Arduino Crash Course

ST. MARY'S HIGH SCHOOL

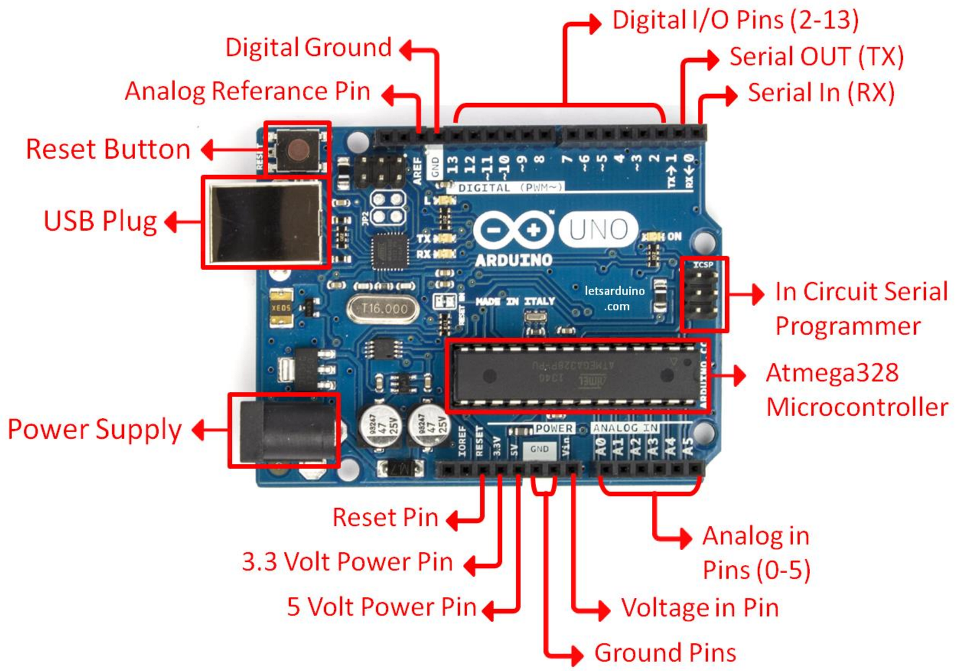
Overview of Today's Tutorial

- What is an Arduino?
- The Arduino UNO board
- Digital vs Analog Signals
- Circuitry basics
  - Navigating a Breadboard
  - A Simple Circuit Schematic
- The Arduino IDE and Language
  - Getting Started
  - Common Functions
- Creating Traffic Lights using the Arduino
- Other Arduino Projects

What is an Arduino?

- Open source platform for building electronic components
- Consists of both hardware and software components
- The Arduino microcontroller is a programmable circuit board, useful for connecting hardware components
- The Arduino IDE is used to program the board and related hardware to perform different tasks
- While fairly simple, the projects that can be made with an Arduino are essentially limitless

The Arduino UNO Board

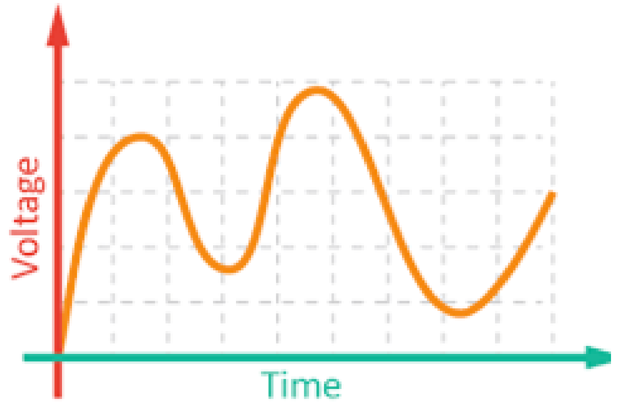


The Arduino UNO Board Summary

- Power supply: Used to connect a battery when deploying the Arduino project
- USB plug: Used to connect the board to the computer to upload code, and for testing. When the board is plugged into the computer, it doesn’t need a power source
- Reset button: Used to reset the board and force it to run its code again from the beginning.
- Digital and Analog I/O pins: Used for connecting and controlling different hardware components

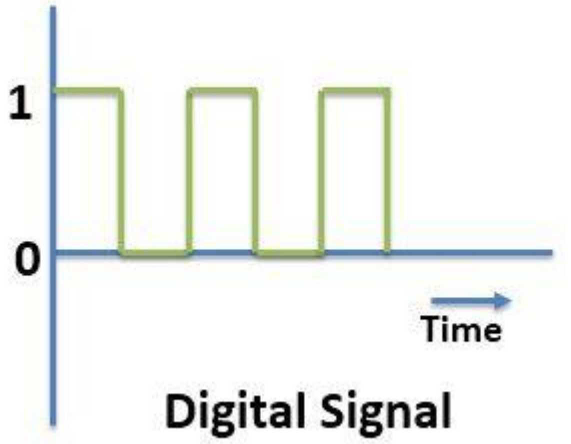
Analog Signals

- Can take on any value
- How most signals in the real world exist
- Example: Me talking right now. My voice is an auditory signal that gets transmitted to your ears. You can hear as I make different sounds, and understand different words.

