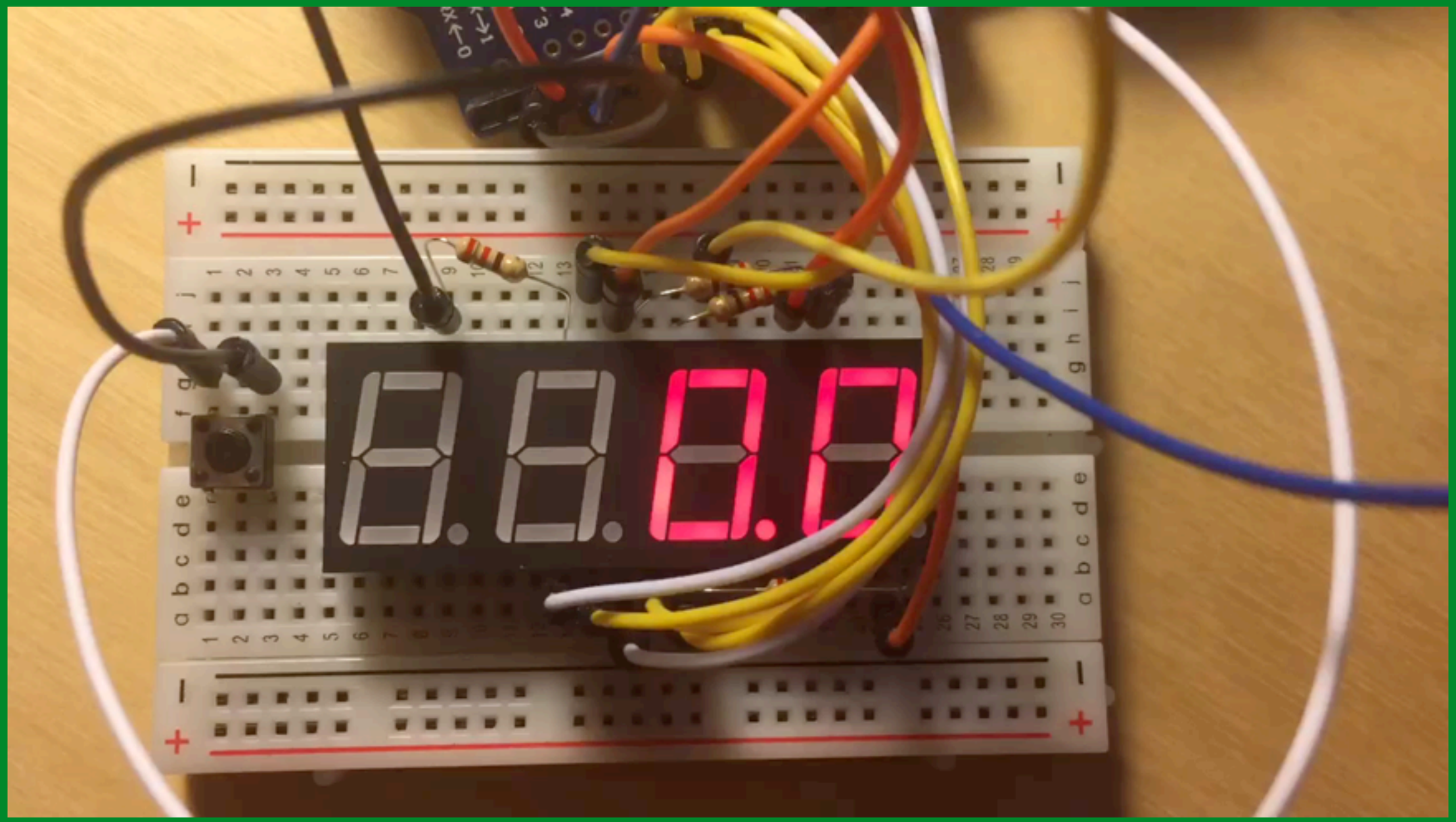
map to output

```
#define FSR_PIN  A0
#define LED_PIN  3

int fsrMin = 0;
int fsrMax = 818;

void setup() {
  // We'll send debugging info via the Serial monitor
  Serial.begin(9600);
  pinMode(LED_PIN, OUTPUT);
  pinMode(FSR_PIN, INPUT);
}

void loop() {
  int fsrReading = analogRead(FSR_PIN);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading); // the raw analog reading

  fsrReading = constrain(fsrReading, fsrMin, fsrMax);

  int outputValue = map(fsrReading, fsrMin, fsrMax, 0, 255);
  analogWrite(LED_PIN, outputValue);

  delay(200);
}
```

