

VIT-TBI ILD WORKSHOP 2024

Documentation Report

Name: Srija Koppar

Reg. No.: 23BCT0057

Introduction to Arduino: Creating Simulations

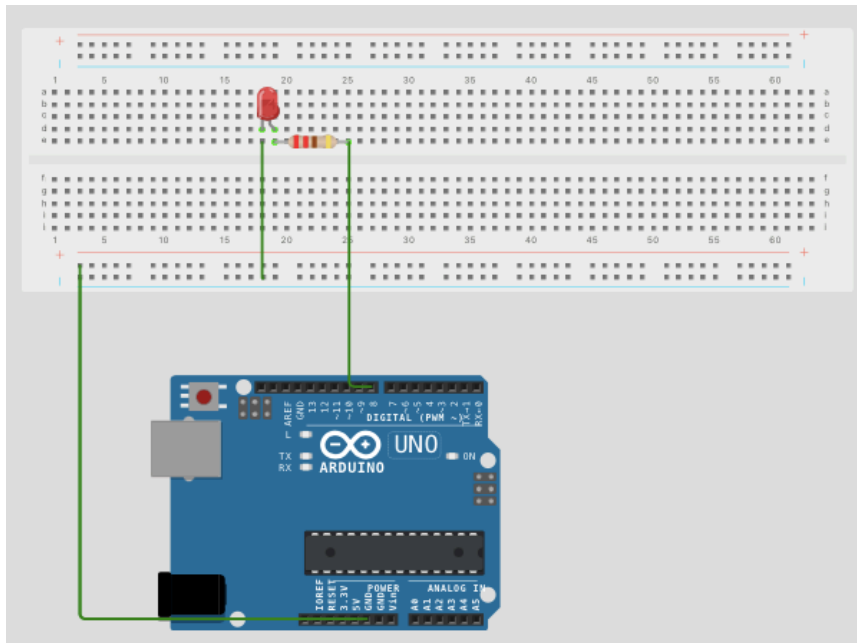
The following projects focus on basic Arduino applications and their implementation using online simulation via tools such as Wokwi. The main aim of the project is to understand how to code with Arduino Uno and how to make connections for various applications.

Project 1: LED Blinking [Simulation]

Aim: This project aims to make an LED bulb blink by connecting it to Arduino Uno.

Components Required: LED bulb, 220 Ohm resistor, Arduino Uno board, breadboard, 3 wires.

Circuit diagram:



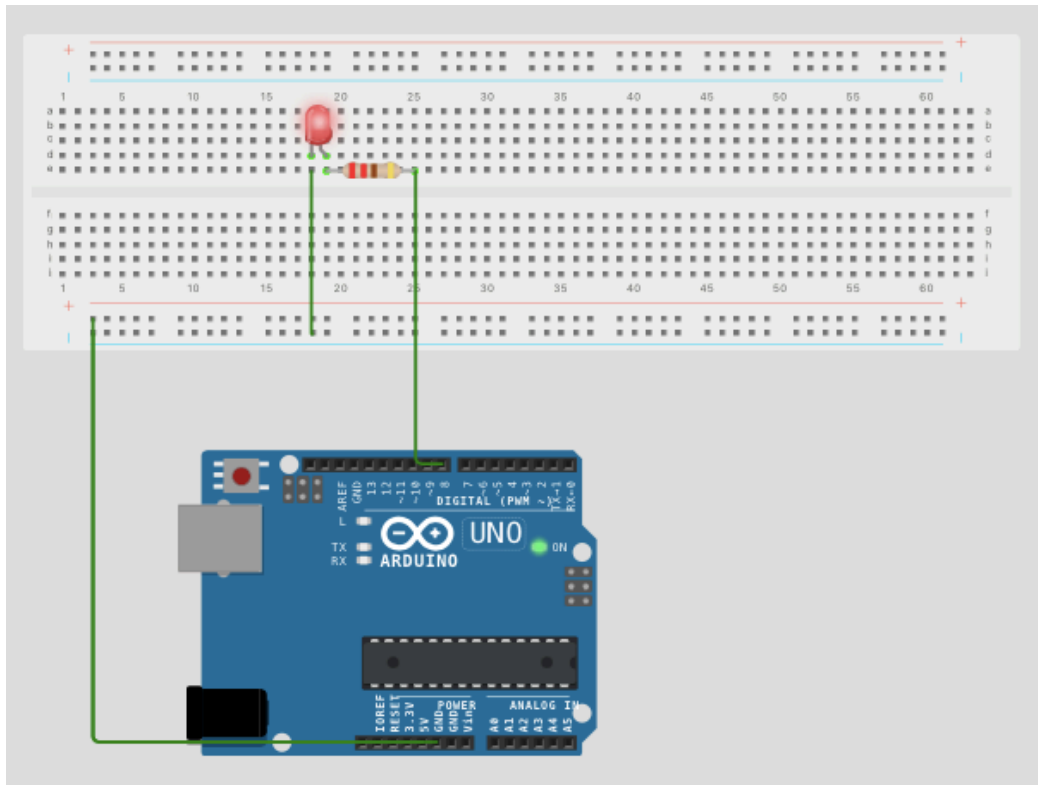
Code:

```
int redLed=8;
void setup() {
  // put your setup code here, to run once:
  pinMode(redLed,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(redLed,HIGH);
  delay(9000);
  digitalWrite(redLed,LOW);
}
```

```
delay(1000);  
}
```