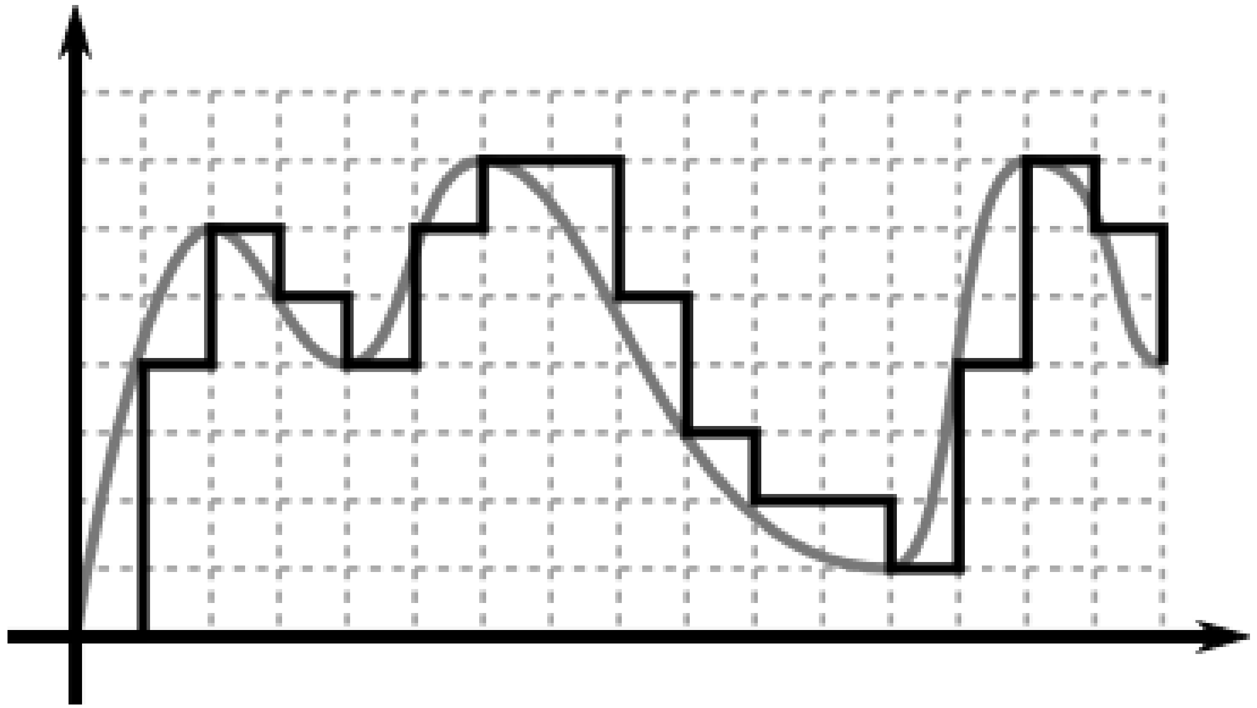


Digital Signals

- Can only take certain values (usually just 0 and 1)
- How most computers are programmed, as it gives them more flexibility
- Going back to the previous example, if my voice was a digital signal, you would either hear me speaking or not hear me speaking, you would not be able to make out words.



Analog vs Digital



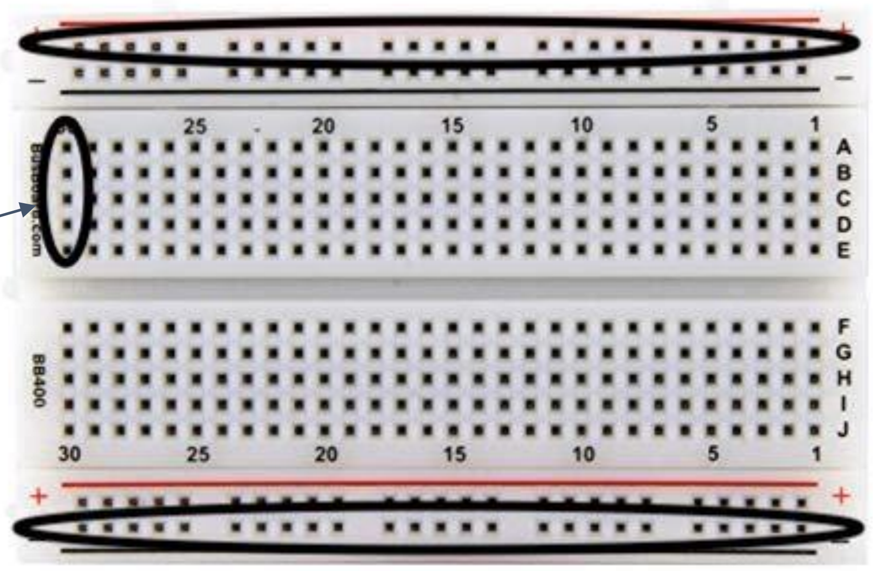
Analog vs. Digital: Why Does it Matter?

