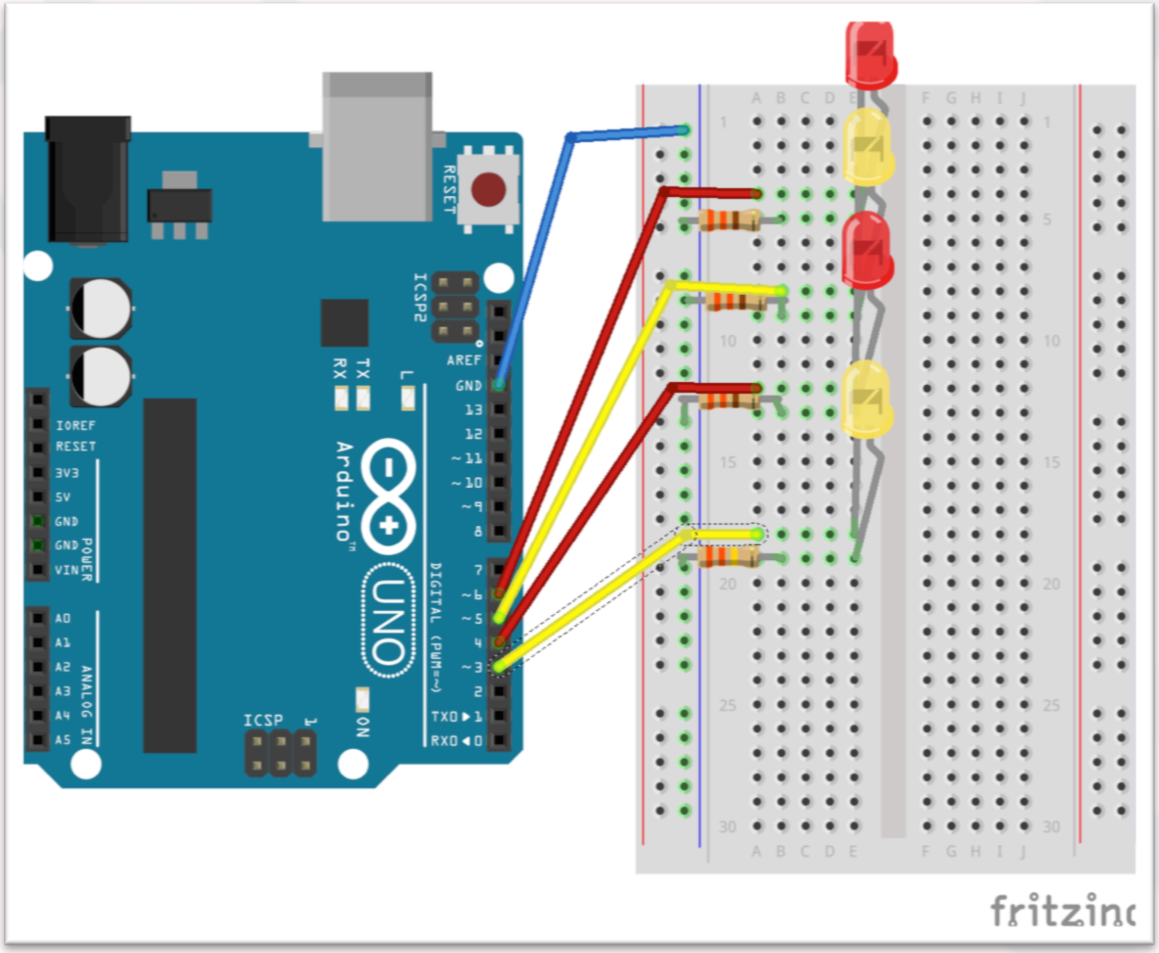
Arduino Part 1- Build Guide

Flashing Lights



The **Flashing Lights** project allows builders to create a prototype that works similarly to traffic lights or car signals

Flashing Lights

Parts List:

2 RED LEDs

2 YELLOW LEDs

1 Long BLUE/GREEN Wire

2 RED Medium Wires

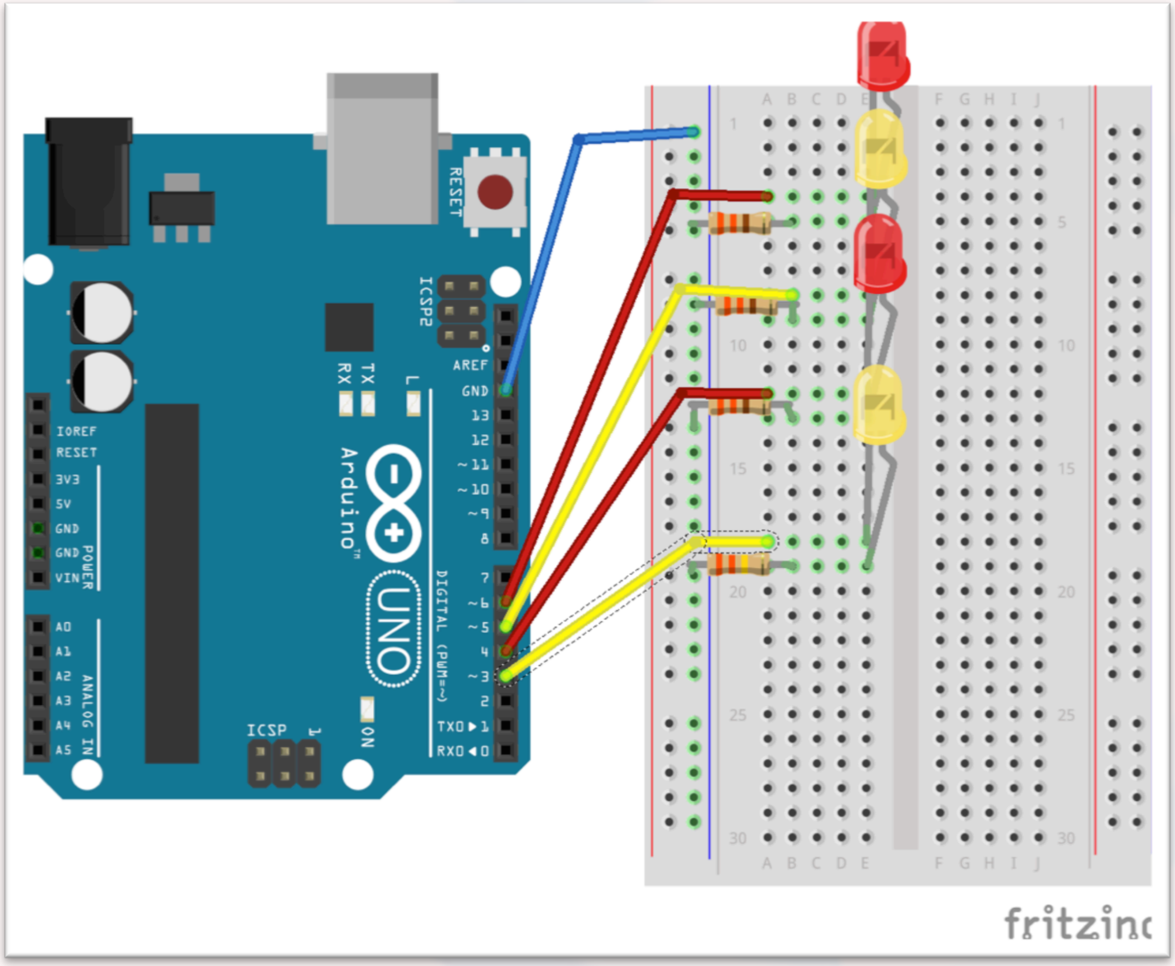
2 YELLOW Medium Wires

4 330 ohm Resistors

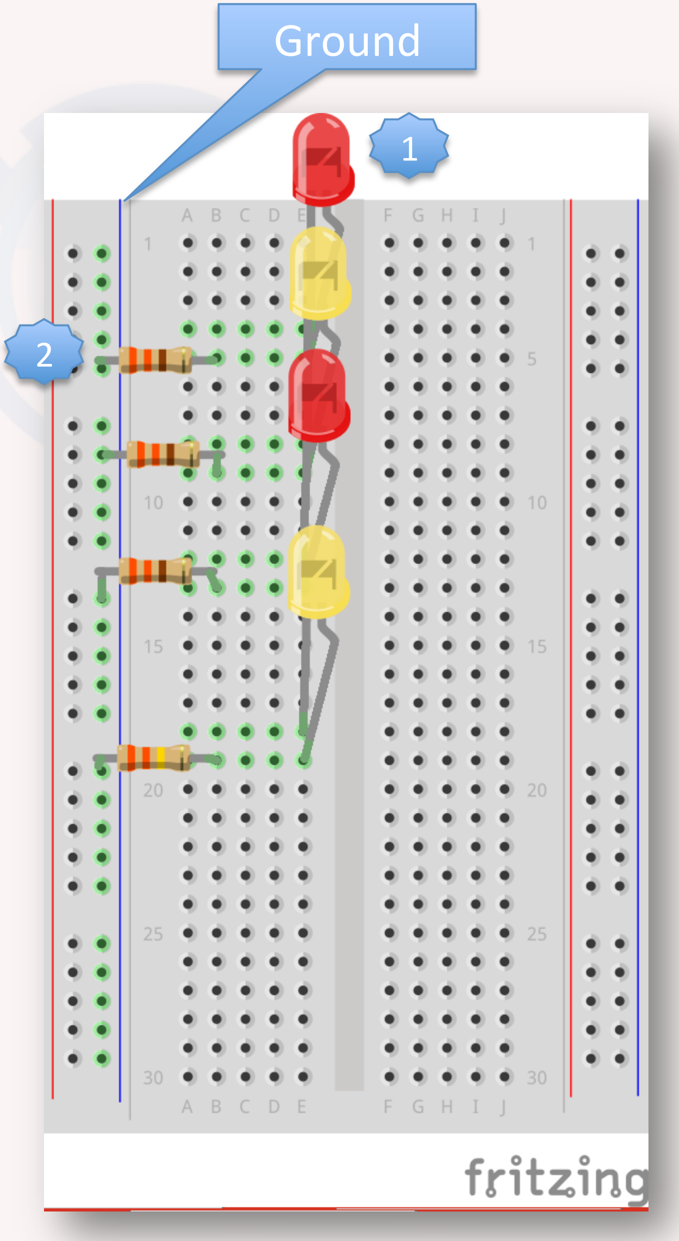
Arduino Board

Breadboard

USB Cable



Flashing Lights:
Attaching Lights and Resistors



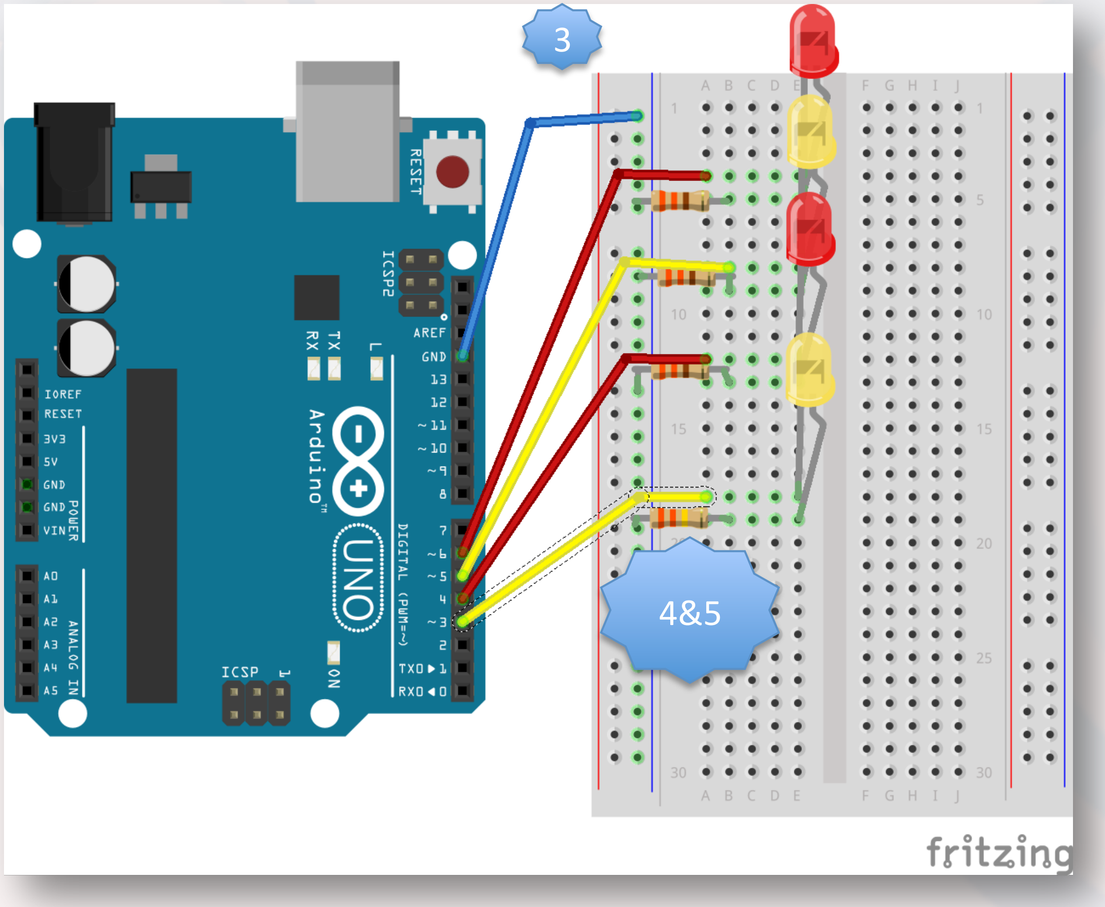
1) Place the LED lights in the following order on the Breadboard

a) Red LED – Long Leg E4, Short Leg E5
b) Yellow LED – Long Leg E8, Short Leg E9
c) Red LED – Long Leg E12, Short Leg E13
d) Yellow LED – Long Leg E18, Short Leg E19

2) Place the resistors in the following order on the Breadboard

a) Left Leg Ground 5, Right Leg A5
b) Left Leg Ground 9, Right Leg A9
c) Left Leg Ground 13, Right Leg A13
d) Left Leg Ground 19, Right Leg A19

Flashing Lights:
Wiring to Arduino



3) Attach a long Blue/Green wire to the GND port on the RIGHT side of the Arduino board. Attach the other end to Ground Row 1 Pin on the breadboard

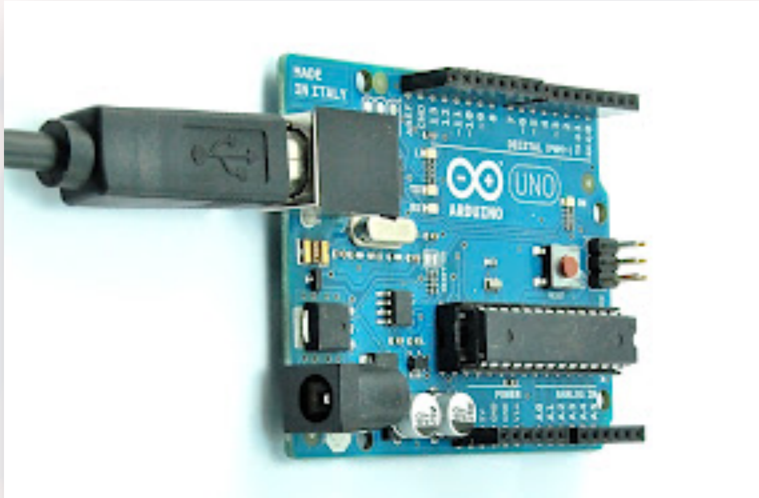
4) Attach RED Wire to Pin 6 on the RIGHT side of the Arduino board. Attach the other end to Column A Row 4 on the breadboard