Output:

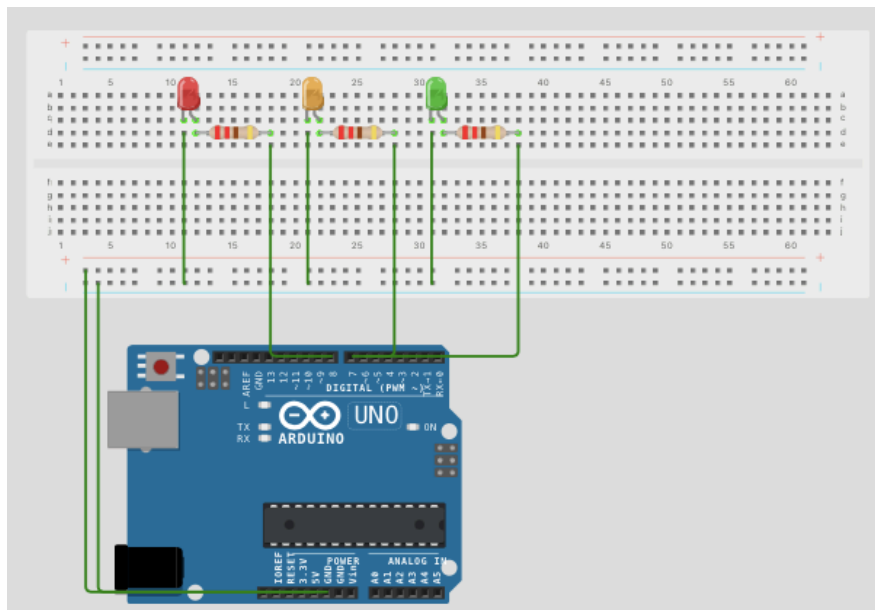


Project 2: Traffic Light [Simulation]

Aim: This project aims to create a traffic light simulation by connecting three LEDs- red, orange and green in series to Arduino Uno.

Components Required: 3 LED bulbs- red, orange and green colour, 3 220 Ohm resistors, Arduino Uno board, breadboard, 8 wires.

Circuit diagram:



Code:

```
int redLed=8;
int orangeLed=7;
int greenLed=6;
void setup() {
  // put your setup code here, to run once:
  pinMode(redLed,OUTPUT);
  pinMode(orangeLed,OUTPUT);
  pinMode(greenLed,OUTPUT);
}

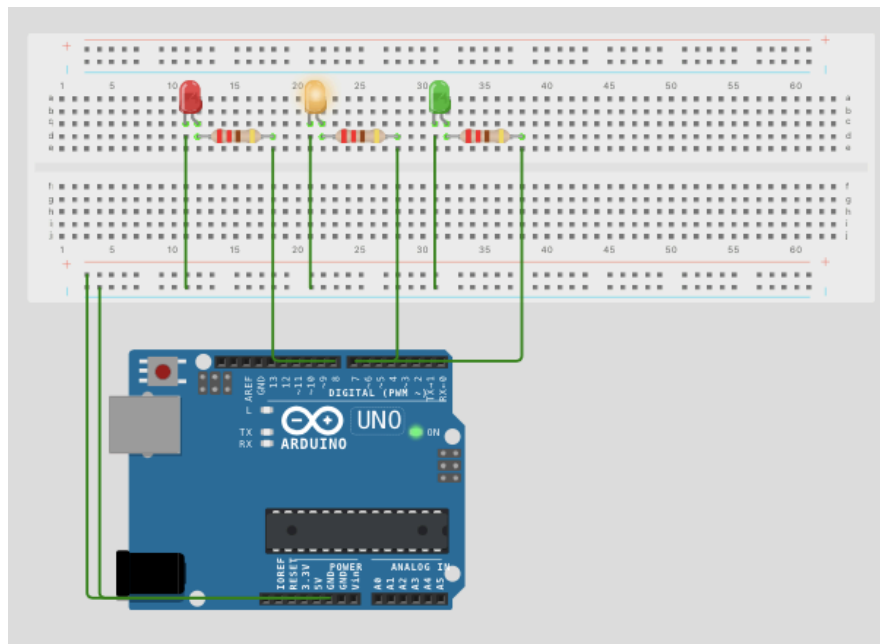
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(redLed,HIGH);
  delay(5000);
  digitalWrite(redLed,LOW);
  delay(1000);
```

```

digitalWrite(orangeLed,HIGH);
delay(5000);
digitalWrite(orangeLed,LOW);
delay(1000);
digitalWrite(greenLed,HIGH);
delay(5000);
digitalWrite(greenLed,LOW);
delay(1000);
}

```

Output:

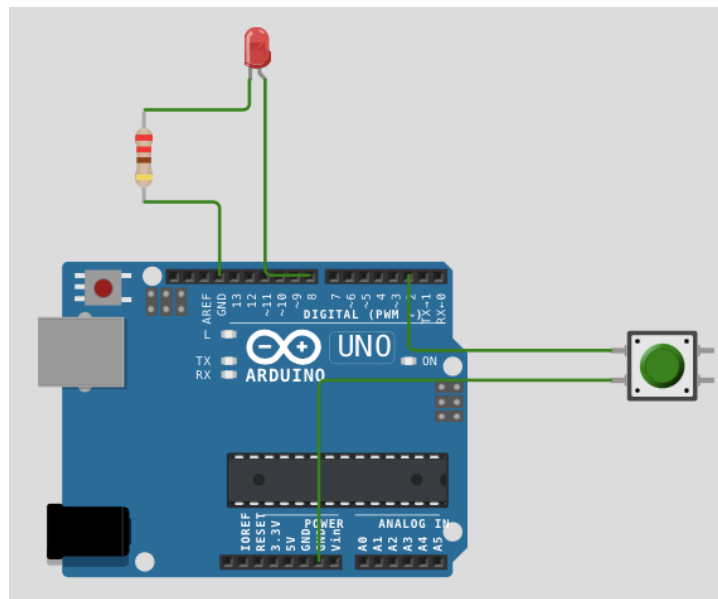


Project 3: Button Controlled LED [Simulation]

Aim: This project aims to control the blinking of an LED bulb using a pushbutton by connecting them to Arduino Uno.

Components Required: LED bulb, 220 Ohm resistors, pushbutton, Arduino Uno board, breadboard, 5 wires.

Circuit diagram:



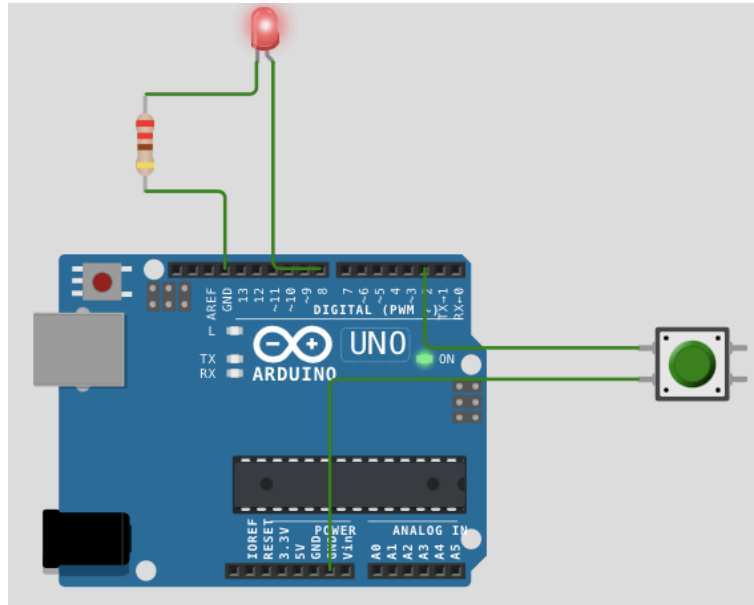
Code:

```
int ledPin=8;
int buttonPin=2;
bool buttonState = false;
void setup() {
  // put your setup code here, to run once:
  pinMode(ledPin,OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  buttonState=digitalRead(buttonPin);
  if(buttonState==HIGH)
  {
    digitalWrite(ledPin,HIGH);
  }
}
```

```
else
{
  digitalWrite(ledPin,LOW);
}
}
```

Output:

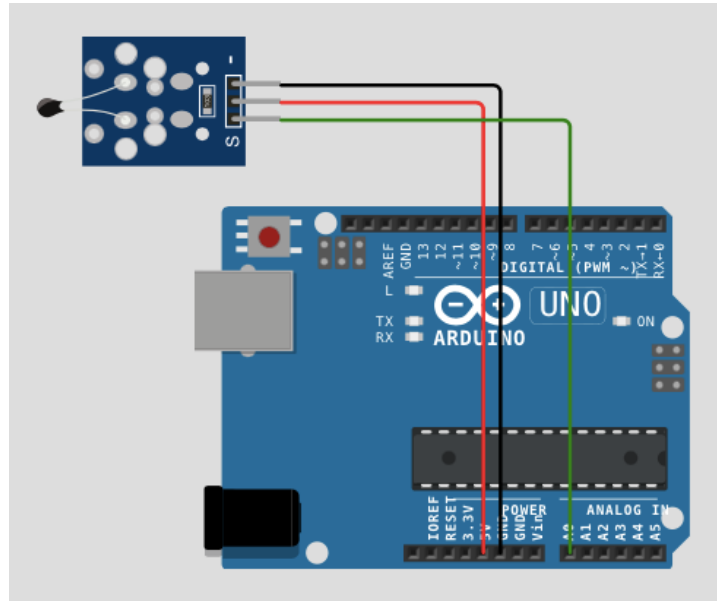


Project 4: Temperature Sensor [Simulation]

Aim: This project aims to set up a temperature sensor by connecting it to Arduino Uno.

Components Required: Analog temperature sensor, Arduino Uno board, 3 wires.