- The Arduino board natively has analog and digital pins.
- Analog pins are often used to collect information. They allow us to bypass having to use an analog to digital converter as the Arduino converts the signal for us
- Digital pins can be used as input and outputs of digital signals, and can also be used to output analog signals. Hence the large difference in analog and digital pins
- A digital signal can also be used to output an analog signal through the use of pulse-width modulation (PWM). Pins that can do PWM are indicated by a tilde (~).

Circuitry Basics

- To create a circuit we need to have a continuous loop that connects a power supply to ground.
- To join multiple hardware components together, we use a breadboard

Navigating a Breadboard



Ground line

Connected in series

Power line

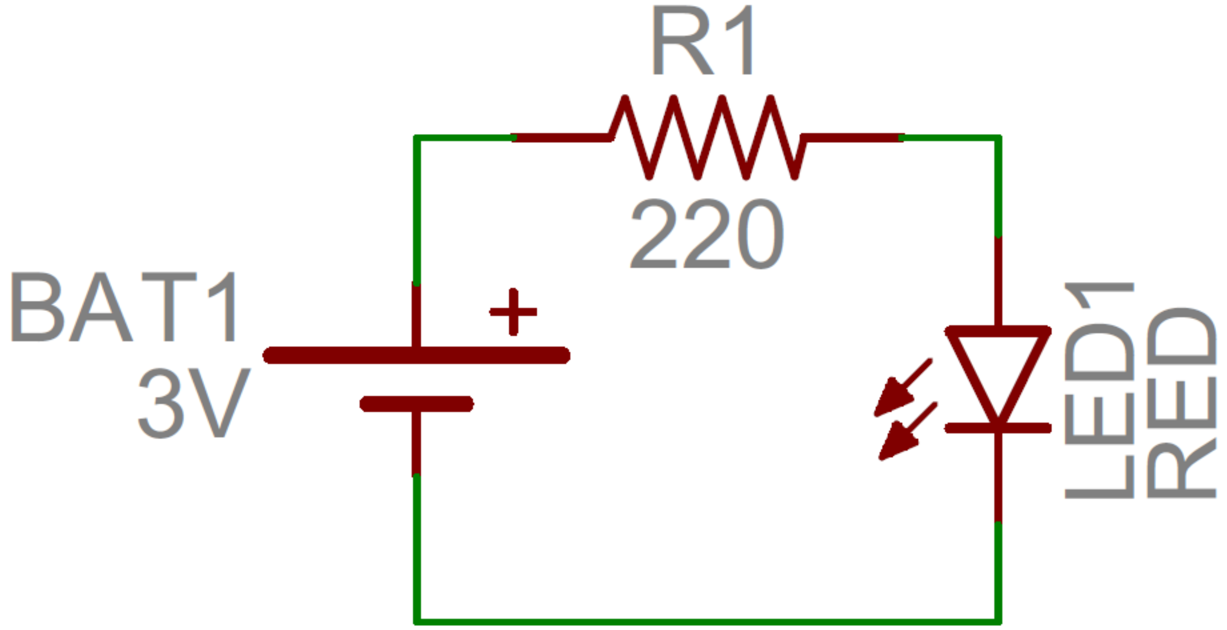
Navigating a Breadboard Summary

- Each numbered column is wired in series, allowing us to connect multiple hardware components without the use of a wire
- Two long connected strips shown by - and + at the top and the bottom of the board are usually used as power and ground lines. These make it easier for us to make circuits by reducing the amount of wiring needed

A Simple Circuit Schematic

- Often, the first step after deciding on a project is developing the circuit schematic for the hardware components.
- A schematic helps us to understand how components interact with each other
- Also helps us maximize space on a breadboard while minimizing wire usage

Traffic Light Circuit



Traffic Light Circuit Summary

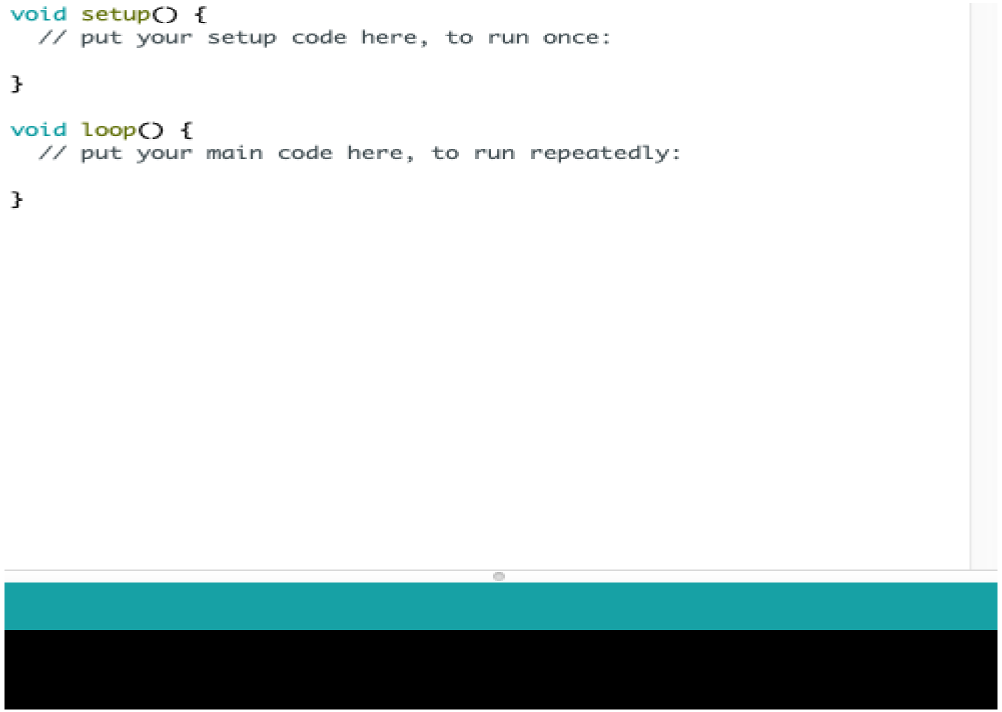
- Battery is replaced with a pin from the Arduino to control the light turning on and off
- Light emitting diode (LED) which turns on when current is flowing from the top.
- Resistor prevents current from flowing. We need to put a small resistor in front of an LED to ensure that it does not have more current than it can handle (common practice is usually 150 ohm resistor)