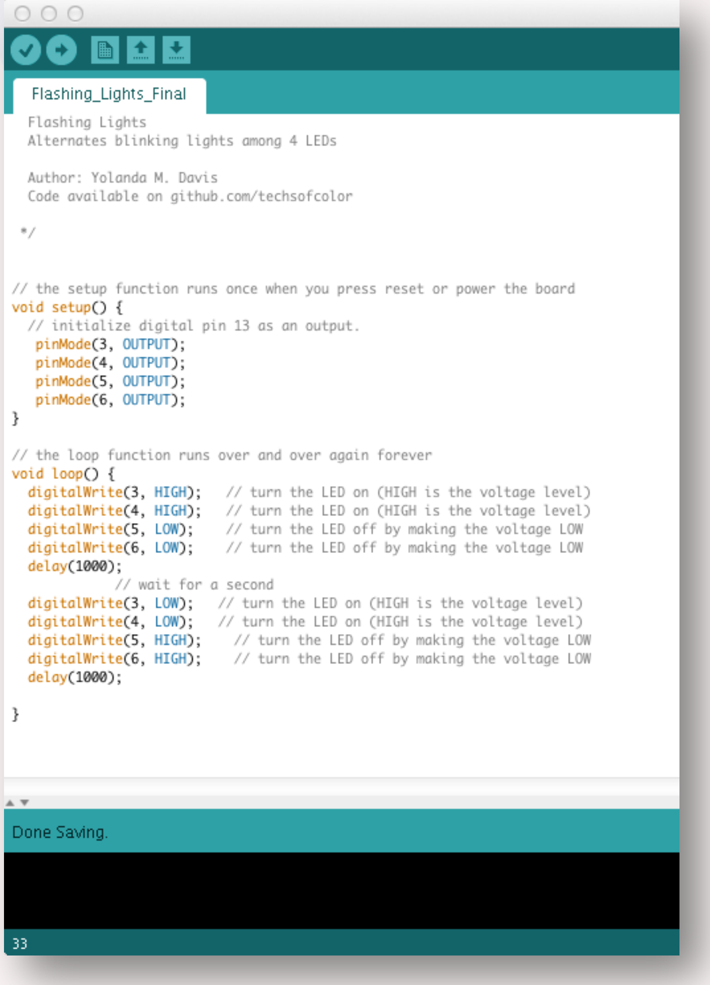
5) Repeat wiring for the other lights:

Yellow Wire: Pin 5 to A8
Red Wire: Pin 4 to A12
Yellow Wire: Pin 3 to A18

Flashing Lights: Installing the Program

1) Connect the USB cord to the Arduino board and Plug cord into laptop





```
  Flashing Lights
  Alternates blinking lights among 4 LEDs

  Author: Yolanda M. Davis
  Code available on github.com/techsofcolor

 */


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
   pinMode(3, OUTPUT);
   pinMode(4, OUTPUT);
   pinMode(5, OUTPUT);
   pinMode(6, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(3, HIGH);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(4, HIGH);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(5, LOW);    // turn the LED off by making the voltage LOW
  digitalWrite(6, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);
             // wait for a second
  digitalWrite(3, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(4, LOW);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(5, HIGH);    // turn the LED off by making the voltage LOW
  digitalWrite(6, HIGH);    // turn the LED off by making the voltage LOW
  delay(1000);

}
```

Done Saving.

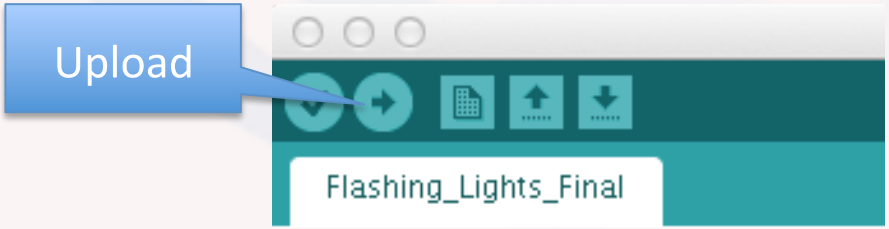
2) Open the Arduino IDE

3) Select “File > Open” and choose the file Flashing_Lights_Final.ino file

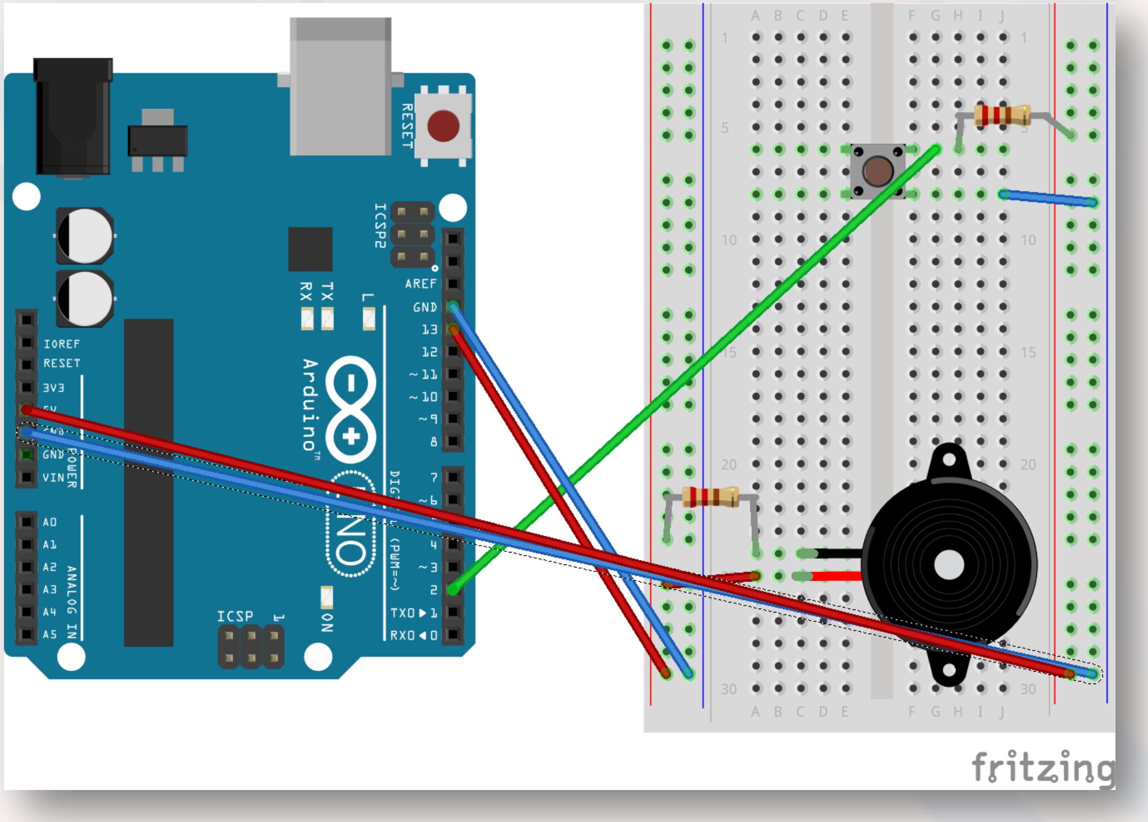
4) Upload the program to Arudino by selecting the Arrow Icon shown below

5) Watch Arduino’s Flashing Lights!