Circuit diagram:



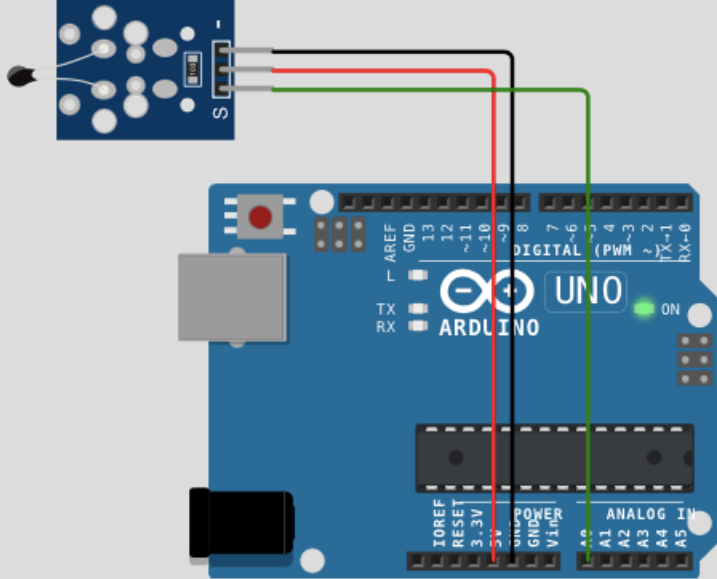
Code:

```
const int beta = 3950;




void setup() {
    Serial.begin(9600);
}

void loop() {
    int value = analogRead(A0);
    float celsius = 1 / (log(1 / (1023. / value - 1)) / beta + 1.0 / 298.15) - 273.15;
    Serial.print("Temp: ");
    Serial.print(celsius);
    Serial.print(" C\n");
    delay(1000);
}
```


Output:



Temp: 23.99 C
Temp: 23.99 C
Temp: 23.99 C
Temp: 23.99 C
Temp: 23.99 C
Temp: 23.99 C
Temp: 23.99 C

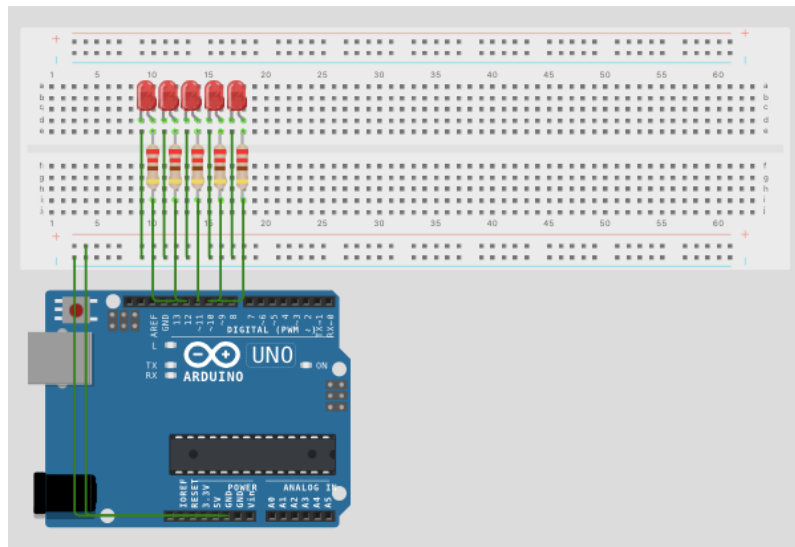


Project 5: Consecutive LED blinking [Simulation]

Aim: This project aims to make LED connected in a row blink consecutively by connecting them to Arduino Uno.

Components Required: 5 LED bulbs, 5 220 Ohm resistors, Arduino Uno board, breadboard, 12 wires.

Circuit diagram:



Code:

```
const int beta = 3950;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
    int value = analogRead(A0);
```

```
    float celsius = 1 / (log(1 / (1023. / value - 1)) / beta + 1.0 / 298.15) - 273.15;
```

```
    Serial.print("Temp: ");
```

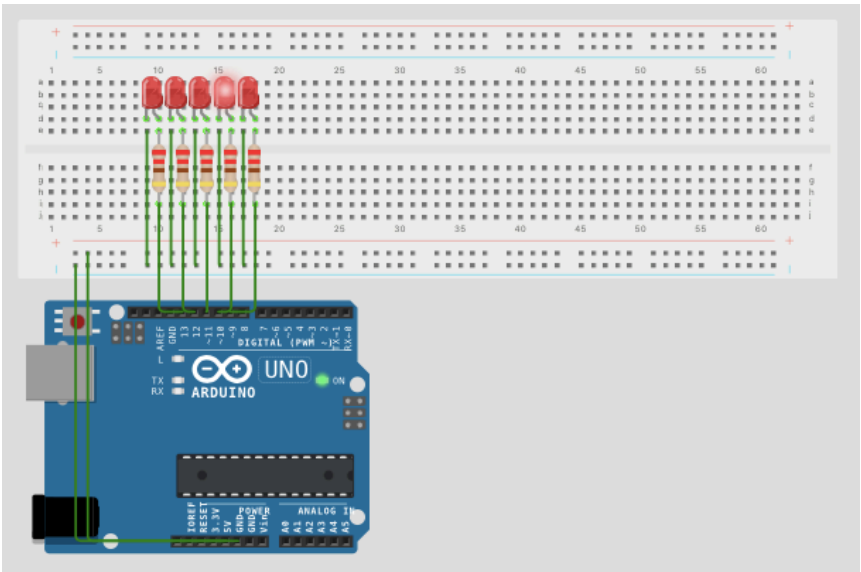
```
    Serial.print(celsius);
```

```
    Serial.print(" C\n");
```

```
    delay(1000);
```

```
}
```