The Arduino IDE

- C++ based, despite looking very similar to Java and C#.
- Language has a lot of abstractions, making it easy to connect hardware and software.
- Syntax-wise very similar to Java and C# (and Processing)
- Fairly easy to pick up and learn, hence its popularity for making electronic components
- Many similarities between Processing and Arduino

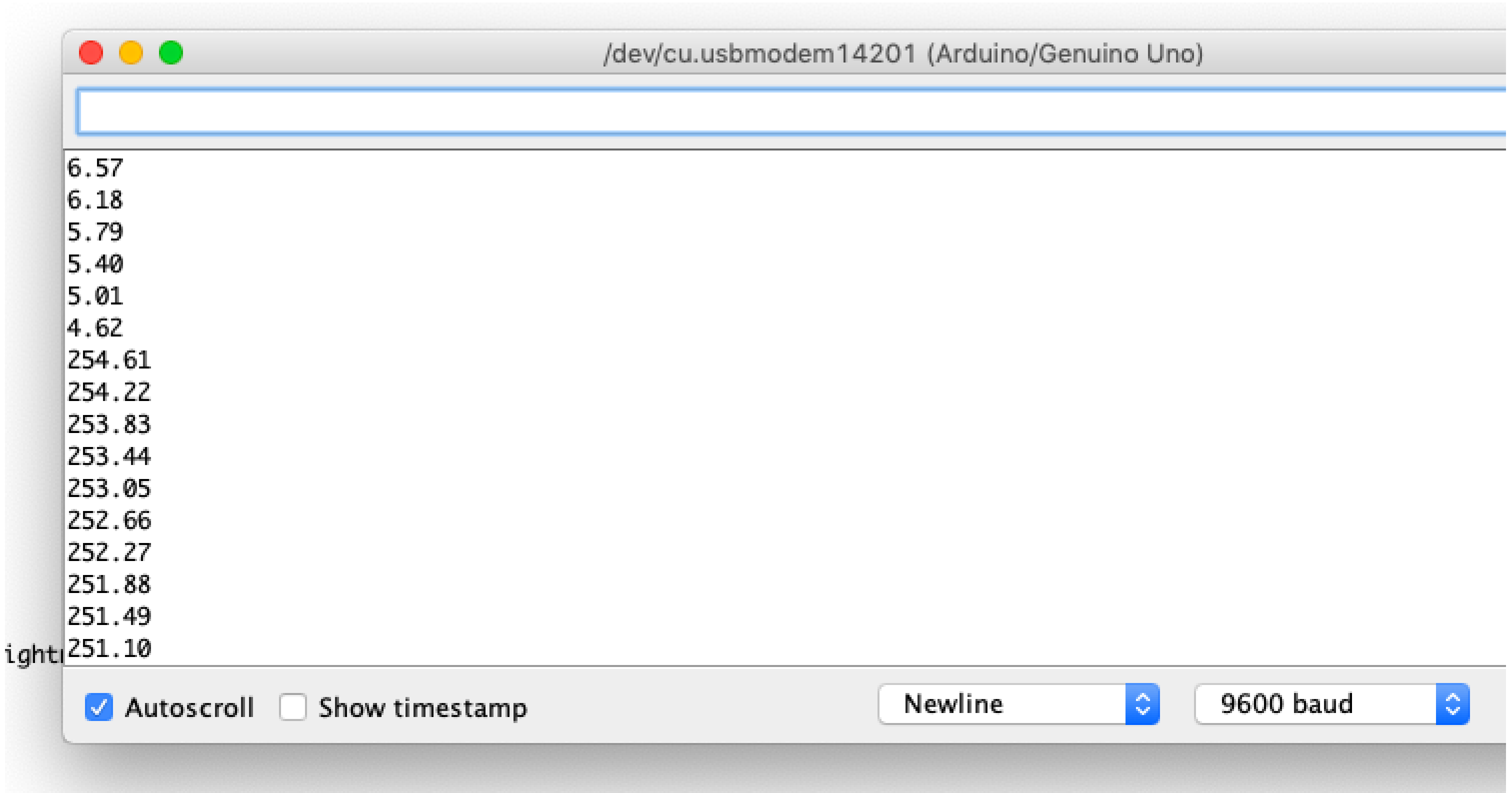
The Arduino IDE

- void setup() and void loop(). Exact same functions as processing
- Terminal at the bottom for debugging and showing errors
- Output terminal has to be opened separately through Tools>Serial Monitor