6) Try adjusting the time or changing the order of light flashes



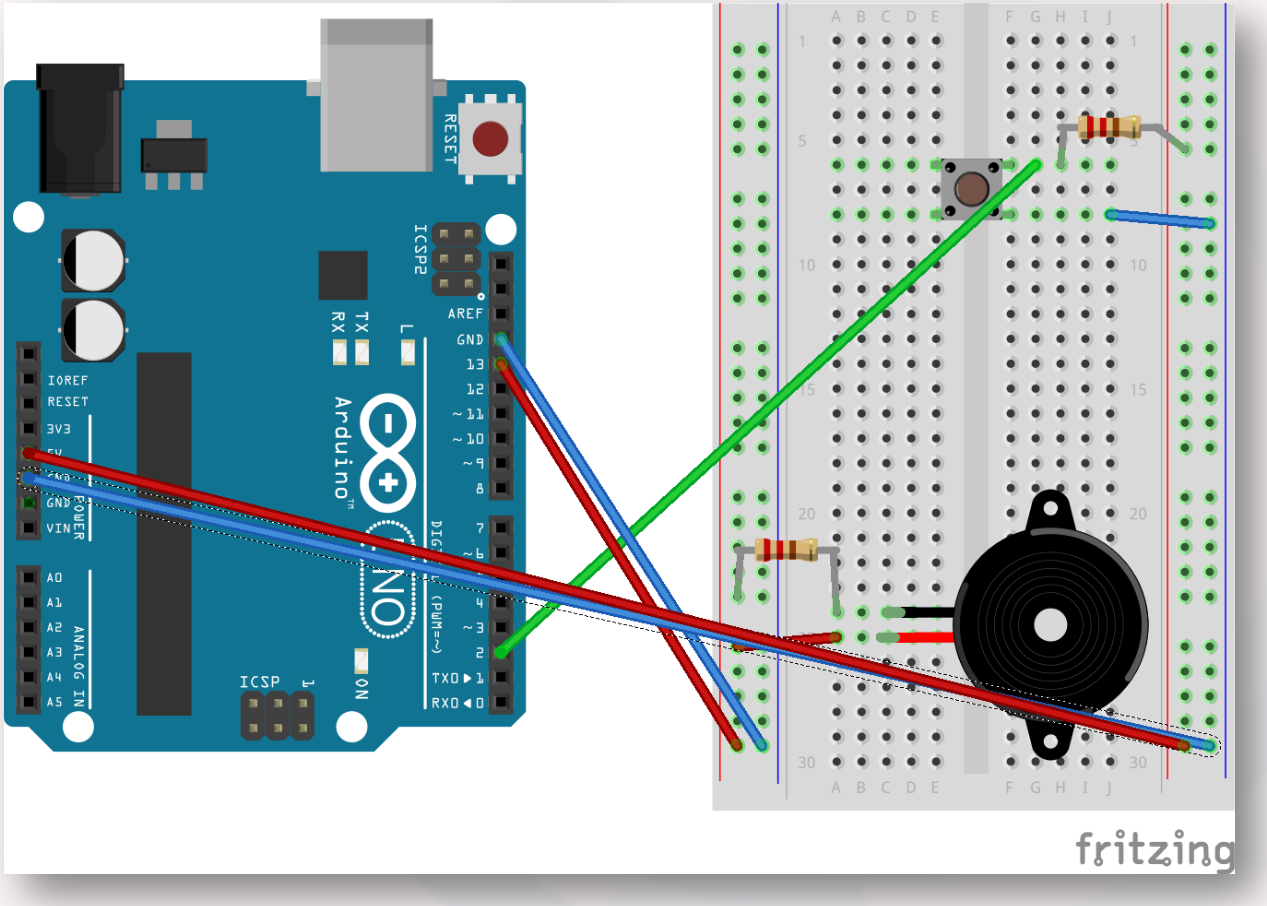
Door Buzzer



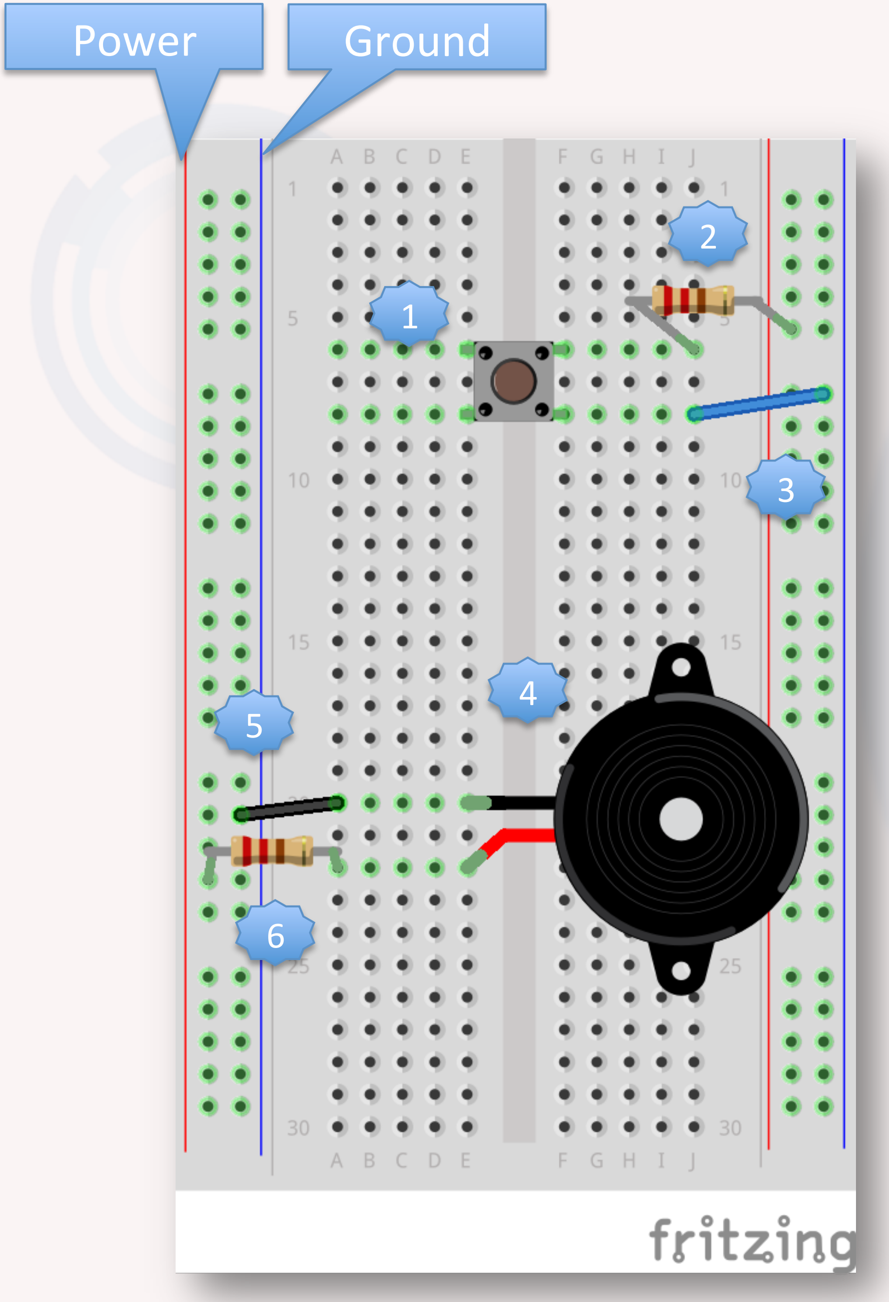
The **Door Buzzer** project allows builders to create a prototype that works similarly to door bells or board game buzzers.

Door Buzzer

Parts List:
1 Button
1 Piezo
2 330 Ohm Resistors
1 Long RED wire
2 Long BLUE/GREEN wire
1 Medium RED wire
1 Medium BLUE/ GREEN
1 Short BLACK wire
1 Short BLUE wire
Arduino Board
Breadboard
USB Cable



Door Buzzer: Attaching Button & Piezo



1) Attach the button in the middle of the board (see left) with the top left corner inserted at Column E row 6

2) Attach the left leg of 1 resistor into Column J Row 6. Attach the right leg to the Right Power Column Row 5.