Analog Outputs (~) are Pulse Width Modulation (PWM)



Segmented Display

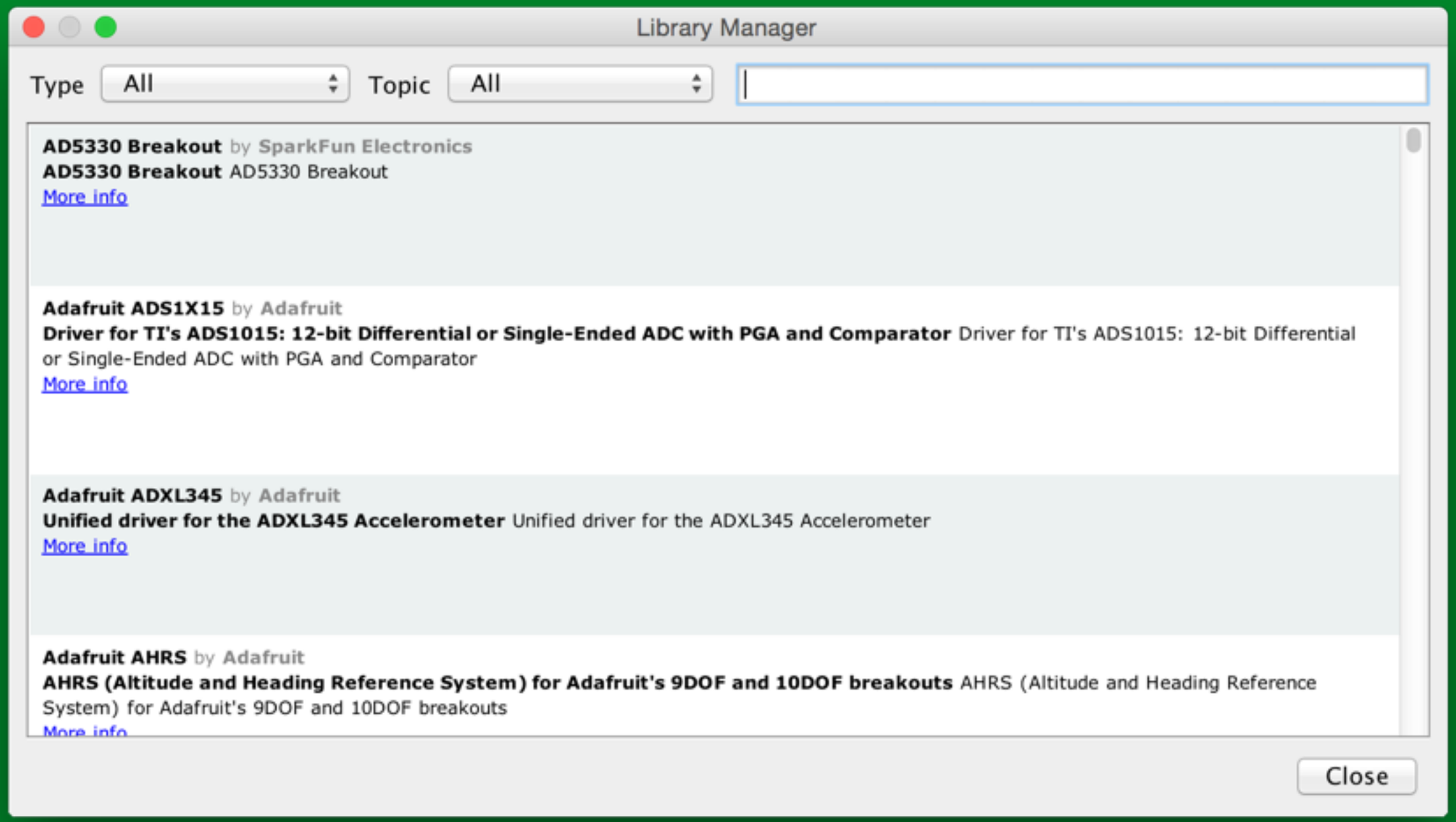


Libraries!

```
#include "SevSeg.h"

SevSeg sevseg;

void setup() {
  byte numDigits = 4;
  byte digitPins[] = {A5, 3, 4, 5};
  byte segmentPins[] = {6, 7, 8, 9, 10, 11, 12, 13};

  sevseg.begin(COMMON_ANODE, numDigits, digitPins,
segmentPins);

  sevseg.setBrightness(90);
}

void loop() {
  sevseg.setNumber(600, 1); // Displays “60.0”

  sevseg.refreshDisplay(); // Must run repeatedly
}
```