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Serial Monitor



```
6.57
6.18
5.79
5.40
5.01
4.62
254.61
254.22
253.83
253.44
253.05
252.66
252.27
251.88
251.49
251.10
```

ight

[✓] Autoscroll [ ] Show timestamp Newline 9600 baud

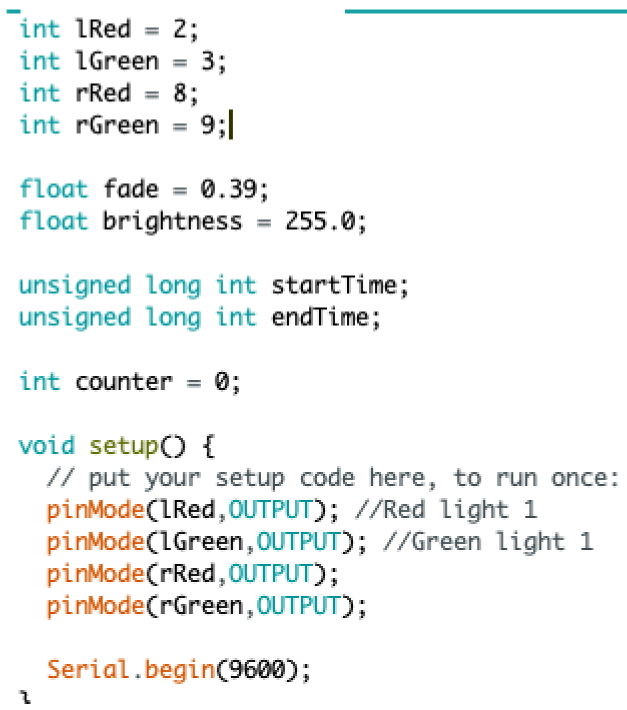
Basic Arduino Functions

- pinMode(pin, mode). Used to set a pin as either output or input. By default, all pins are set to input
- Serial.begin(speed). Begins the serial monitor for printing and general debugging purposes. NOTE: Serial monitor speed must match the serial monitor open, otherwise nothing will show up
- Serial.print(val)/Serial.println(val). Functions to write output to the monitor.

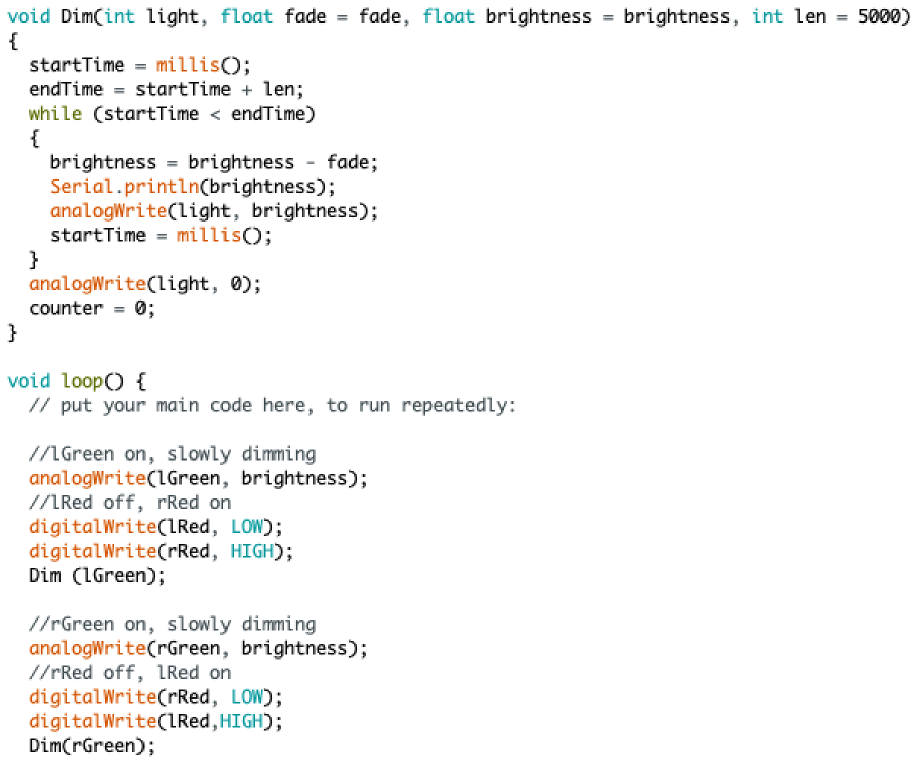
Basic Arduino Functions Continued

- digitalRead(pin)/analogRead(pin). Used for obtaining a value from a pin that was set as an input pin. digitalRead and analogRead can only be used for reading digital signals and analog signals respectively
- digitalWrite(value). Used to write output to a digital output pin. Since it is digital, can only be 0 or 255 (0V or 5V)
- analogWrite(value). Used to write output to an analog output pin. Unlike digital, it can take any value between 0 and 255, and will deliver the correlating amount of volts
- delay(ms). Used to stop the arduino code for running for the specified amount of ms. Useful for debugging and stopping code