Output:

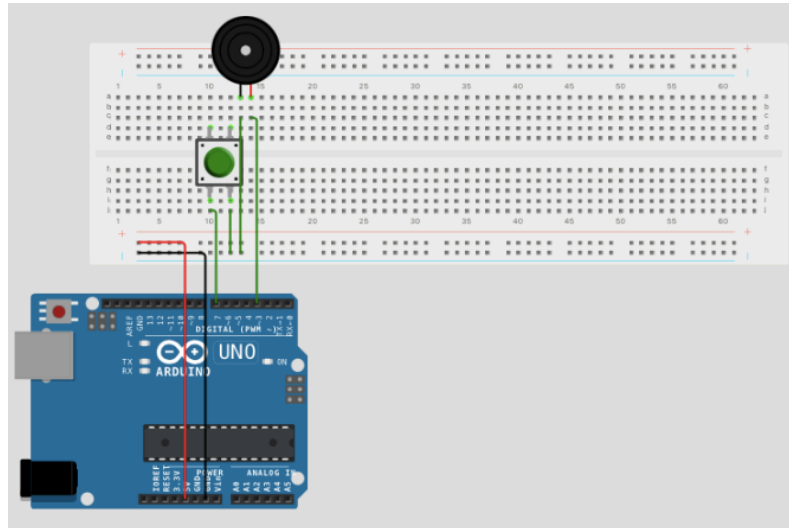


Project 6: Button- Controlled Buzzer Alarm [Simulation]

Aim: This project aims to make a buzzer alarm beep by controlling it using a pushbutton by connecting them to Arduino Uno.

Components Required: Buzzer alarm, pushbutton, Arduino Uno board, breadboard, 6 wires.

Circuit diagram:

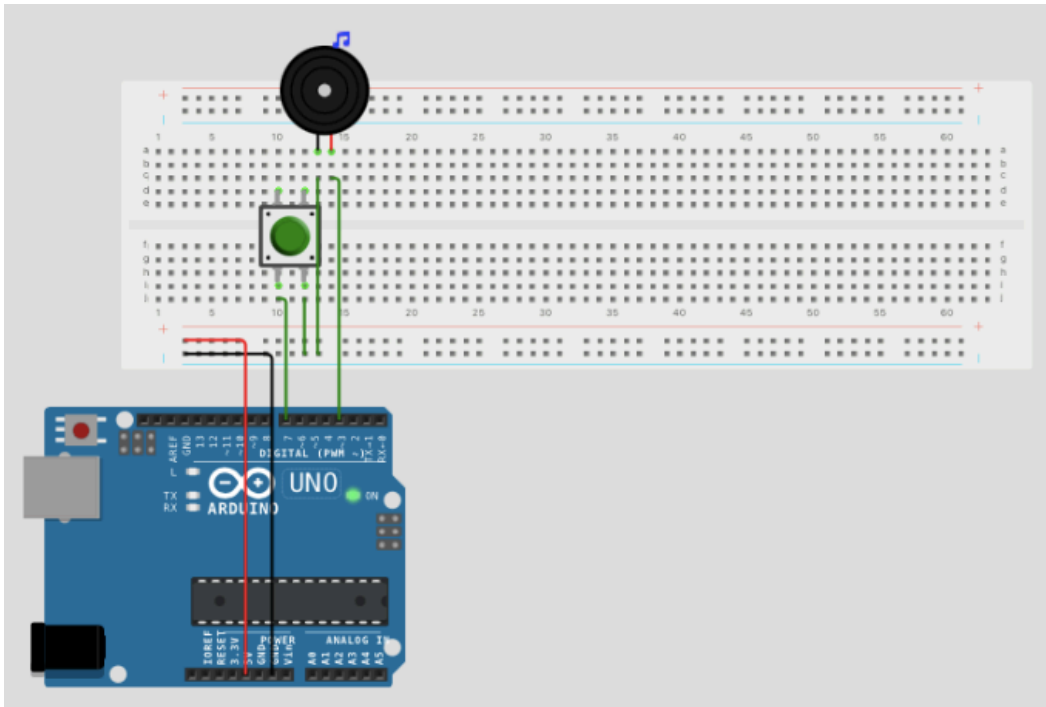


Code:

```
void setup()
{
  pinMode(7, INPUT);
  pinMode(3, OUTPUT);
}

void loop()
{
  if(digitalRead(7)==HIGH)
  {
    tone(3, 15000);
  }
  else noTone(3);
}
```

Output:

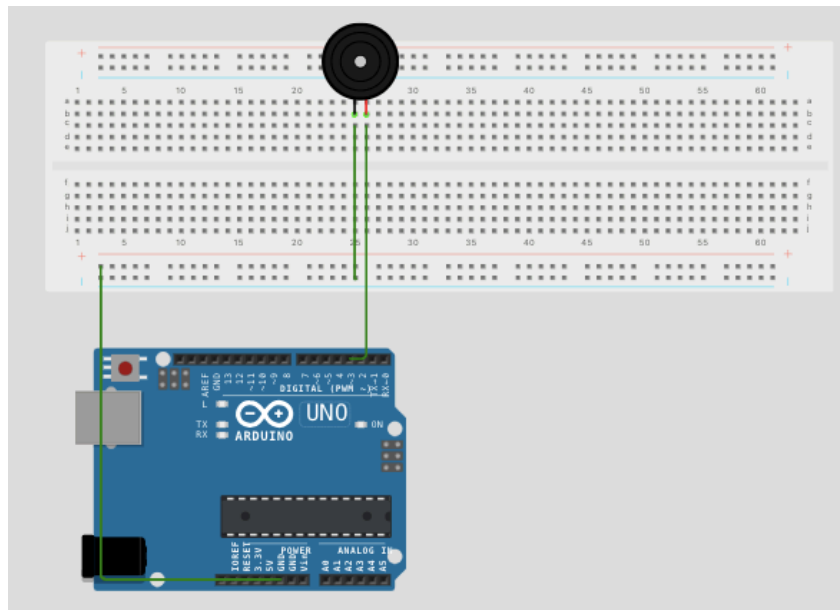


Project 7: Playing Happy Birthday Tune using Buzzer Alarm [Simulation]

Aim: This project aims to play “Happy Birthday” tune on buzzer alarm by connecting it to Arduino Uno.

Components Required: Buzzer alarm, Arduino Uno board, breadboard, 2 wires.

Circuit diagram:



Code:

```
int buzzerPin=3;
int length = 28; // the number of notes
char notes[] = "GGAGcB GGAGdc GGxecBA yyecdc";
int beats[] = {2,2,8,8,8,16,1,2,2,8,8,8,16,1,2,2,8,8,8,16,1,2,2,8,8,8,16};
int tempo = 200; // time delay between notes
```

```
void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2)
  {
    digitalWrite(buzzerPin,HIGH);
    delayMicroseconds(tone);
    digitalWrite(buzzerPin, LOW);
    delayMicroseconds(tone);
  }
}
```

```
void playNote(char note, int duration) {
  char names[] = {'C', 'D', 'E', 'F', 'G', 'A', 'B',
                  'c', 'd', 'e', 'f', 'g', 'a', 'b',
```

```

        'x', 'y' };

int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014,
               956, 834, 765, 593, 468, 346, 224,
               655, 715 };

int SPEE = 5;

// play the tone corresponding to the note name

for (int i = 0; i < 17; i++) {

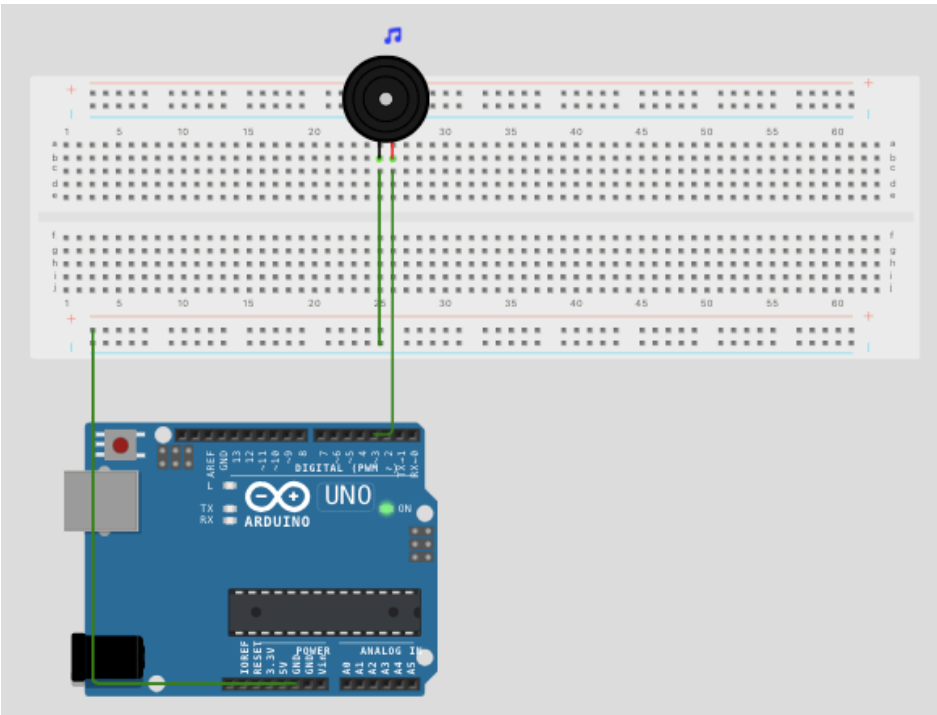
    if (names[i] == note) {
        int newduration = duration/SPEE;
        playTone(tones[i], newduration);
    }
}

void setup() {
    pinMode(buzzerPin, OUTPUT);
}

void loop() {
    for (int i = 0; i < length; i++) {
        if (notes[i] == ' ') {
            delay(beats[i] * tempo); // delay between notes
        } else {
            playNote(notes[i], beats[i] * tempo);
        }
        // time delay between notes
        delay(tempo);
    }
}