Libraries: Option 1 - Arduino IDE

Sketch > Include Library > Manage Libraries

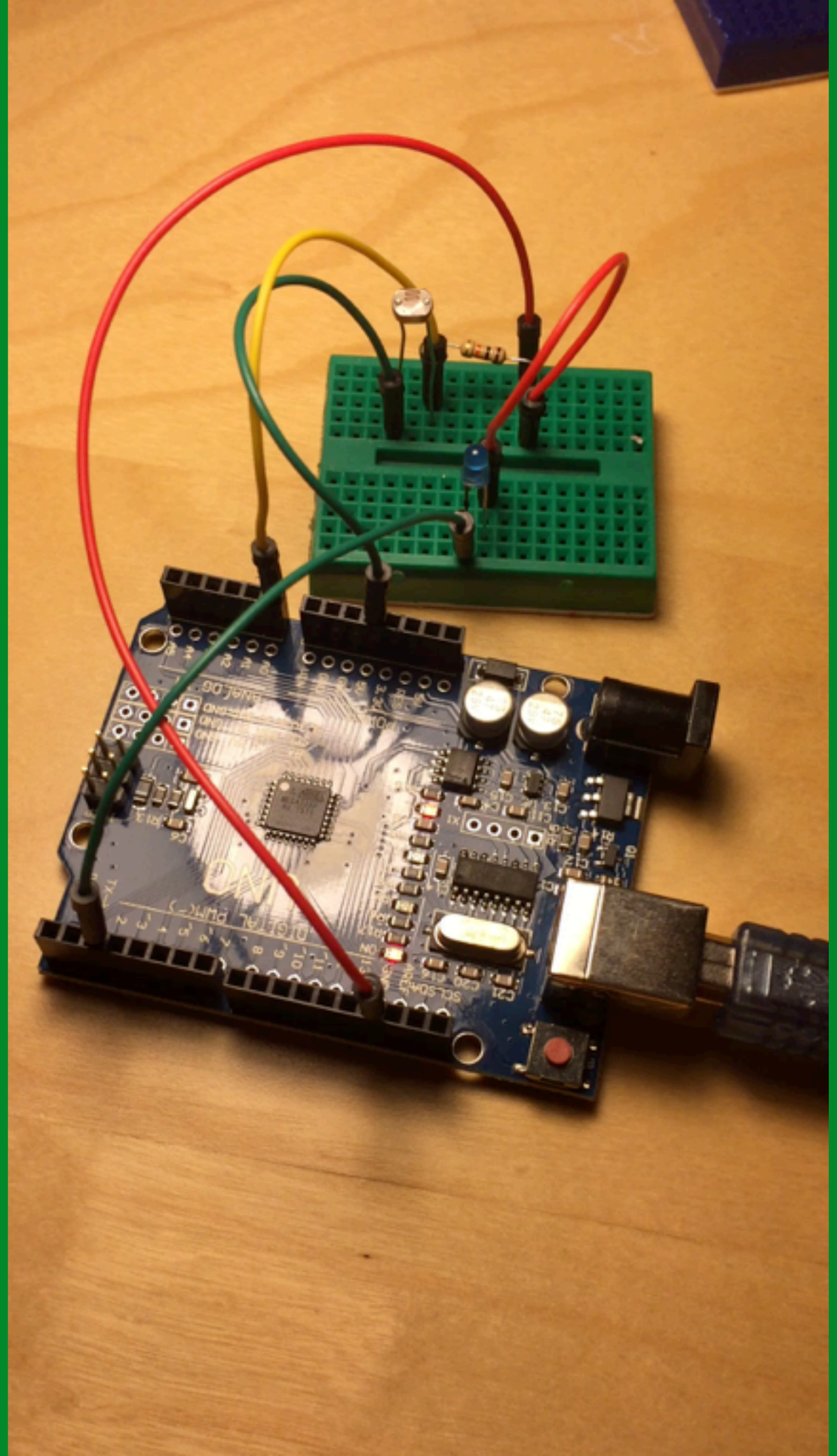


Libraries: Option 2 - Manual

Arduino	Today, 9:46 PM
hardware	Apr 2, 2014, 2:35 PM
hardware-old	Mar 29, 2014, 12:02 AM
libraries	Today, 7:04 PM
I2Cdev	Jan 21, 2014, 2:56 AM
jeelib	May 30, 2014, 9:42 PM
MPU6050	Jan 21, 2014, 2:56 AM
MyCommon	May 30, 2014, 9:41 PM
NewPing	Today, 7:04 PM
NewTone	Today, 7:03 PM
readme.txt	Mar 28, 2014, 11:54 PM
RFM12B	Aug 5, 2014, 12:41 AM
SevSeg	Feb 2, 2015, 2:58 AM
examples	Feb 2, 2015, 2:58 AM
keywords.txt	Feb 2, 2015, 2:58 AM
LICENSE.txt	Feb 2, 2015, 2:58 AM
README.md	Feb 2, 2015, 2:58 AM
SevSeg.cpp	Feb 2, 2015, 2:58 AM
SevSeg.h	Feb 2, 2015, 2:58 AM
TinyDebugSerial	Apr 2, 2014, 6:05 PM
TinyWireS	Apr 2, 2014, 6:58 PM