Traffic Light Example Extended

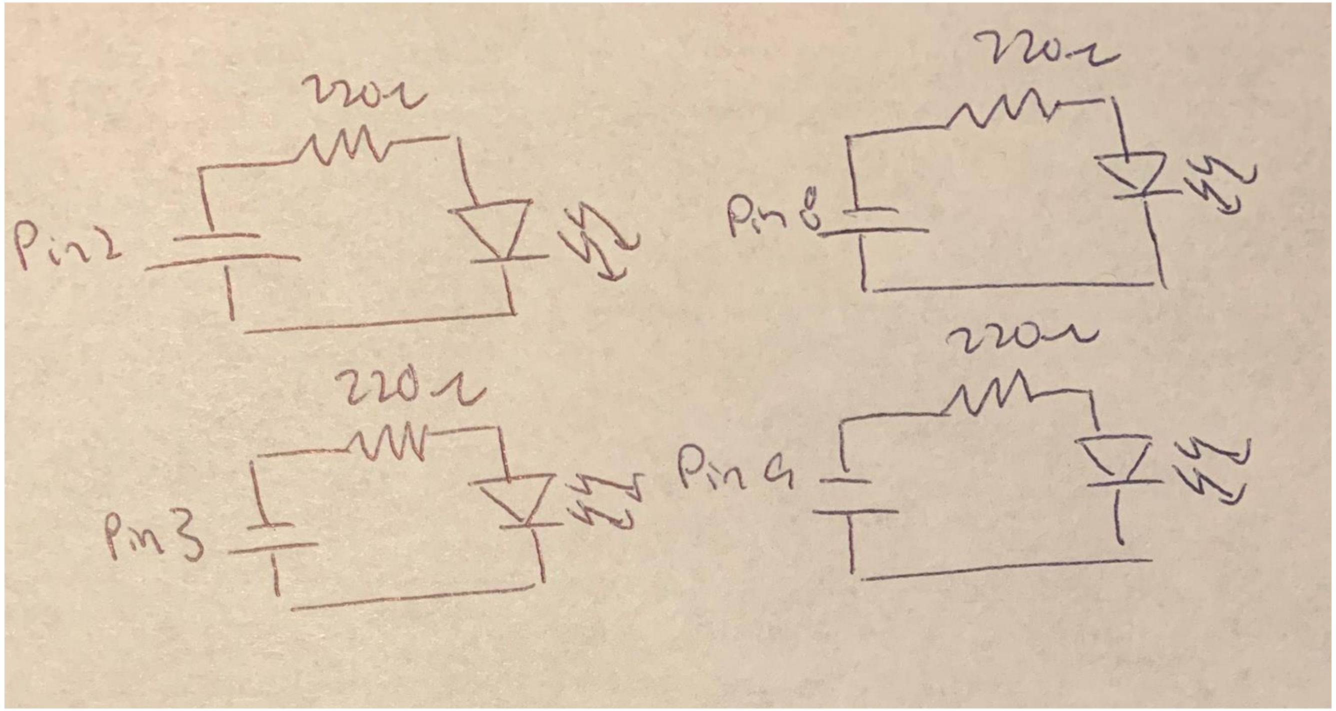


```
int lRed = 2;
int lGreen = 3;
int rRed = 8;
int rGreen = 9;

float fade = 0.39;
float brightness = 255.0;

unsigned long int startTime;
unsigned long int endTime;

int counter = 0;

void setup() {
  // put your setup code here, to run once:
  pinMode(lRed,OUTPUT); //Red light 1
  pinMode(lGreen,OUTPUT); //Green light 1
  pinMode(rRed,OUTPUT);
  pinMode(rGreen,OUTPUT);

  Serial.begin(9600);
}
```