```

Output:

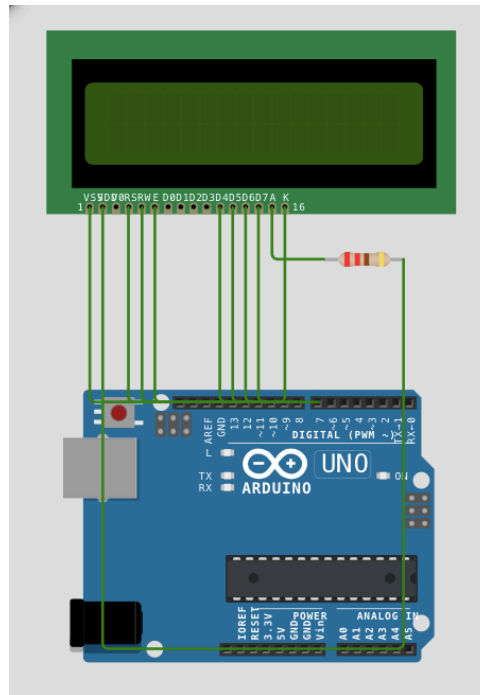


Project 8: Displaying “Hello World!” On LCD Screen [Simulation]

Aim: This project aims to make a display Hello World on an LCD Screen by connecting it to Arduino Uno.

Components Required: LCD (16x2) screen, Arduino Uno board, 4 wires.

Circuit diagram:

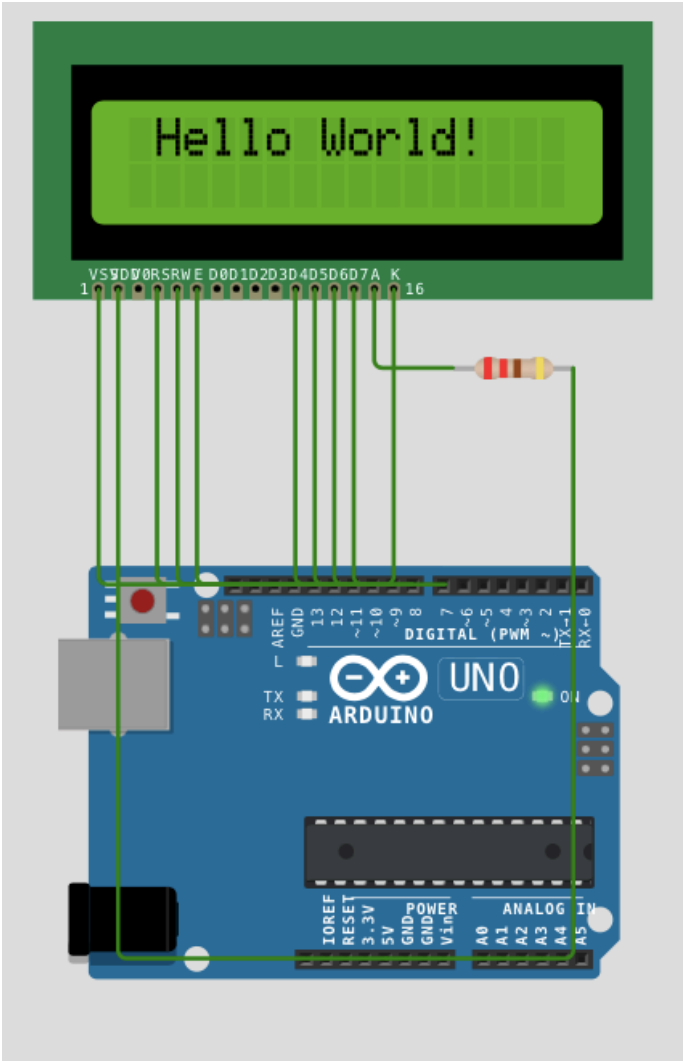


Code:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11,10,9,8,7);
void setup() {
  // put your setup code here, to run once:
  lcd.begin(16,2);
  lcd.print("Hello World!");
}