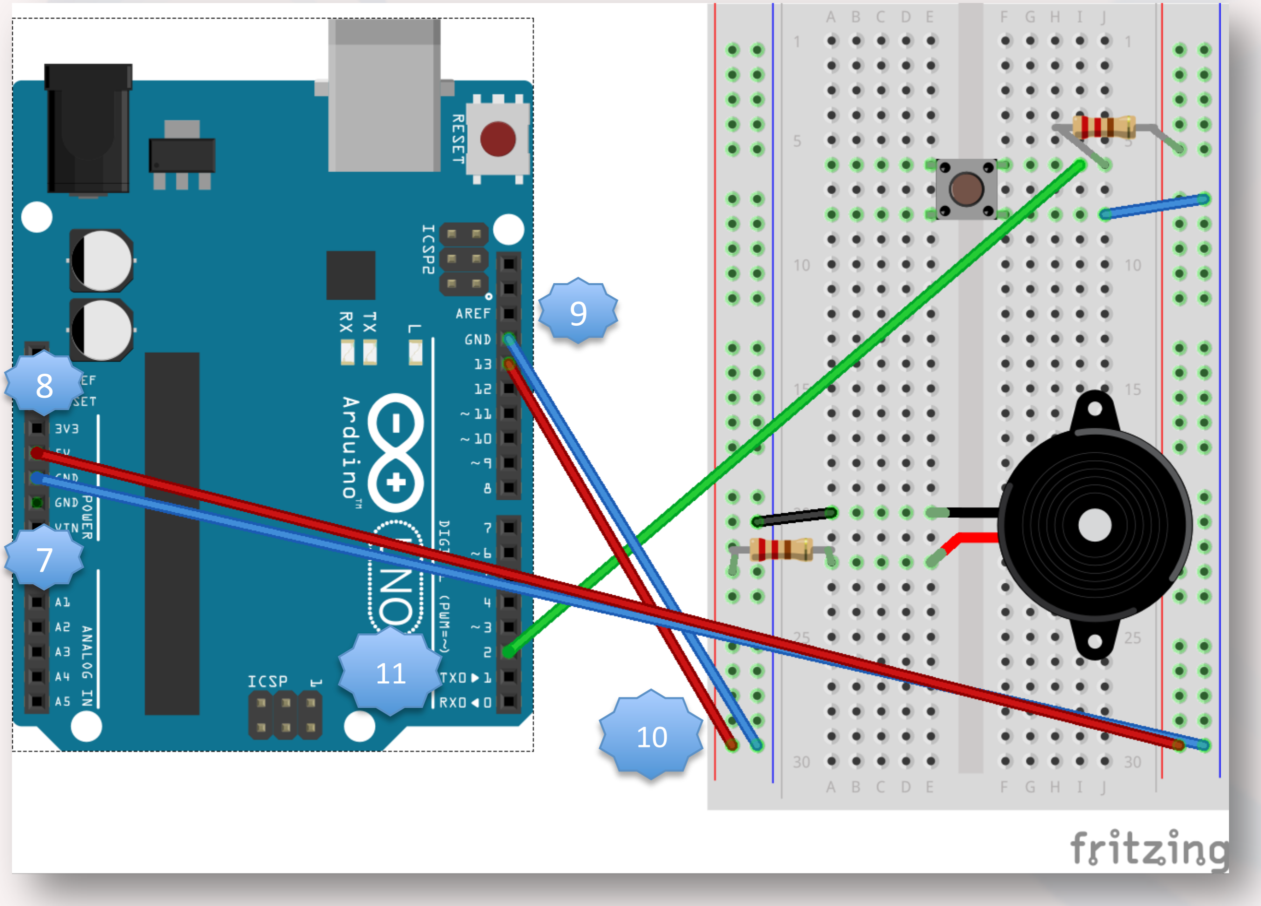
3) Attach the right end of the short BLUE wire to Ground Row 6. Attach the left end to Column J, Row 8.

4) Attach the Piezo on Column E across rows 20 and 23 (this part covers 3 rows).

5) Attach the right end of the short BLACK wire to Column A Row 20. Attach the left end to to the nearest pin the Ground column.

6) Attach the right leg of the second resistor to the Column A Row 23. Attach the left leg to the nearest pin in the Power column.

Door Buzzer: Attaching Wires

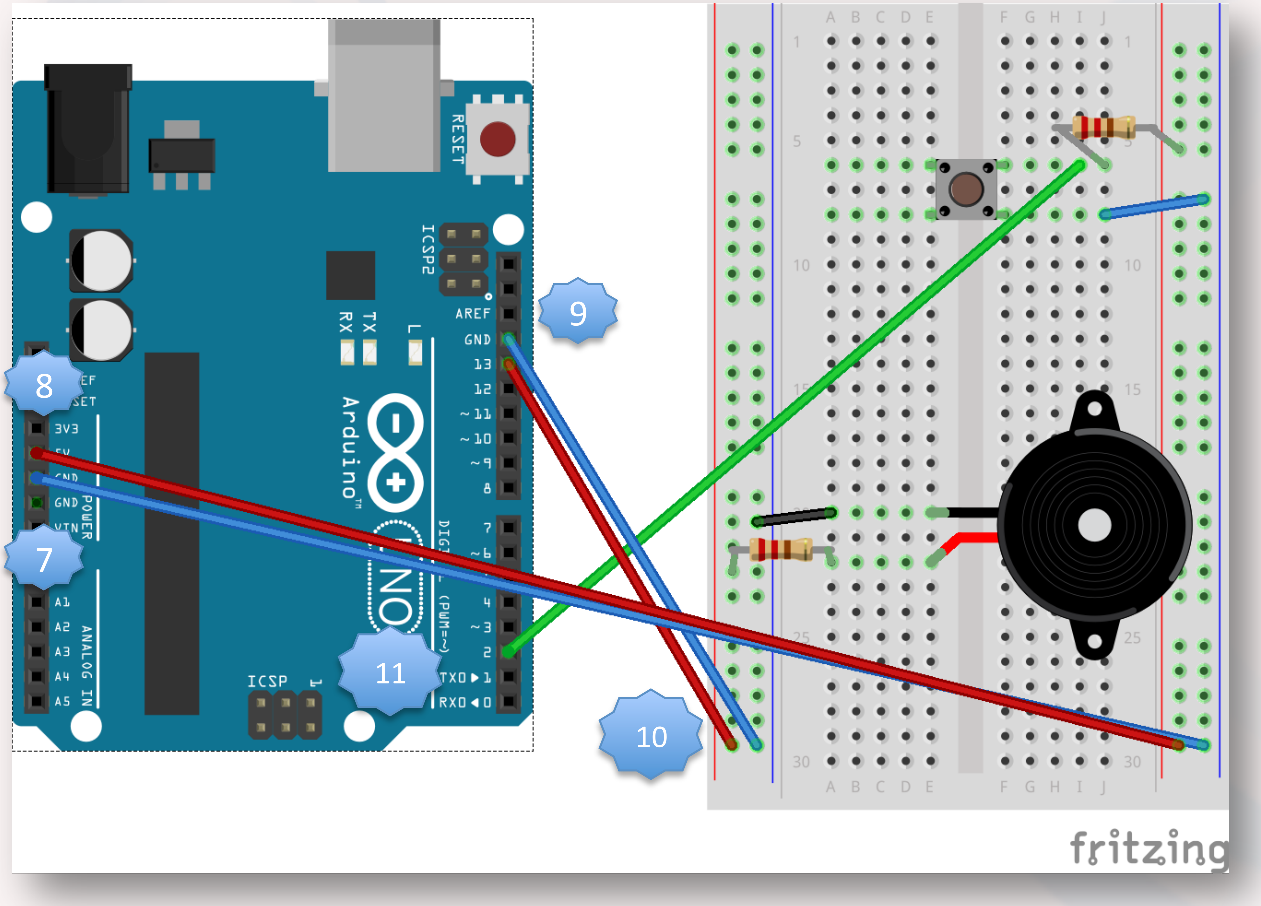


7) Attach a long BLUE/GREEN wire to the GND pin on the LEFT side of the Arduino board. Attach the other end to the RIGHT Ground Column, Row 30 Pin on the breadboard.