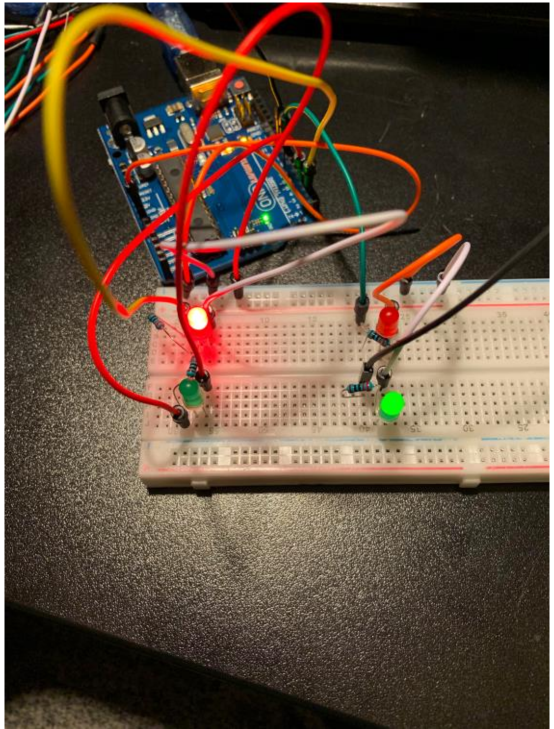


```
void Dim(int light, float fade = fade, float brightness = brightness, int len = 5000)
{
  startTime = millis();
  endTime = startTime + len;
  while (startTime < endTime)
  {
    brightness = brightness - fade;
    Serial.println(brightness);
    analogWrite(light, brightness);
    startTime = millis();
  }
  analogWrite(light, 0);
  counter = 0;
}

void loop() {
  // put your main code here, to run repeatedly:

  //lGreen on, slowly dimming
  analogWrite(lGreen, brightness);
  //lRed off, rRed on
  digitalWrite(lRed, LOW);
  digitalWrite(rRed, HIGH);
  Dim (lGreen);

  //rGreen on, slowly dimming
  analogWrite(rGreen, brightness);
  //rRed off, lRed on
  digitalWrite(rRed, LOW);
  digitalWrite(lRed,HIGH);
  Dim(rGreen);
```

Traffic Light Example Extended



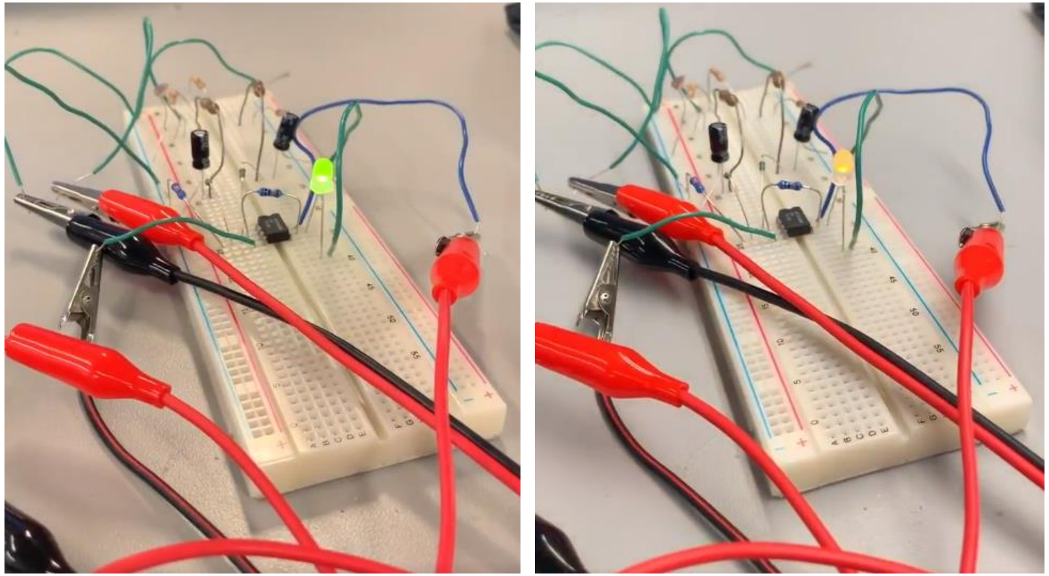
Traffic Light Example



Real Arduino Applications

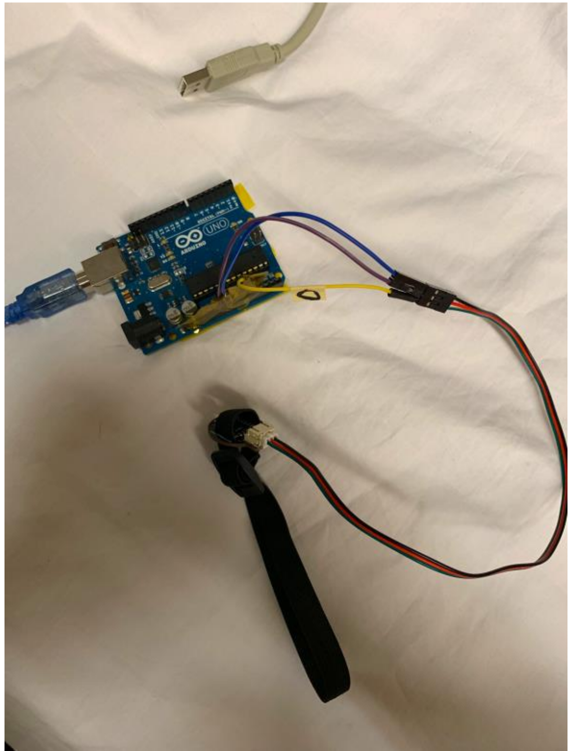
Light-Dependent Circuit

- Used a light-variable resistor to create a circuit that was light dependent
- Zener diode (two-way diode) would shine a different color if it was light or dark



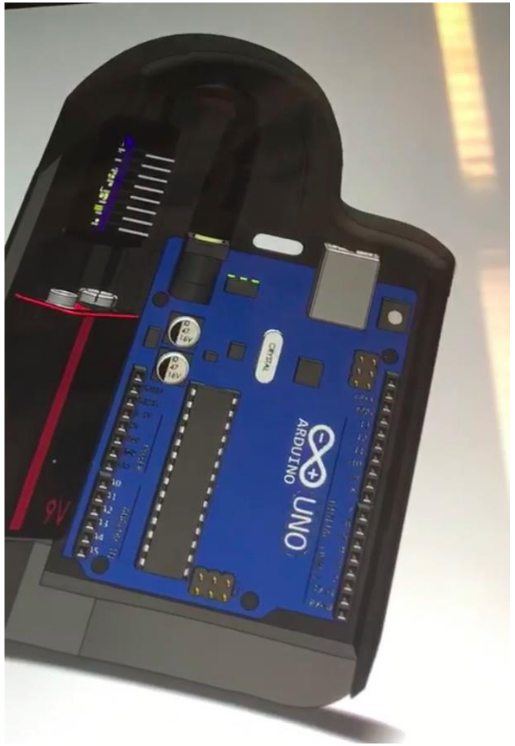
Photophletysmography (PPG) Device

- Simple device to calculate blood oximetry levels and heart rate
- Uses light refraction to calculate amount of oxygen in the blood
- Surprisingly accurate and easy to build (I use it at work as a ground truth for heart rate!)



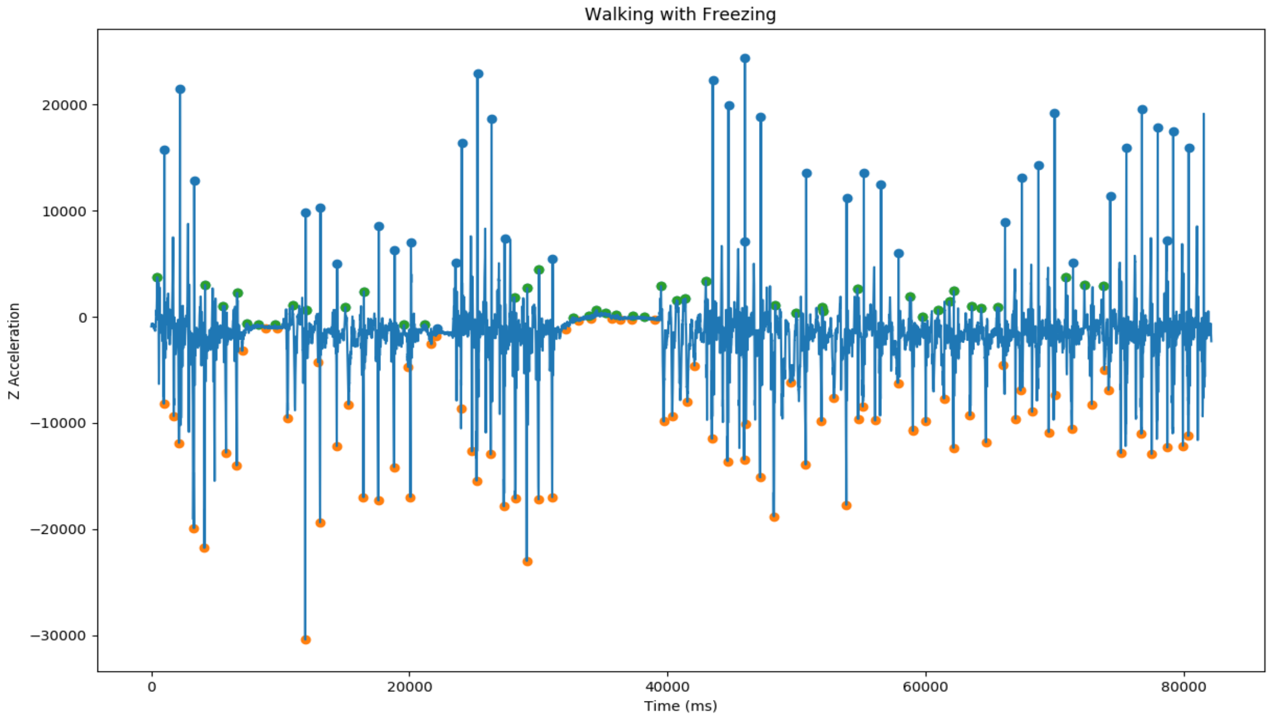
Freezing of Gait Detection System

- Fairly more complex system using an inertial measurement unit (IMU) to capture acceleration in 3 directions
- Using signal processing and machine learning, Arduino was able to detect the differences between a normal walking cycle and a freezing episode. When it detected freezing, it rang a buzzer to help the user get back in step



ST. MARY'S HIGH SCHOOL

Freezing of Gait Detection System


Walking with Freezing
Z Acceleration
Time (ms)