http://arduino.cc/en/Guide/Libraries

interrupts



photoresistor

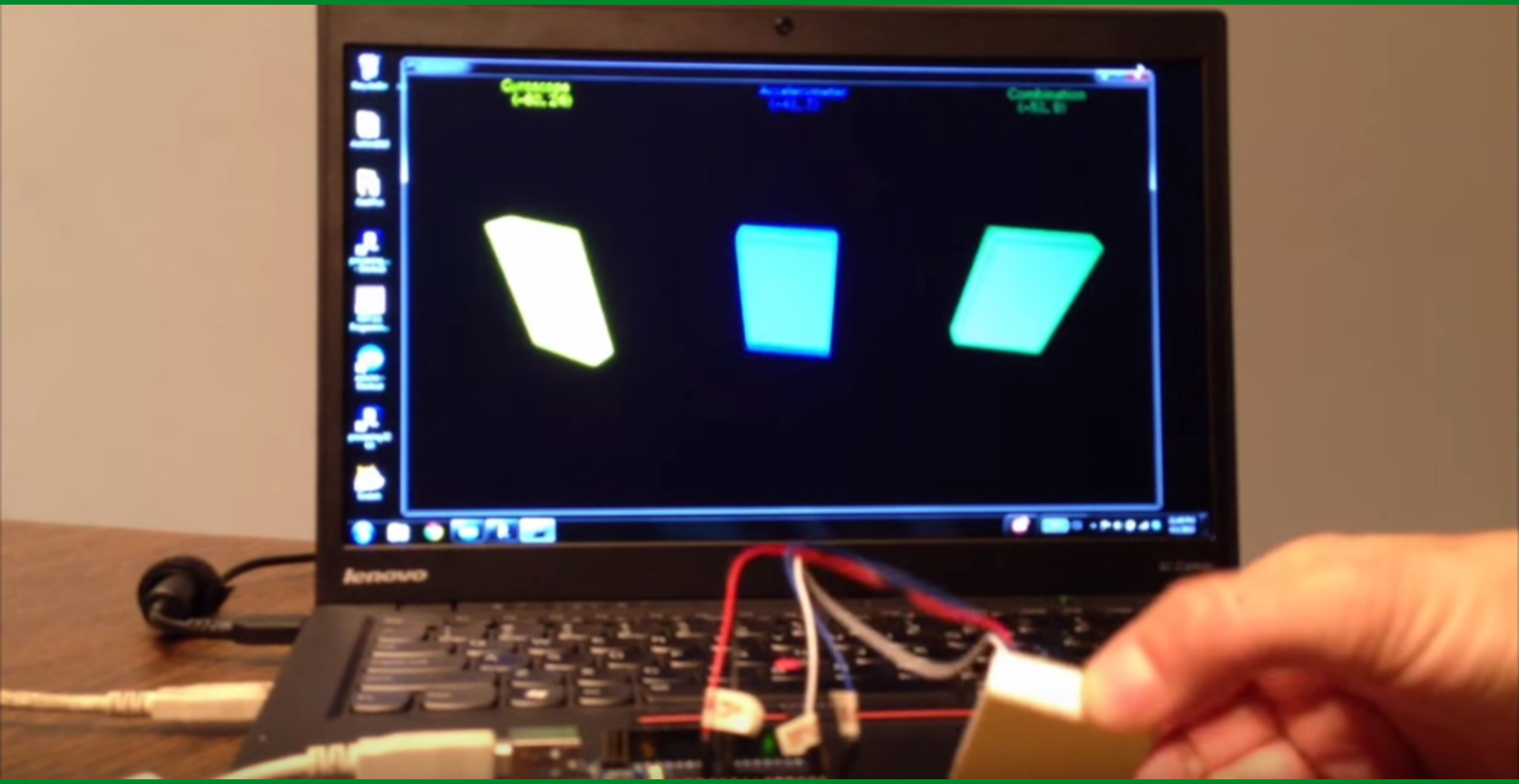
ultrasonic sensor

RGB LED

rgb led code

```
int redPin = 11;
int greenPin = 10;
int bluePin = 9;

void setup() {
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop() {
  setColor(255, 0, 0);  // red
  delay(1000);
  setColor(0, 255, 0);  // green
  delay(1000);
}

void setColor(int red, int green, int blue) {
  red = 255 - red;
  green = 255 - green;
  blue = 255 - blue;

  analogWrite(redPin, red);
  analogWrite(greenPin, green);
  analogWrite(bluePin, blue);
}
```

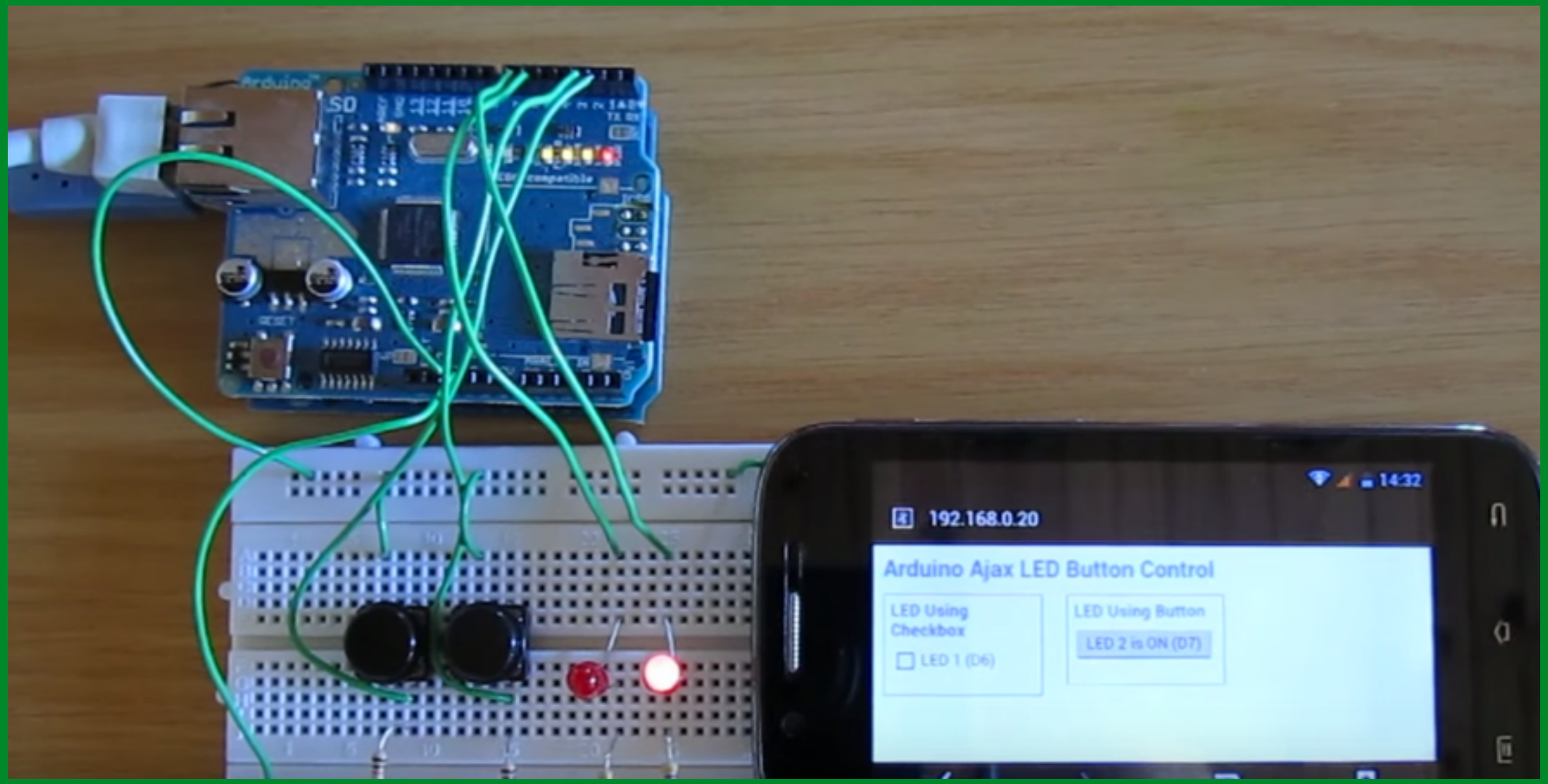
Inertial Measurement Unit (IMU)



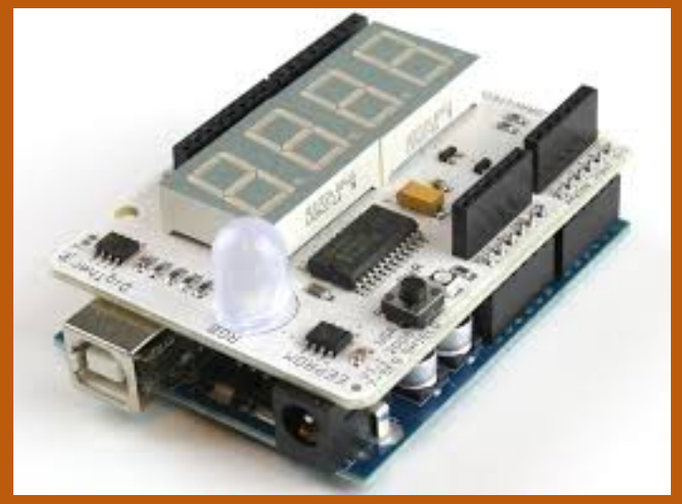
Ethernet

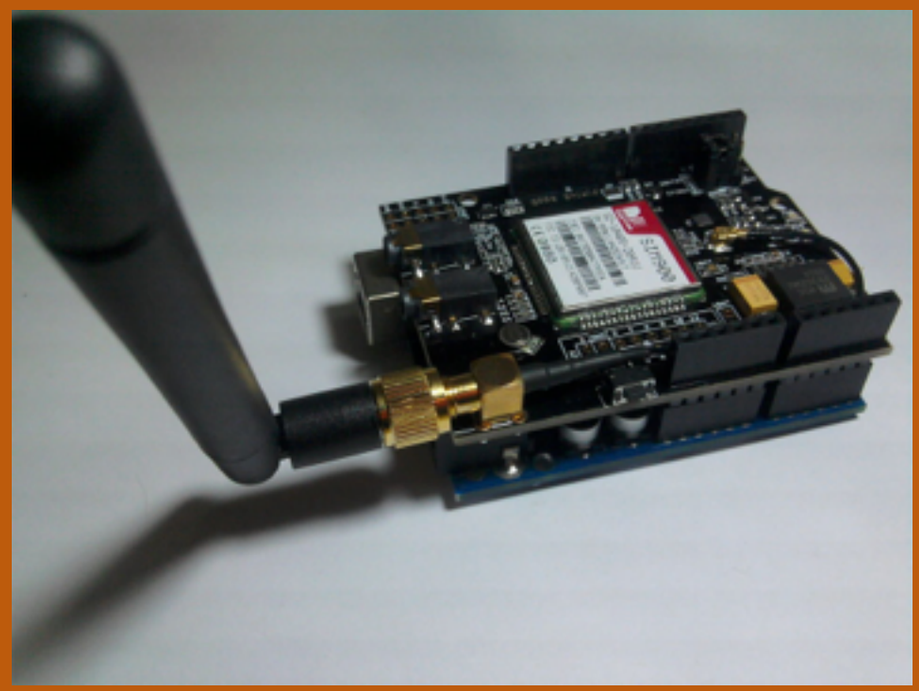
Basic Example - https://www.arduino.cc/en/Tutorial/WebServer

More Sophisticated - https://www.youtube.com/watch?v=la1i8NhXUwE&feature=youtu.be



Shields!





317 on http://shieldlist.org/

Arduino without the IDE

make
make upload
make monitor

https://github.com/sudar/Arduino-Makefile

http://hardwarefun.com/tutorials/compiling-arduino-sketches-using-makefile

Sourcing Components

What's next?

- Check out YouTube for project inspiration & continued learning
- Browse SparkFun!
- Consider learning to solder

Your Turn!!

- Try out some of the samples or projects
- Feel free to come up and borrow any of the sensors or other components
- Mix and match the samples to create your own project

Your Kit:

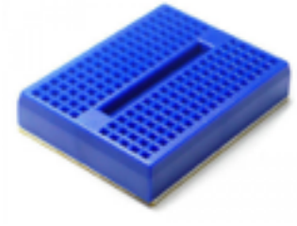
(1) Arduino UNO

(1) USB Cable

(10) M-M Wires

(4) F-F Wires

(1) Mini Breadboard



(3) Push Buttons

(1) RGB LED (Common Anode)

(3) LEDs

(1) Piezo Buzzer

(1) Photoresistor

(4) 220 Ohm Resistors

(3) 10K Ohm Resistors

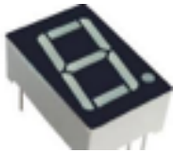
Shared Components:

Ultrasonic Sensors

Tilt Sensors

Breadboards

Segmented Displays



Temperature Sensors

Force Sensitive Resistor

Ethernet Ports

Download course materials:

github.com/bolandrm/arduino_workshop