8)Attach a long RED wire to the 5V pin on the LEFT side of the Arduino board. Attach the other end to the RIGHT Power Column, Row 30 Pin on the breadboard

9) Attach a medium BLUE/GREEN wire to the GND pin on the RIGHT side of the Arduino board. Attach the other end to the LEFT Ground Column, Row 30 pin on the breadboard

Door Buzzer: Attaching Wires

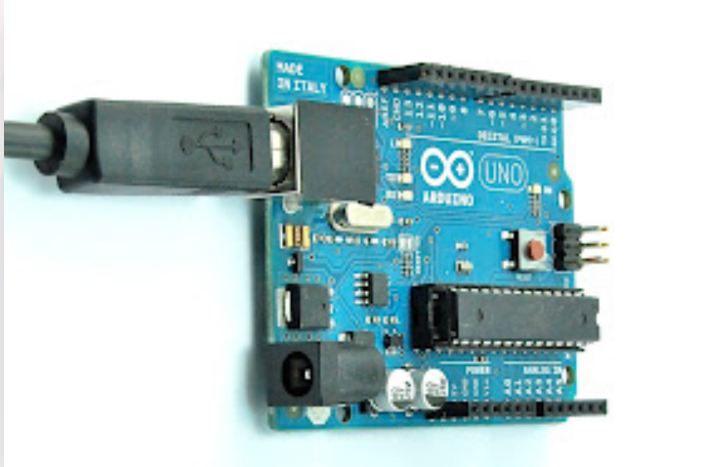


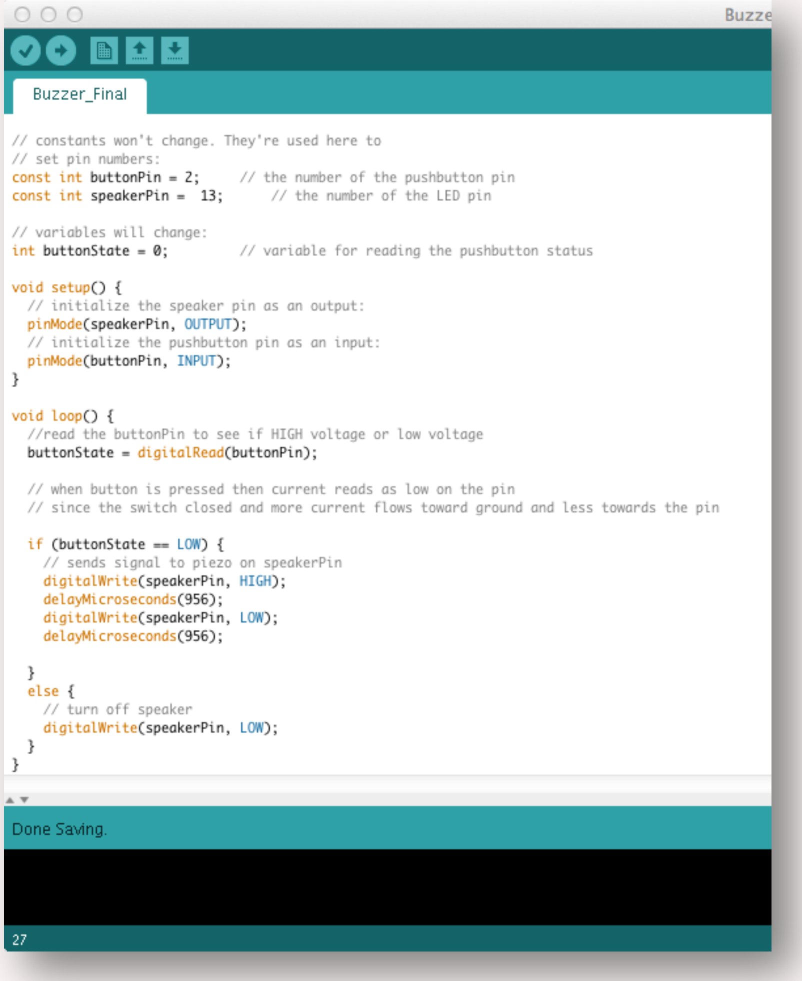
10) Attach a medium RED wire to pin 13 pin on the <u>RIGHT</u> side of the Arduino board. Attach the other end to the LEFT Power Column, Row 30 pin on the breadboard

11) Attach a long GREEN wire to pin 2 on the <u>RIGHT</u> side of the Arduino board. Attach the other end to Column I Row 6 on the breadboard.

Door Buzzer: Installing the Program

1) Connect the USB cord to the Arduino board and Plug cord into laptop





```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int speakerPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the speaker pin as an output:
  pinMode(speakerPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  //read the buttonPin to see if HIGH voltage or low voltage
  buttonState = digitalRead(buttonPin);

  // when button is pressed then current reads as low on the pin
  // since the switch closed and more current flows toward ground and less towards the pin

  if (buttonState == LOW) {
    // sends signal to piezo on speakerPin
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(956);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(956);

  }
  else {
    // turn off speaker
    digitalWrite(speakerPin, LOW);
  }
}
```

Done Saving.

2) Open the Arduino IDE

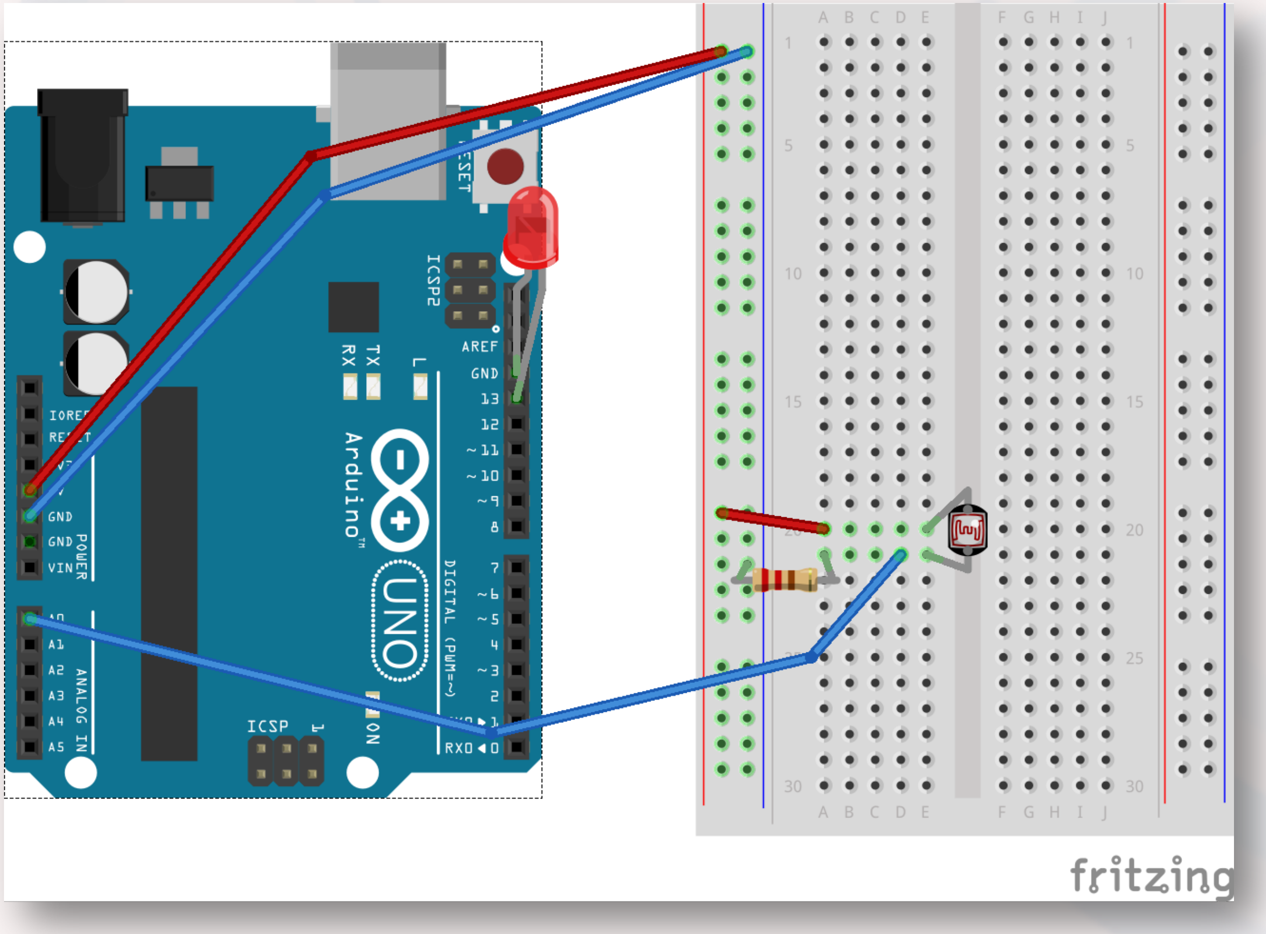
3) Select “File > Open” and choose the file Buzzer_Final.ino file

4) Upload the program to Arudino by selecting the Arrow Icon

5) Test out the Buzzer by pressing the button on on the breadboard

6) Try adding an LED that will light along with the Buzzer sound when the button is pressed

Night Light



The **Night Light** project allows builders to create a prototype that works similarly to night lights or outdoor lawn lights.

Night Light

Parts List:

1 RED LED

2 Long BLUE/GREEN Wires

1 Long RED Wire

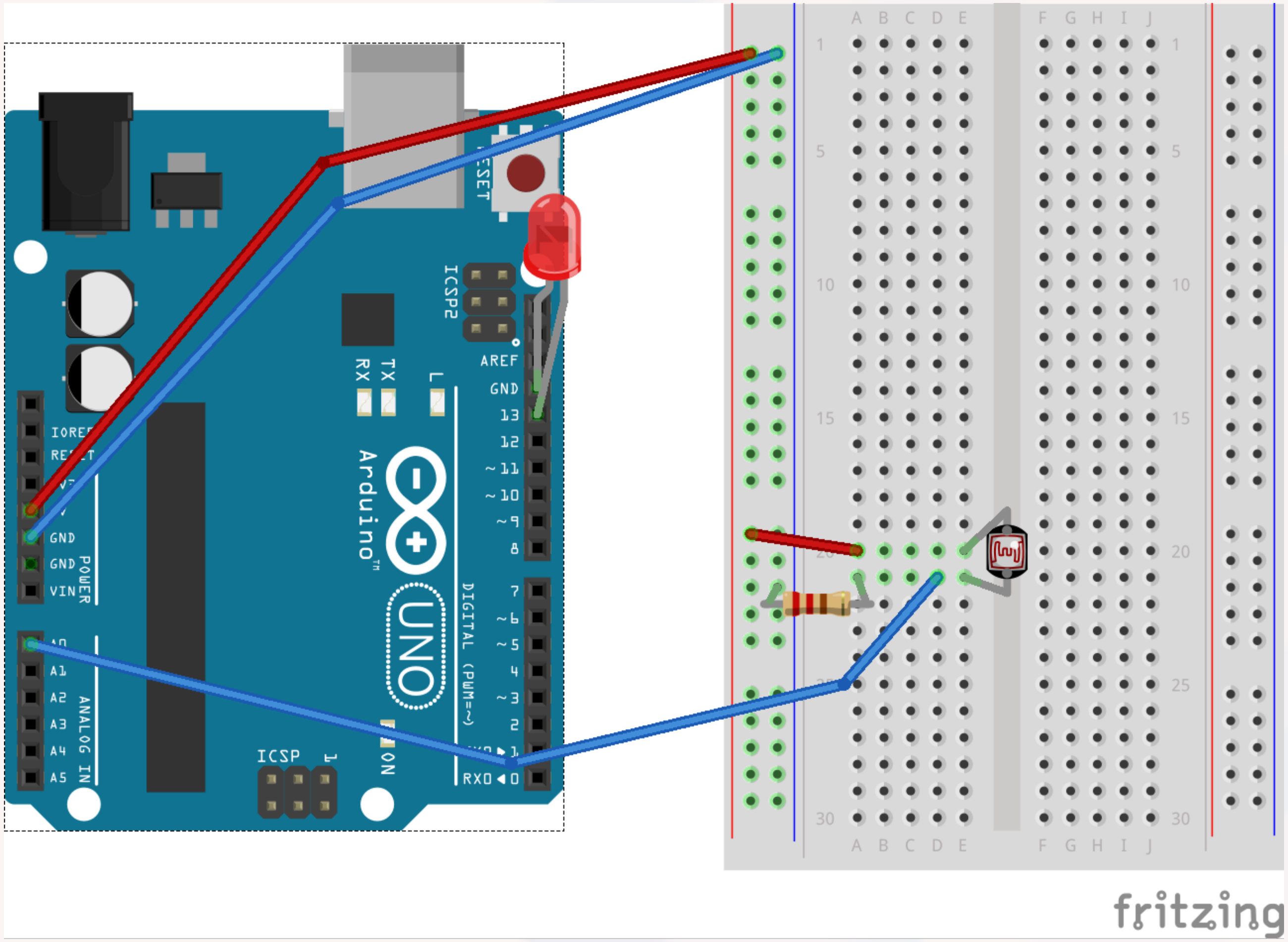
1 Short RED Wire

1 330 ohm Resistor

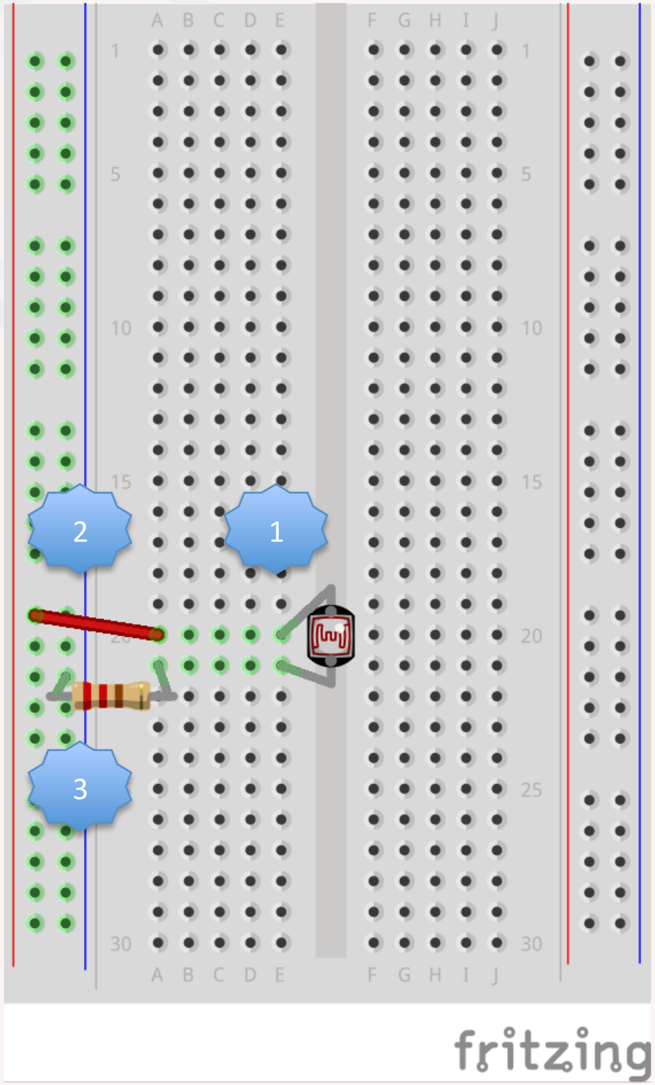
Arduino Board

Breadboard

USB Cable

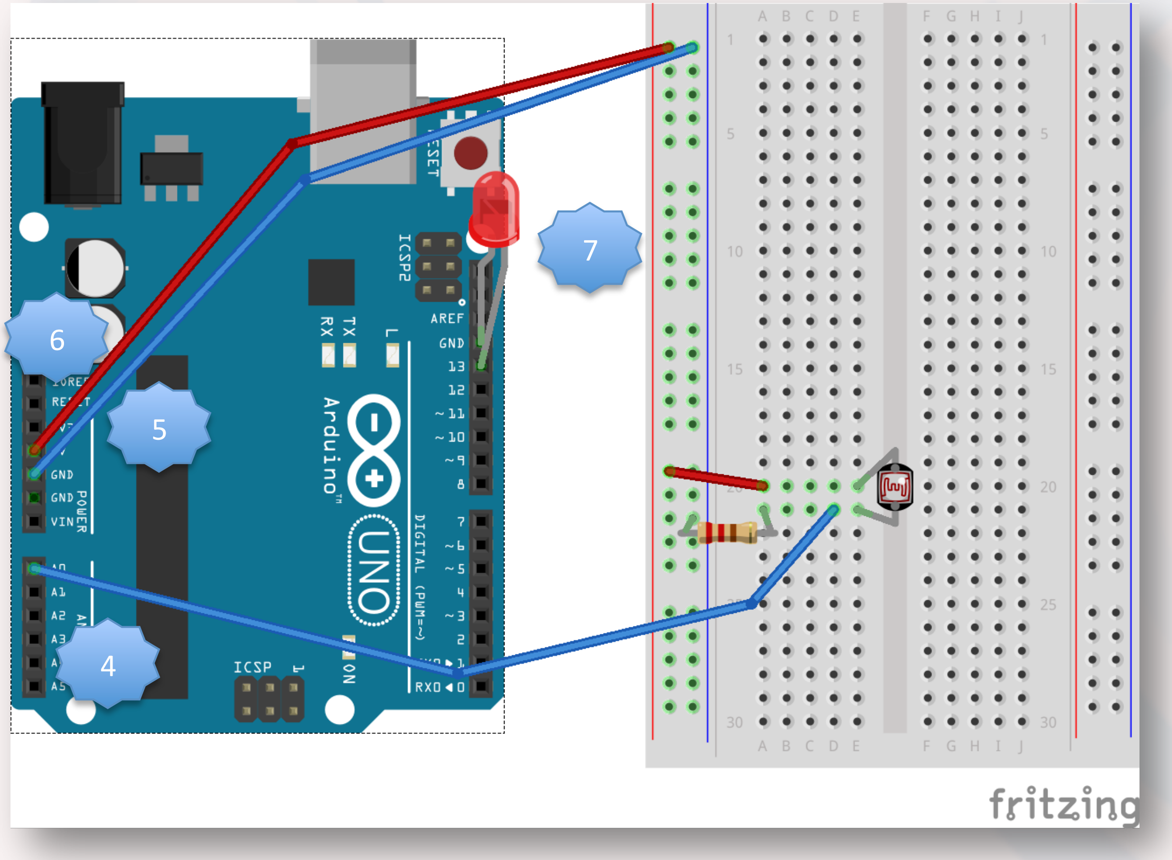


Night Light



1) Attach the Photocell to Column E, across rows 20 and 21.
2) Attach the right end of the short RED wire to Column A Row 20 pin. Attach the left end to the nearest Power pin.
3) Attach the right end of the resistor to Column A Row 21. Attach the left side to the nearest Ground pin

Night Light



4) Attach 1 BLUE/GREEN wire to the A0 (Analog) pin on the left side of the Arduino board. Attach the other end in front of the photocell on Column D Row 21

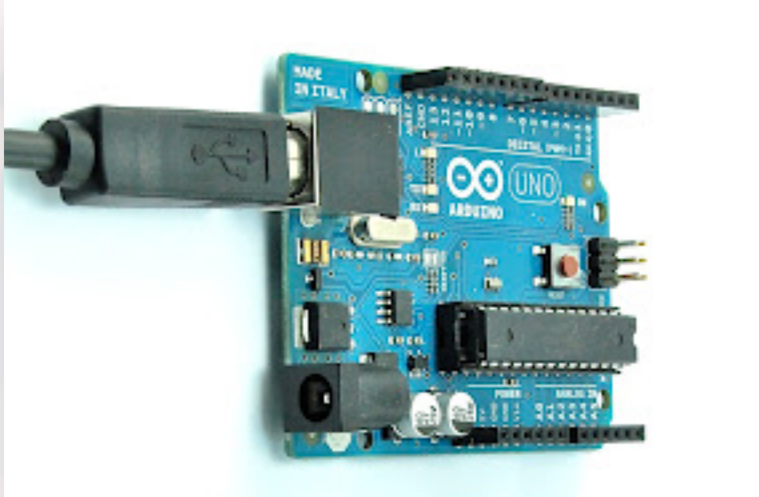
5) Attach the second BLUE/GREEN wire to the GND pin on the left side of the Arduino board. Attach the other end to the left Ground Column Row 1 of the breadboard

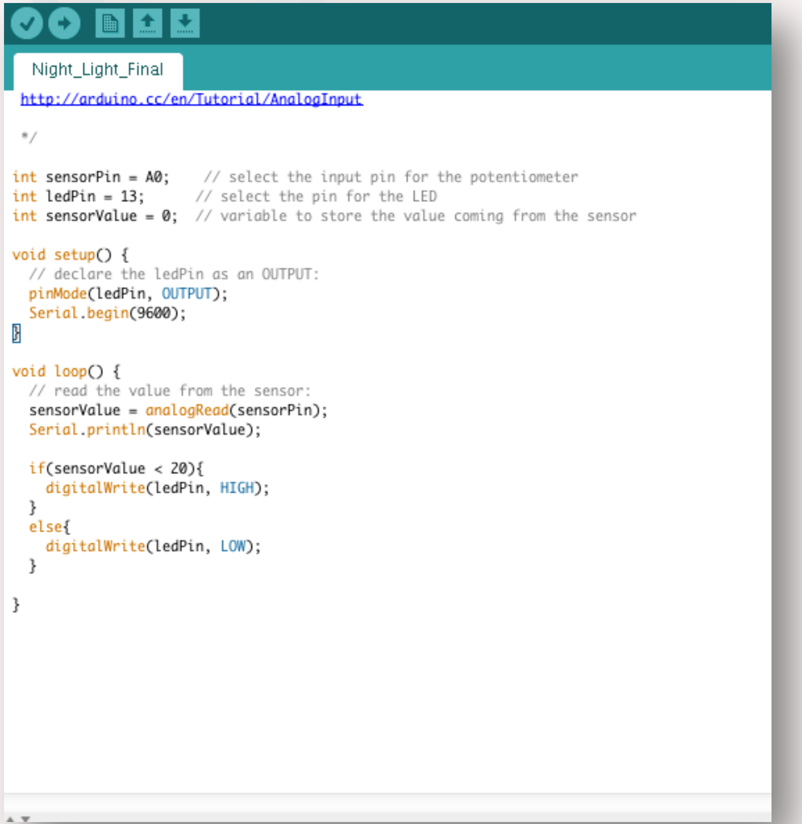
6) Attach the RED wire to the 5V pin on the left side of the Arduino board. Attach the right side to the LEFT Power Column Row 1 of the breadboard.

7) Attach the RED light directly to the Arduino board with the long leg attached to pin 13 and the short leg attached to GND pin

Night Light: Installing the Program

1) Connect the USB cord to the Arduino board and Plug cord into laptop





```
http://arduino.cc/en/Tutorial/AnalogInput

 */

int sensorPin = A0;    // select the input pin for the potentiometer
int ledPin = 13;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue);

  if(sensorValue < 20){
    digitalWrite(ledPin, HIGH);
  }
  else{
    digitalWrite(ledPin, LOW);
  }

}
```

2) Open the Arduino IDE

3) Select “File > Open” and choose the file Night_Light_Final.ino file

4) Upload the program to Arudino by selecting the Arrow Icon

5) Test out the Night Light on Arduino by turning off and on room light or covering the photo cell.