void loop() {
  // put your main code here, to run repeatedly:
  lcd.scrollDisplayLeft();
  delay(200);
  // lcd.scrollDisplayRight();
  // delay(200);
}
```

Output:

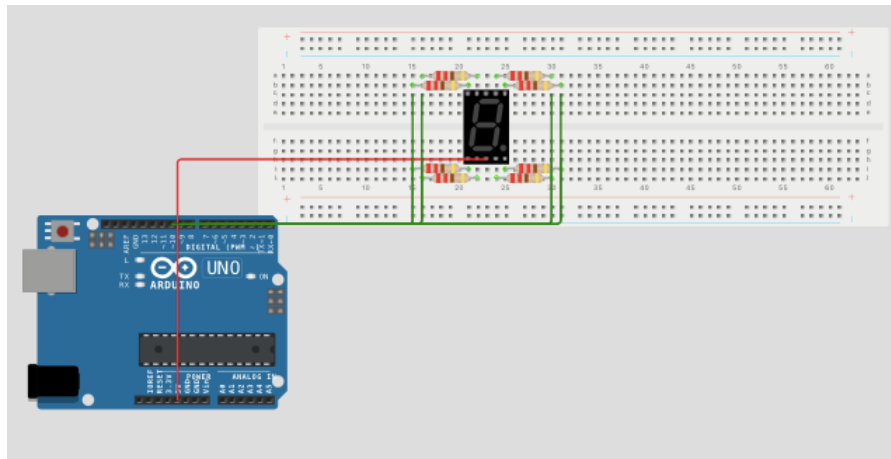


Project 9: Displaying digits on Seven Segment Display [Simulation]

Aim: This project aims to make a display digits on a seven segment display by connecting it to Arduino Uno.

Components Required: Seven Segment Display, 8 220 Ohm resistors, breadboard, Arduino Uno board, wires.

Circuit diagram:



Code:

```
const int a = 3;
const int b = 4;
const int c = 5;
const int d = 6;
const int e = 7;
const int f = 8;
const int g = 9;
const int h = 10;
void setup()
{
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);
  pinMode(h, OUTPUT);
}
```

```
}  
void one()  
{  
    digitalWrite(a, HIGH);  
    digitalWrite(b, LOW);  
    digitalWrite(c, LOW);  
    digitalWrite(d, HIGH);  
    digitalWrite(e, HIGH);  
    digitalWrite(f, HIGH);  
    digitalWrite(g, HIGH);  
    // digitalWrite(h, LOW);  
}  
void two()  
{  
    digitalWrite(a, LOW);  
    digitalWrite(b, LOW);  
    digitalWrite(c, HIGH);  
    digitalWrite(d, LOW);  
    digitalWrite(e, LOW);  
    digitalWrite(f, HIGH);  
    digitalWrite(g, LOW);  
    // digitalWrite(h, HIGH);  
}  
void three()  
{  
    digitalWrite(a, LOW);  
    digitalWrite(b, LOW);  
    digitalWrite(c, LOW);  
    digitalWrite(d, LOW);  
    digitalWrite(e, HIGH);  
    digitalWrite(f, HIGH);  
    digitalWrite(g, LOW);  
    // digitalWrite(h, HIGH);  
}  
void four()  
{  
    digitalWrite(a, HIGH);  
    digitalWrite(b, LOW);  
    digitalWrite(c, LOW);  
    digitalWrite(d, HIGH);  
    digitalWrite(e, HIGH);  
    digitalWrite(f, LOW);  
    digitalWrite(g, LOW);  
    // digitalWrite(h, HIGH);  
}  
void five()  
{  
    digitalWrite(a, LOW);
```

```
digitalWrite(b, HIGH);
digitalWrite(c, LOW);
digitalWrite(d, LOW);
digitalWrite(e, HIGH);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
// digitalWrite(h, HIGH);
}
void six()
{
digitalWrite(a, LOW);
digitalWrite(b, HIGH);
digitalWrite(c, LOW);
digitalWrite(d, LOW);
digitalWrite(e, LOW);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
// digitalWrite(h, HIGH);
}
void seven()
{
digitalWrite(a, LOW);
digitalWrite(b, LOW);
digitalWrite(c, LOW);
digitalWrite(d, HIGH);
digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
digitalWrite(g, HIGH);
// digitalWrite(h, HIGH);
}
void eight()
{
digitalWrite(a, LOW);
digitalWrite(b, LOW);
digitalWrite(c, LOW);
digitalWrite(d, LOW);
digitalWrite(e, LOW);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
// digitalWrite(h, HIGH);
}
void nine()
{
digitalWrite(a, LOW);
digitalWrite(b, LOW);
digitalWrite(c, LOW);
digitalWrite(d, LOW);
digitalWrite(e, HIGH);
```

```
digitalWrite(f, LOW);
digitalWrite(g, LOW);
// digitalWrite(h, HIGH);
}
void zero()
{
digitalWrite(a, LOW);
digitalWrite(b, LOW);
digitalWrite(c, LOW);
digitalWrite(d, LOW);
digitalWrite(e, LOW);
digitalWrite(f, LOW);
digitalWrite(g, HIGH);
// digitalWrite(h, HIGH);
}
void dot()
{
digitalWrite(a, HIGH);
digitalWrite(b, HIGH);
digitalWrite(c, HIGH);
digitalWrite(d, HIGH);
digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
digitalWrite(g, HIGH);
digitalWrite(h, LOW);
}
void loop()
{
one();
delay(1000);
two();
delay(1000);
three();
delay(1000);
four();
delay(1000);
five();
delay(1000);
six();
delay(1000);
seven();
delay(1000);
eight();
delay(1000);
nine();
delay(1000);
zero();
delay(1000);
```

```
dot();  
delay(1000);  
}
```

Output:

