Practical Robotics Projects with Arduino

(CSE 4571)

Lab Assignment No – 02

Push Button Control

Submission Date: _____

Branch: CSE		**Section:19**
Name	**Registration No.**	**Signature**

Department of Computer Science and Engineering
Institute of Technical Education and Research (Faculty of Engineering)
Siksha 'O' Anusandhan (Deemed to be University)
Bhubaneswar, Odisha-751030.

Aim:

Push Button Control – To interface push buttons with Arduino UNO to switch LEDs ON/OFF and display the state using the Serial Plotter.

Objectives:

1) **Introduction to Push Buttons and Pull-up/Pull-down resistors**

   - ✔ Briefly discussing the concept of push buttons and their usage in electronic circuits
   - ✔ Explaining the need for pull-up and pull-down resistors in push button circuits
   - ✔ Demonstrating how to use pull-up and pull-down resistors with push buttons in an LED circuit.

2) **Interfacing a Push Button with Arduino**

   - ✔ Introducing the circuit diagram for interfacing a push button with Arduino
   - ✔ **2.1 Checking the button state in serial monitor with external pull up resistor** (when the button is pressed logic 0 will be achieved)
   - ✔ **2.2 Checking the button state in serial monitor with internal pull up resistor** (when the button is pressed logic 0 will be achieved)
   - ✔ **2.3 Checking the button state in serial monitor with external pull-down resistor** (when the button is pressed logic 1 will be achieved)

3) **Interfacing a Push Button with Arduino to Control an LED based on its State.**

   - ✔ Introducing the circuit diagram for interfacing a push button to control an LED
   - ✔ **3.1 When the push button is pressed, the LED should turn off.** When the push button is released, the LED should turn on.
   - ✔ **3.2 When the push button is pressed, the LED should turn on.** When the push button is released, the LED should turn off.

4) **Utilizing the Push Button as a Toggle Switch.**

- ✔ Introducing the concept of a toggle switch and how it is used in electronic circuits
- ✔ **4.1 Demonstrates using a push button as a toggle switch and displaying the switch state on the serial monitor.** (Every time the button is pressed, it will alternate between an "ON" and "OFF" state. The state of the button is then displayed on the serial monitor, which is a tool used to display data that is being sent between the Arduino board and the computer.)
- ✔ **4.2 An external LED controlled by the push button as a toggle switch and displaying the switch state and LED state on the serial monitor.** (When the button is pressed, the LED turns on, and when it is released, the LED turns off. Additionally, the state of the switch and the LED is displayed on the serial monitor, allowing the user to monitor the current state of the system.)

5) **Controlling the Brightness of an LED with Push Buttons.**

- ✔ Introducing the concept of pulse width modulation (PWM) and how it can be used to control LED brightness
- ✔ Explaining how to use a push button to increment and decrement the PWM value
- ✔ **5.1 Using a single Push Button to Control PWM Fading of an LED with Serial Monitor** (increase or decrease the brightness of an LED using a single push button)
- ✔ **5.2 Controlling LED brightness using two push buttons and generating a buzz sound at maximum and minimum brightness levels** (Increment the brightness of an LED with one push button and decrement the brightness with the other push button, accompanied by a buzzer sound when the maximum or minimum brightness level is reached.)

Pre-Lab Questionnaire:

1) What is a push button?
2) What are the different types of push buttons?
3) What is the purpose of interfacing a push button with an Arduino?
4) What is the difference between a pull-up and pull-down resistor?
5) How is a pull-up or pull-down resistor connected in a push button circuit?
6) What is the role of an external pull-up resistor in interfacing a push button with an Arduino?
7) What is pulse width modulation (PWM) and how is it used to control LED brightness?
8) Can PWM be used to control the brightness of other types of lights besides LEDs? Why or why not?

Components/Equipment Required:

Sl. No.	Name of the Component / Equipment	Specification	Quantity
1)	Arduino UNO R3	16MHz	1
2)	Arduino UNO cable	USB Type A to B	1
3)	Resistors (carbon type)	¼ watt (330Ω)	1
		¼ watt (1kΩ)	2
4)	LED	Any colour of your choice	1
5)	Push Button	4-legged Tactile switch (5mm)	9
6)	Buzzer	Active (Small)	1
7)	Breadboard	840 Tie points	1
8)	Jumper Wire	----------	As per requirement

Objective 2

Interfacing a Push Button with Arduino

- ✔ **2.1 Checking the button state in serial monitor with external pull up resistor** (when the button is pressed logic 0 will be achieved)
- ✔ **2.2 Checking the button state in serial monitor with internal pull up resistor** (when the button is pressed logic 0 will be achieved)
- ✔ **2.3 Checking the button state in serial monitor with external pull-down resistor** (when the button is pressed logic 1 will be achieved)

Circuit / Schematic Diagram

2.1 Checking the button state in serial monitor with external pull up resistor (when the button is pressed logic 0 will be achieved)

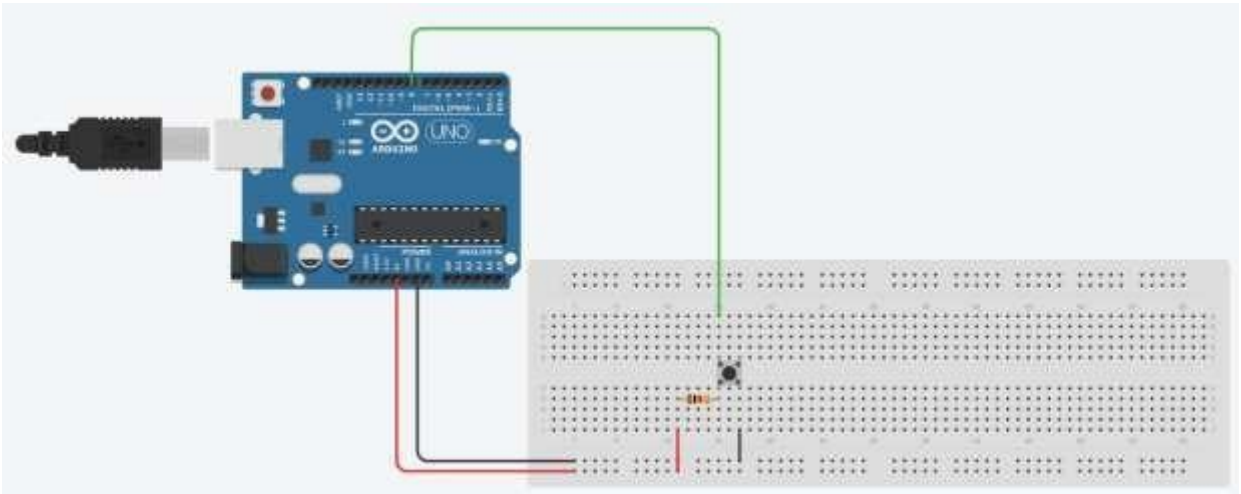
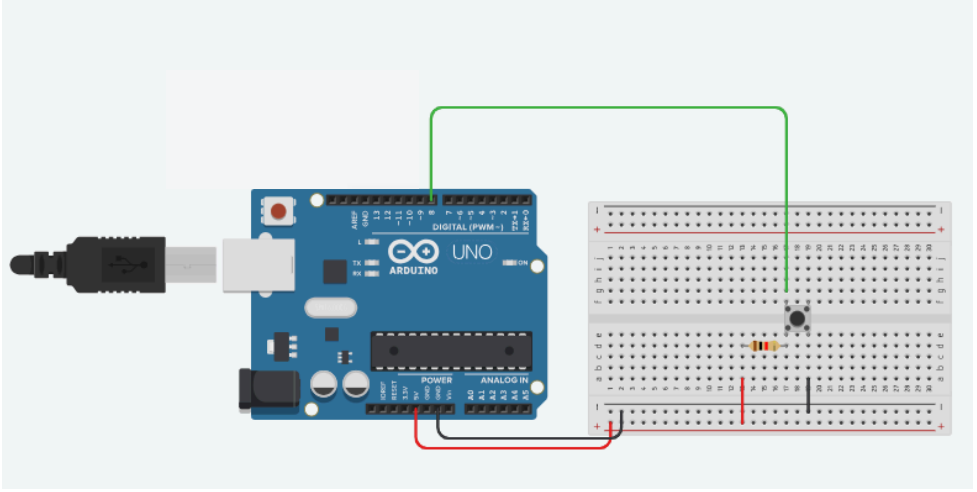
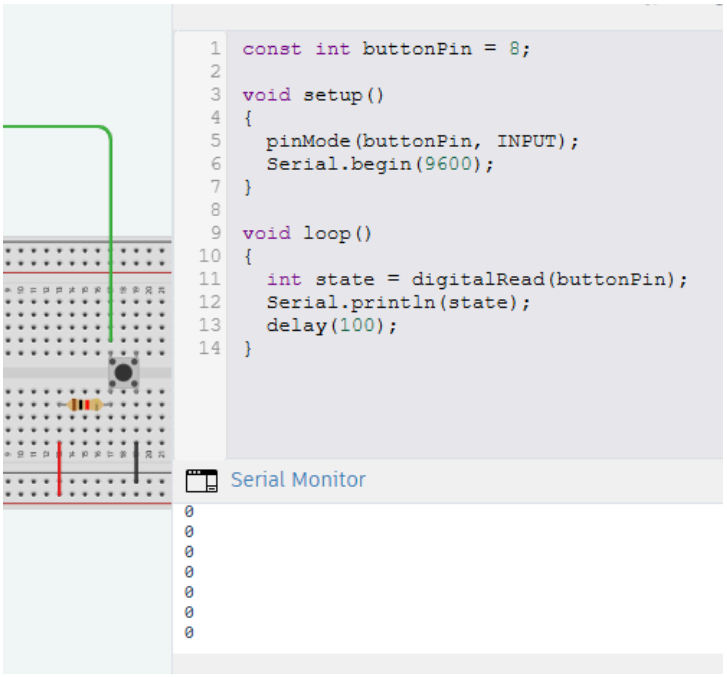


Figure 1: Schematic of Interfacing a Push Button with Arduino with external pull up resistor





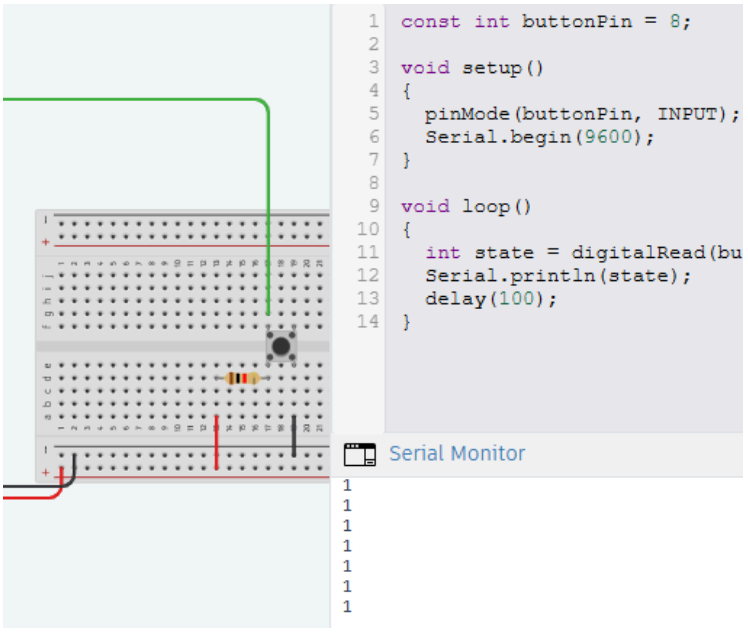
```
const int buttonPin = 8;

void setup()
{
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  int state = digitalRead(buttonPin);
  Serial.println(state);
  delay(100);
}
```

Serial Monitor

```
0
0
0
0
0
0
0
0
```



```
const int buttonPin = 8;

void setup()
{
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  int state = digitalRead(bu
  Serial.println(state);
  delay(100);
}
```

Serial Monitor

```
1
1
1
1
1
1
1
```

is pressed logic 0 will be achieved)

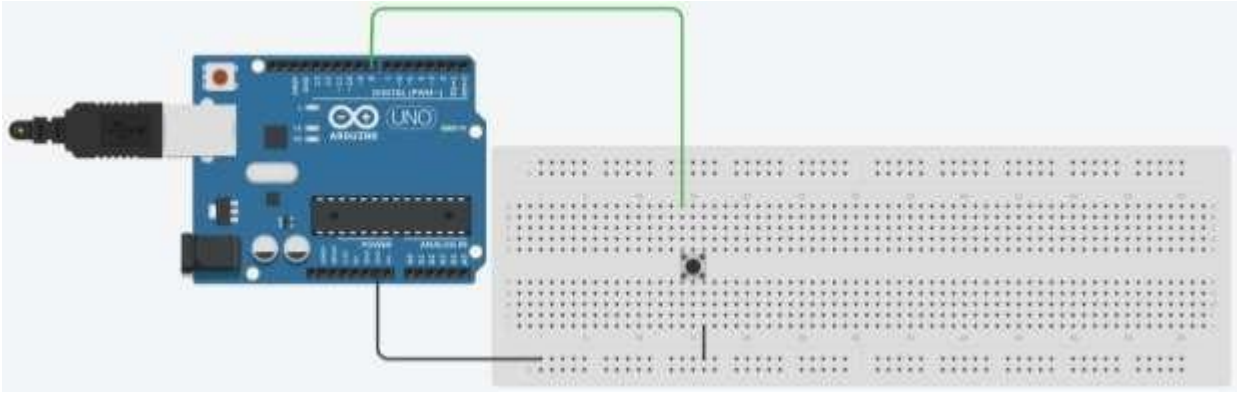


Figure 2: Schematic of Interfacing a Push Button with Arduino with internal pull-up resistor

2.3 Checking the button state in serial monitor with external pull-down resistor (when the button is pressed logic 1 will be achieved)

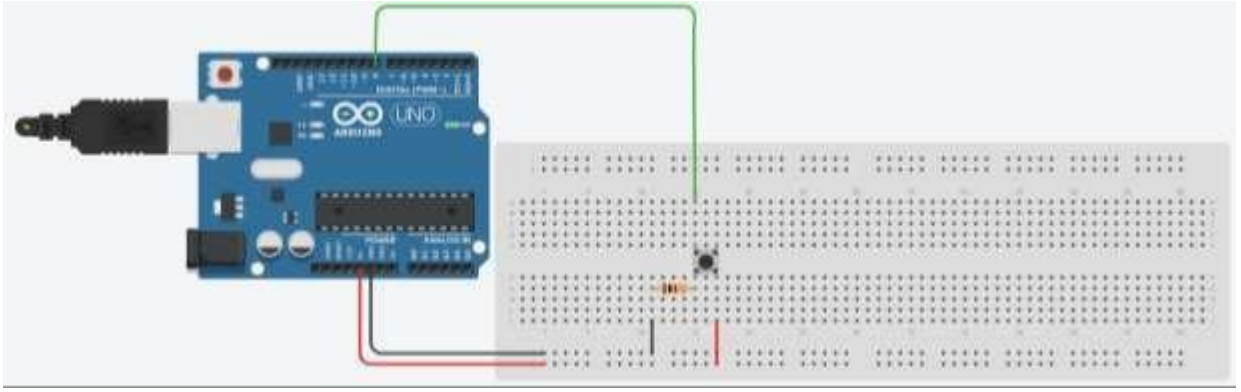


Figure 3: Schematic of Interfacing a Push Button with Arduino with external pull-down resistor

Code

2.1 Checking the button state in serial monitor with external pull up resistor (when the button is pressed logic 0 will be achieved)

```
const int buttonPin = 8;
void setup(){
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}
void loop(){
  int state = digitalRead(buttonPin);
  Serial.println(state);
  delay(100);
}
```

2.2: Checking the button state in serial monitor with internal pull up resistor (when the button is pressed logic 0 will be achieved)

```
const int buttonPin = 8;
void setup()
{
 Serial.begin(9600);
 pinMode(buttonPin, INPUT_PULLUP);
}
void loop()
{
 int state = digitalRead(buttonPin);
 Serial.println(state);
 delay(100);
}
```

2.3 Checking the button state in serial monitor with external pull-down resistor (when the button is pressed logic 1 will be achieved)

```
#define BUTTON_PIN 8
void setup() {
 Serial.begin(9600);
 pinMode(BUTTON_PIN, INPUT);
}
void loop() {
 int buttonState = digitalRead(BUTTON_PIN);
 Serial.println(buttonState);
 delay(100);
}
```

Observation

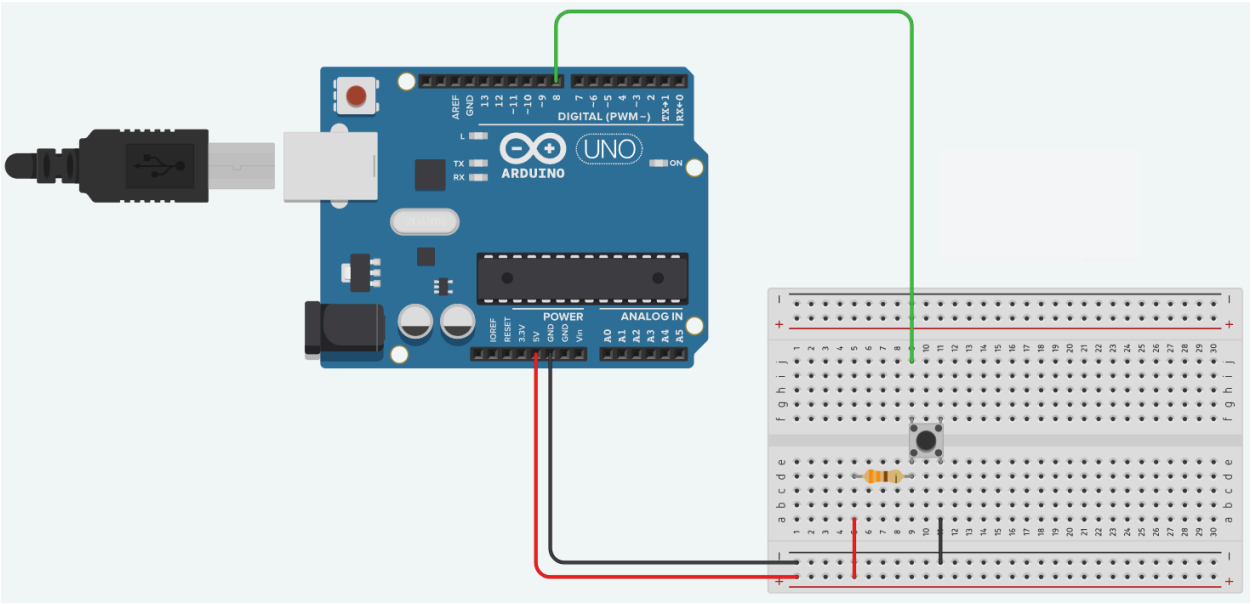


Figure 4: (Simulation based Interfacing a Push Button with Arduino with external pull-up resistor)

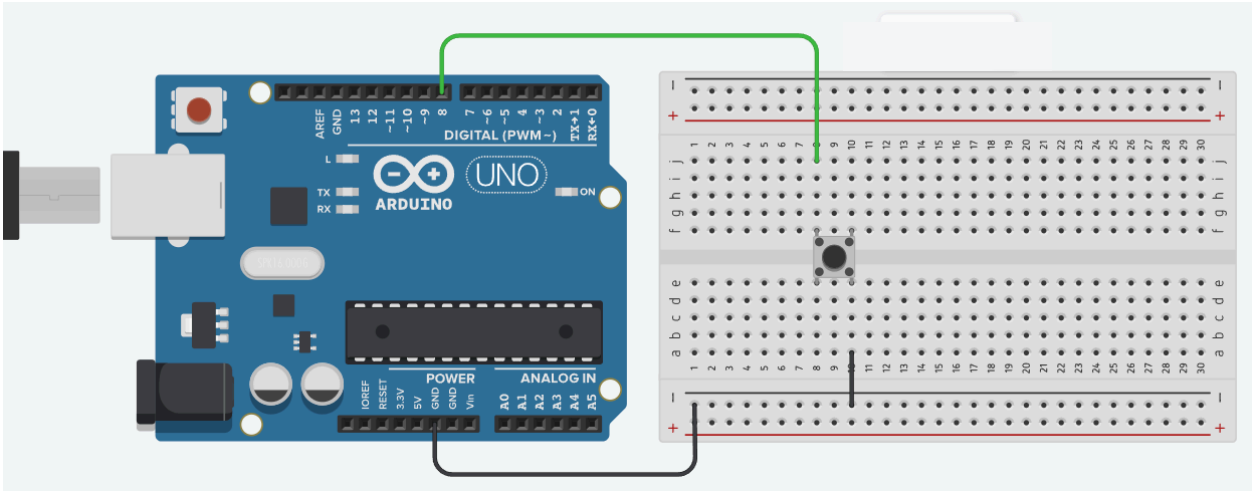


Figure 5: (Simulation based Interfacing a Push Button with Arduino with internal pull-up resistor)

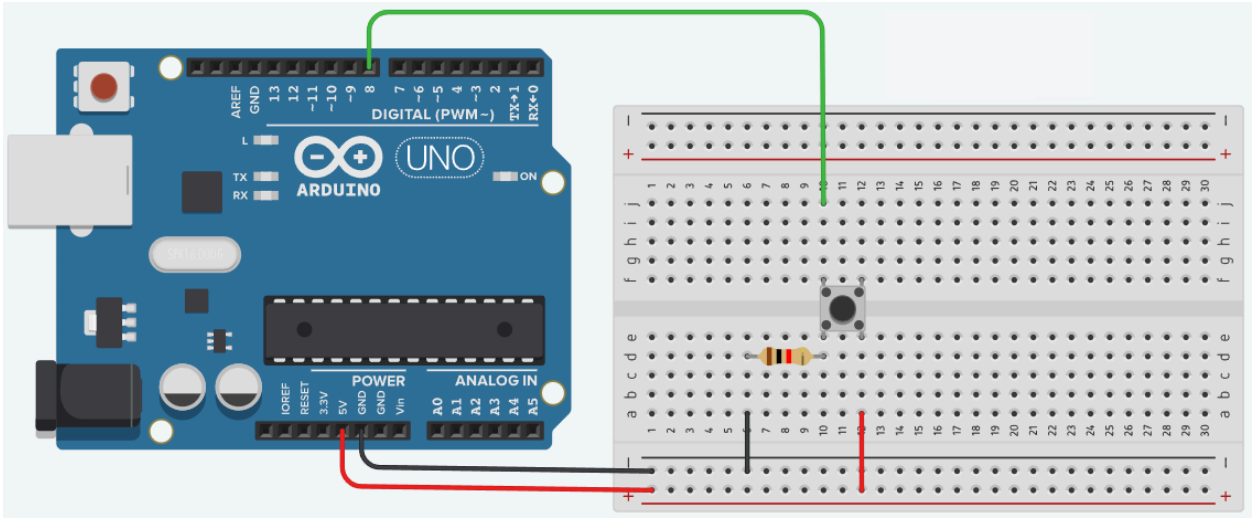


Figure 6: (Simulation based Interfacing a Push Button with Arduino with external pull-down resistor)

Objective 3

Interfacing a Push Button with Arduino to Control an LED based on its State.

- ✔ **3.1 When the push button is pressed, the LED should turn off.** When the push button is released, the LED should turn on.
- ✔ **3.2 When the push button is pressed, the LED should turn on.** When the push button is released, the LED should turn off.

Circuit / Schematic Diagram

3.1 When the push button is pressed, the LED should turn off. When the push button is released, the LED should turn on.

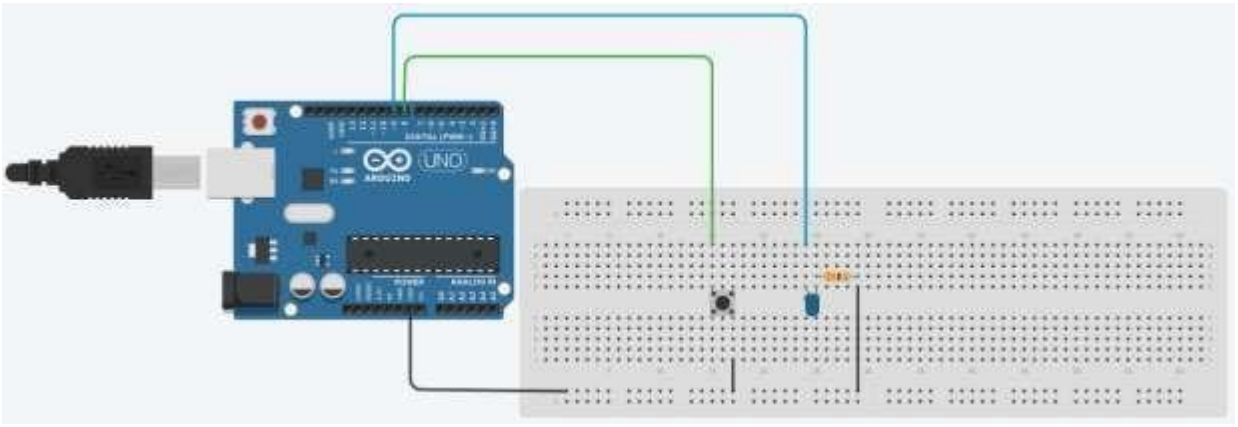


Figure 7: Schematic of interfacing a Push Button with Arduino to Control an LED: When the push button is pressed, the LED should turn off

3.2 When the push button is pressed, the LED should turn on. When the push button is released, the LED should turn off.

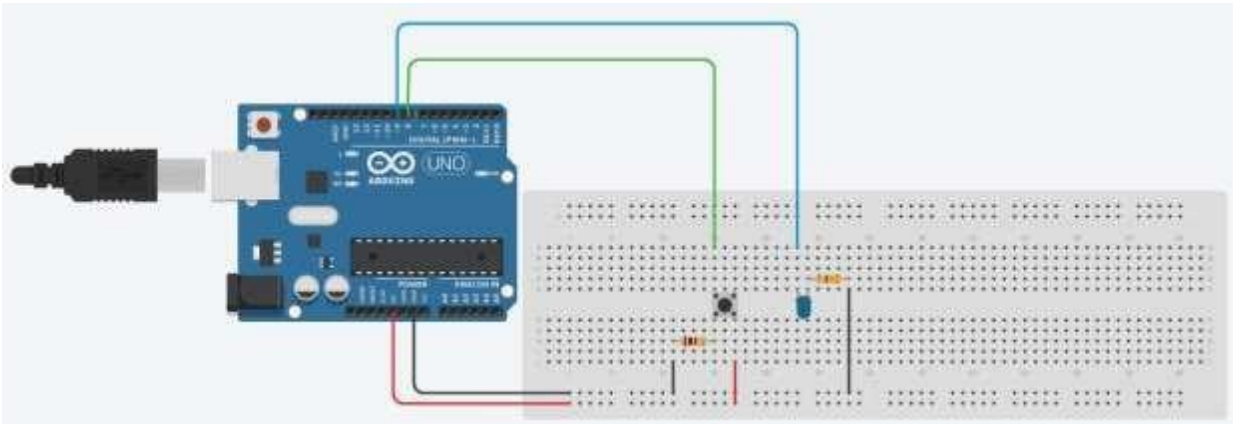


Figure 8: Schematic of interfacing a Push Button with Arduino to Control an LED: When the push button is pressed, the LED should turn on.

Code

3.1 When the push button is pressed, the LED should turn off. When the push button is released, the LED should turn on.

```
int buttonPin = 8;
int ledPin = 9;
int buttonState = 0;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == LOW) {
    digitalWrite(ledPin, LOW);
  } else {
    digitalWrite(ledPin, HIGH);
  }
}
```

3.2 When the push button is pressed, the LED should turn on. When the push button is released, the LED should turn off.

```
int buttonPin = 8;
int ledPin = 9;
int buttonState = 0;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == LOW) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

Observation

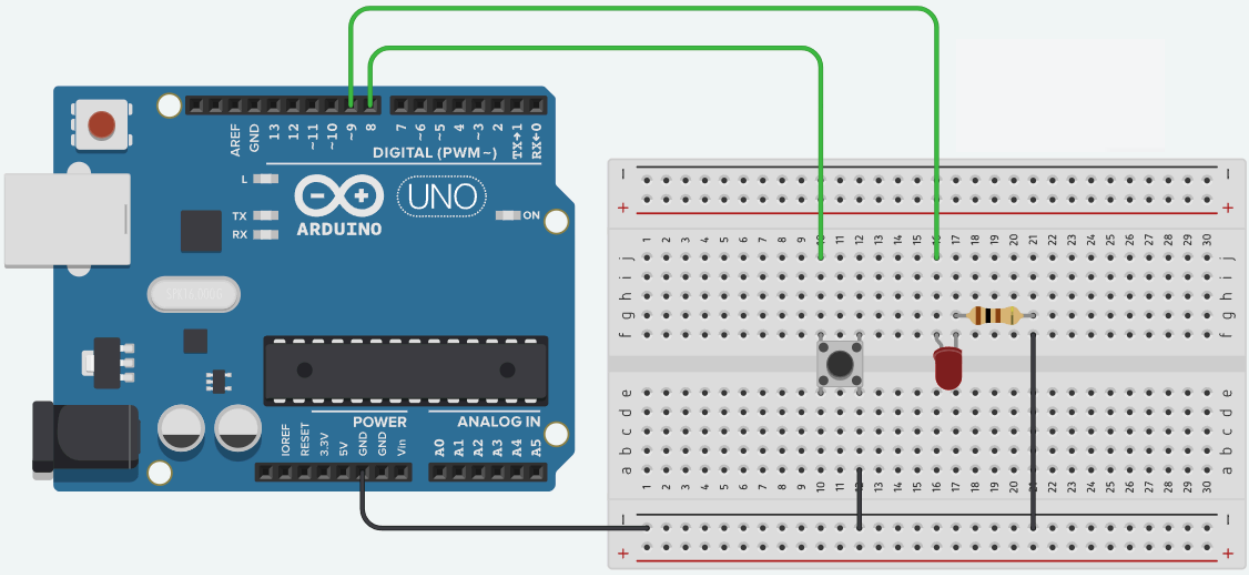


Figure 9: (Simulation based interfacing a Push Button with Arduino to Control an LED: When the push button is pressed, the LED should turn off.)

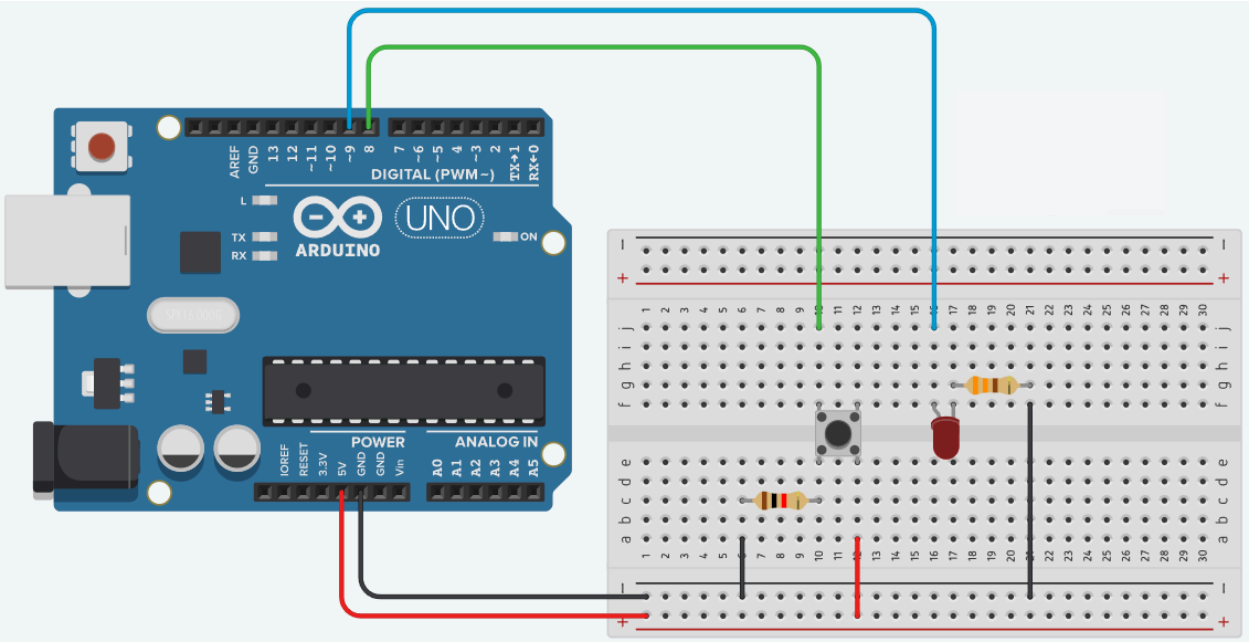


Figure 10: (Simulation based interfacing a Push Button with Arduino to Control an LED: When the push button is pressed, the LED should turn on.)

Objective 4

Utilizing the Push Button as a Toggle Switch.

- ✔ **4.1 Demonstrates using a push button as a toggle switch and displaying the switch state on the serial monitor.** (Every time the button is pressed, it will alternate between an "ON" and "OFF" state. The state of the button is then displayed on the serial monitor, which is a tool used to display data that is being sent between the Arduino board and the computer.)

- ✔ **4.2 An external LED controlled by the push button as a toggle switch and displaying the switch state and LED state on the serial monitor.** (When the button is pressed, the LED turns on, and when it is released, the LED turns off. Additionally, the state of the switch and the LED is displayed on the serial monitor, allowing the user to monitor the current state of the system.

Circuit / Schematic Diagram

4.1 Demonstrates using a push button as a toggle switch and displaying the switch state on the serial monitor.

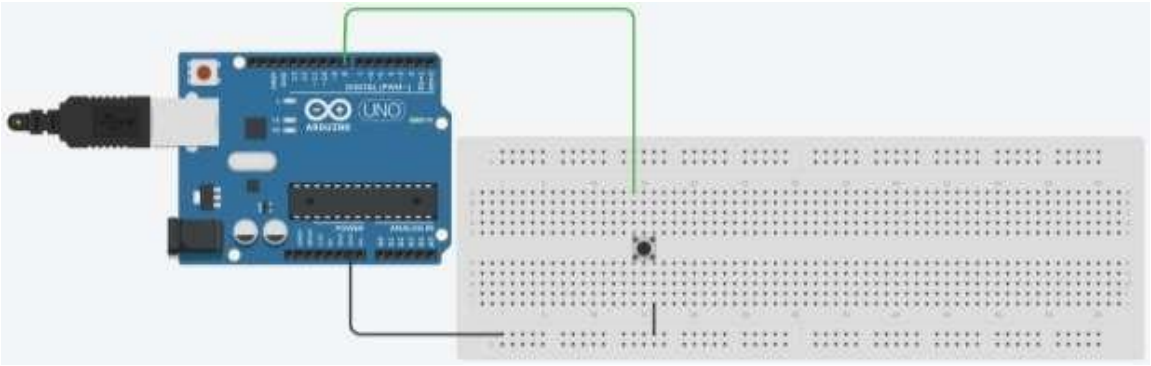


Figure 11: Schematic of interfacing a Push Button with Arduino as a toggle switch and displaying the switch state on the serial monitor.

4.2 An external LED controlled by the push button as a toggle switch and displaying the switch state and LED state on the serial monitor.

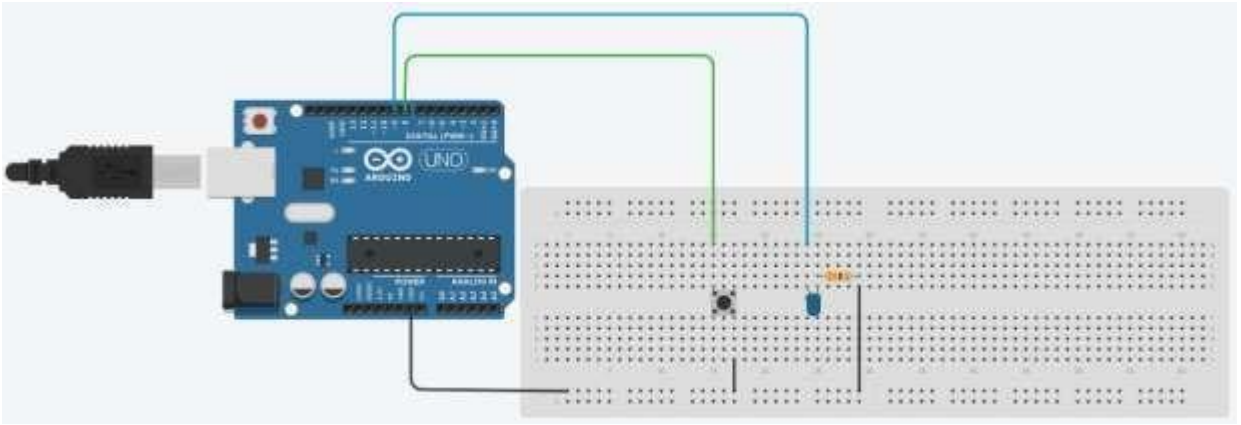


Figure 12: Schematic of interfacing a Push Button with Arduino as a toggle switch and displaying the switch state and LED state on the serial monitor.

Code

4.1 Demonstrates using a push button as a toggle switch and displaying the switch state on the serial monitor.

```
int buttonPin = 2;
int buttonState = 0;
int lastButtonState = 0;
bool toggleState = false;

void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH && lastButtonState == LOW) {
    toggleState = !toggleState;  // Toggle the state
    if (toggleState) {
      Serial.println("ON");
    } else {
      Serial.println("OFF");
    }
    delay(50); // Debounce delay
  }

  lastButtonState = buttonState;
}
```

4.2: An external LED controlled by the push button as a toggle switch and displaying the switch state and LED state on the serial monitor.

```
int buttonPin = 8;
int ledPin = 9;
int buttonState = 0;
int lastButtonState = 0;
bool ledState = false;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH && lastButtonState == LOW) {
    ledState = !ledState;  // Toggle LED state
    if (ledState) {
      digitalWrite(ledPin, HIGH);
      Serial.println("LED ON");
    } else {
      digitalWrite(ledPin, LOW);
      Serial.println("LED OFF");
    }
    delay(50); // Debounce delay
  }

  lastButtonState = buttonState;
}
```

Observation

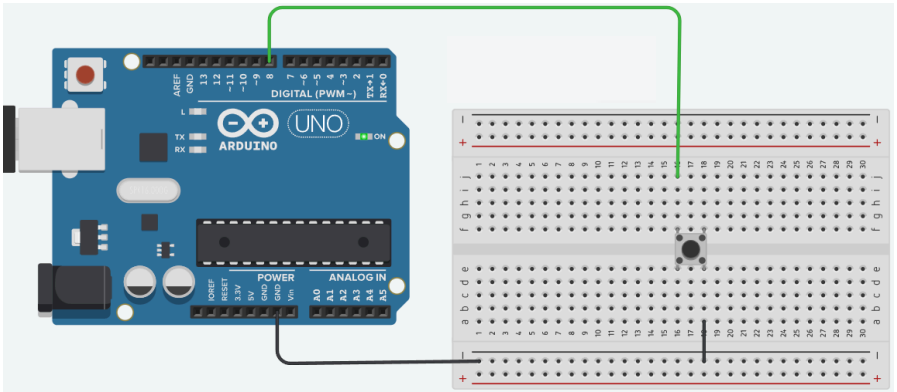


Figure 13: (Simulation based demonstrating a push button as a toggle switch and displaying the switch state on the serial monitor.)

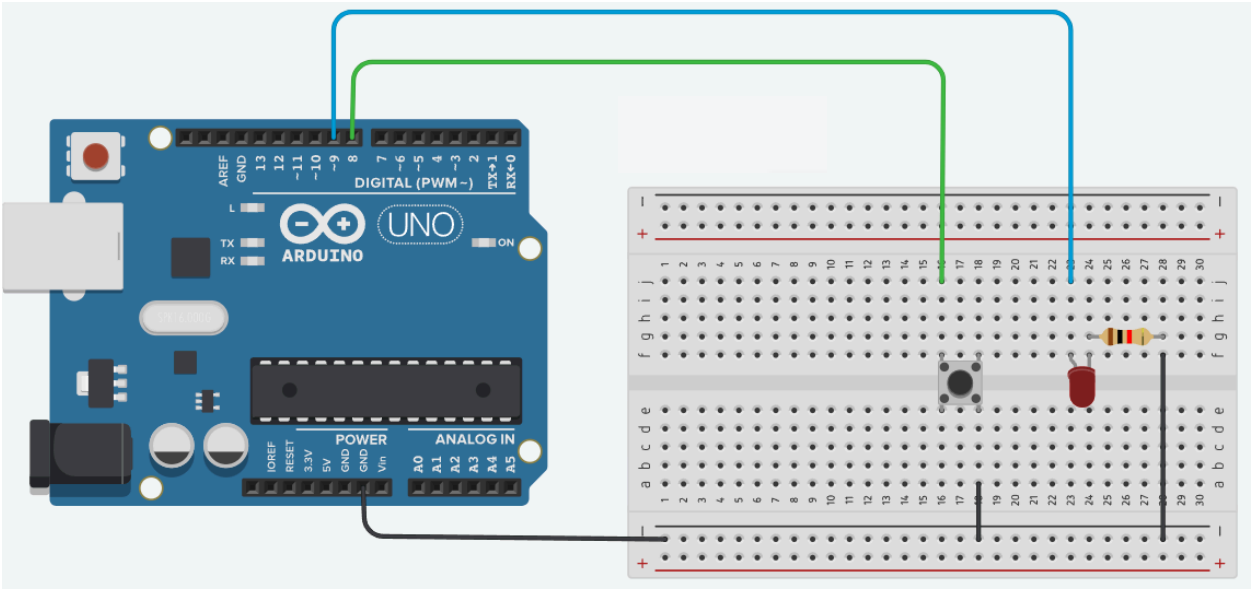


Figure 14: (Simulation based an external LED controlled by the push button as a toggle switch and displaying the switch state and LED state on the serial monitor.

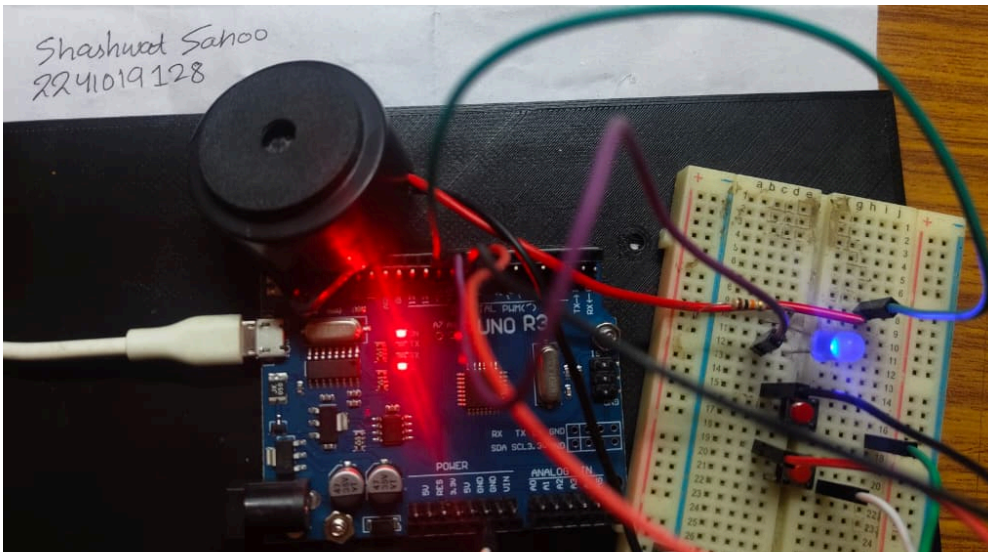


Figure 15: (Hardware Implementation based an external LED controlled by the push button as a toggle switch and displaying the switch state and LED state on the serial monitor.)

Objective 5

Controlling the Brightness of an LED with Push Buttons.

- ✔ **5.1 Using a single Push Button to Control PWM Fading of an LED with Serial Monitor** (increase or decrease the brightness of an LED using a single push button)
- ✔ **5.2 Controlling LED brightness using two push buttons and generating a buzz sound at maximum and minimum brightness levels** (Increment the brightness of an

LED with one push button and decrement the brightness with the other push button, accompanied by a buzzer sound when the maximum or minimum brightness level is reached.)

Circuit / Schematic Diagram

5.1 Using a single Push Button to Control PWM Fading of an LED with Serial Monitor

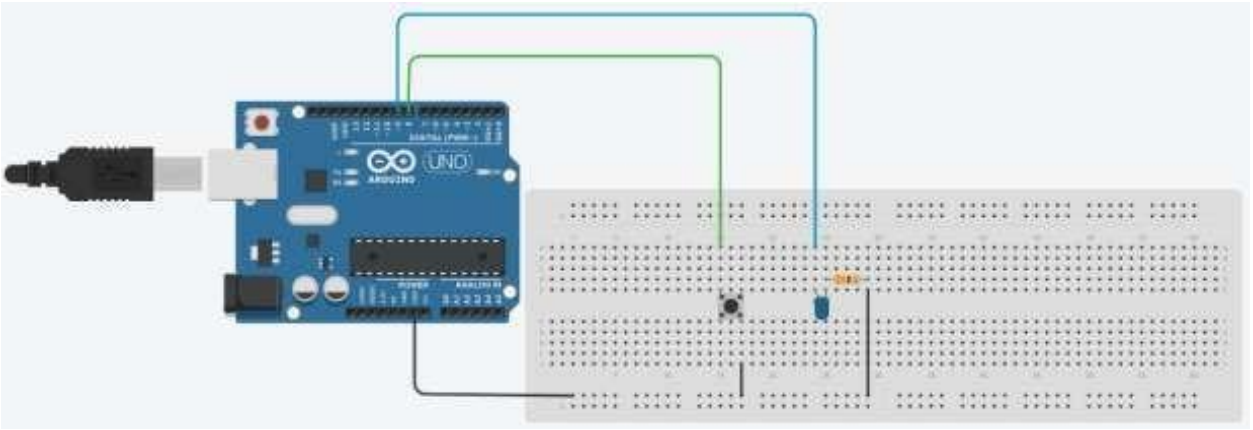


Figure 16: Schematic of interfacing a single Push Button with Arduino to Control PWM Fading of an LED with Serial Monitor

5.2: Controlling LED brightness using two push buttons and generating a buzz sound at maximum and minimum brightness levels

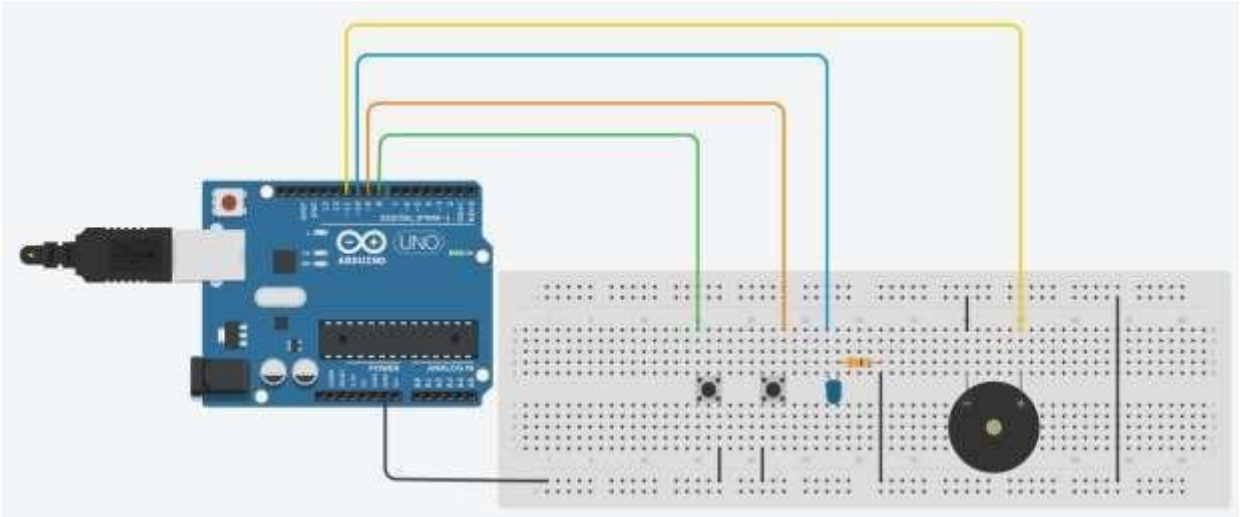


Figure 17: Schematic of controlling LED brightness using two push buttons and generating a buzz sound at maximum and minimum brightness levels

Code

5.1 Using a single Push Button to Control PWM Fading of an LED with Serial Monitor

```
int buttonPin = 8;
int ledPin = 9;
int brightness = 0;
int step = 25;
int buttonState = 0;
int lastButtonState = 0;
bool increasing = true;
void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH && lastButtonState == LOW) {
    if (increasing) {
      brightness += step;
      if (brightness >= 255) {
        brightness = 255;
        increasing = false;
      }
    } else {
      brightness -= step;
      if (brightness <= 0) {
        brightness = 0;
        increasing = true;
      }
    }
    analogWrite(ledPin, brightness);
    Serial.print("Brightness: ");
    Serial.println(brightness);
    delay(150); // Debounce delay
  }
  lastButtonState = buttonState;
}
```

5.2 Controlling LED brightness using two push buttons and generating a buzz sound at maximum and minimum brightness levels

```
int incButton = 8;
int decButton = 9;
int ledPin = 10;
int buzzerPin =11;
int brightness = 0;
```

```
int step = 25;
void setup() {
  pinMode(incButton, INPUT);
  pinMode(decButton, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (digitalRead(incButton) == HIGH) {
    if (brightness < 255) {
      brightness += step;
      if (brightness > 255) brightness = 255;
      analogWrite(ledPin, brightness);
      Serial.print("Brightness: ");
      Serial.println(brightness);
      delay(150);
    } else {
      tone(buzzerPin, 1000, 200);
      Serial.println("Max Brightness Reached");
      delay(300);
    }
  }

  if (digitalRead(decButton) == HIGH) {
    if (brightness > 0) {
      brightness -= step;
      if (brightness < 0) brightness = 0;
      analogWrite(ledPin, brightness);
      Serial.print("Brightness: ");
      Serial.println(brightness);
      delay(150);
    } else {
      tone(buzzerPin, 500, 200);
      Serial.println("Min Brightness Reached");
      delay(300);
    }
  }
}
```

Observation

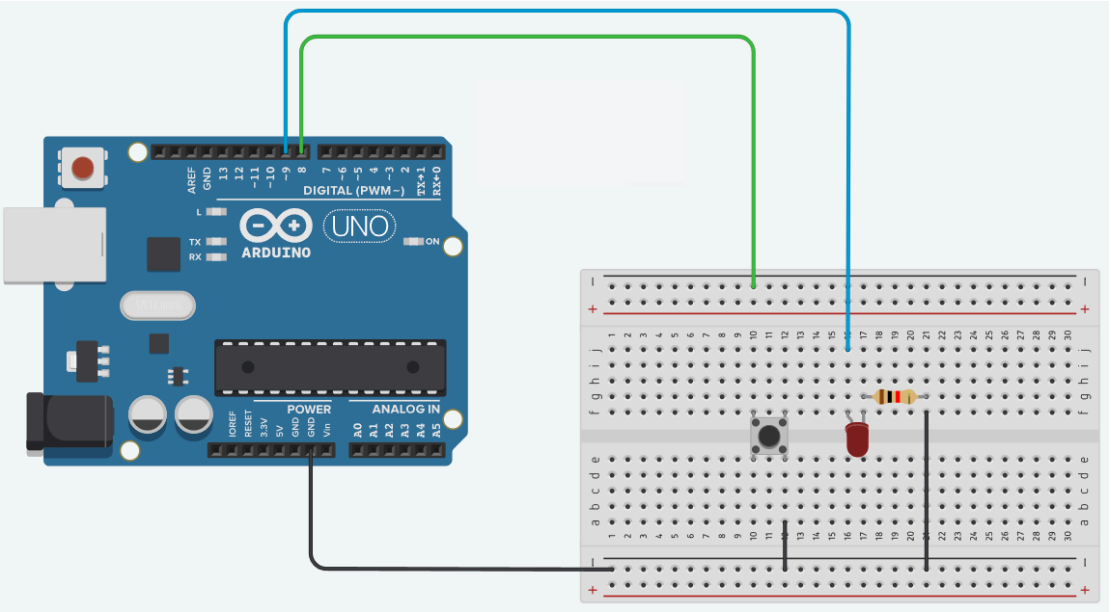


Figure 18: (Simulation based using a single Push Button to Control PWM Fading of an LED with Serial Monitor)

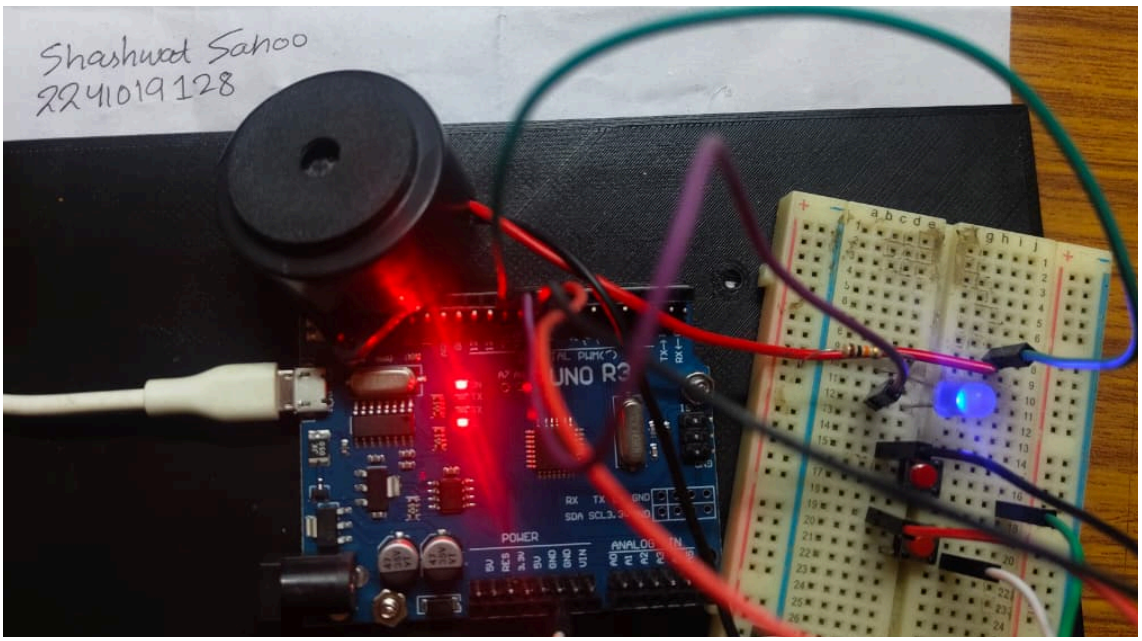


Figure 19: (Hardware Implementation based using a single Push Button to Control PWM Fading of an LED with Serial Monitor)

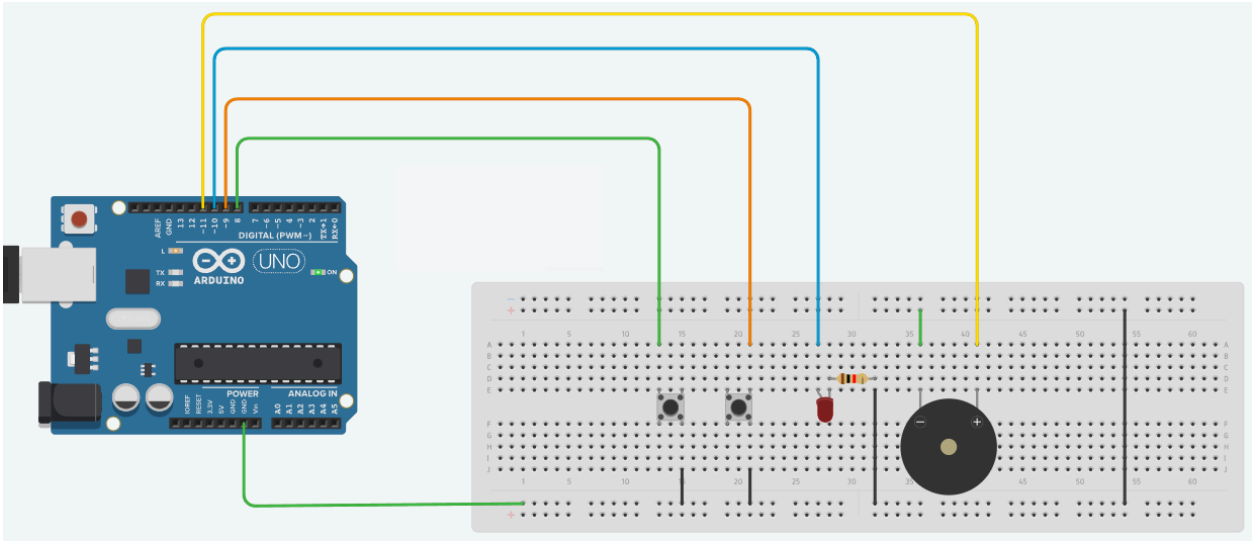


Figure 20: (Simulation based controlling LED brightness using two push buttons and generating a buzz sound at maximum and minimum brightness levels)

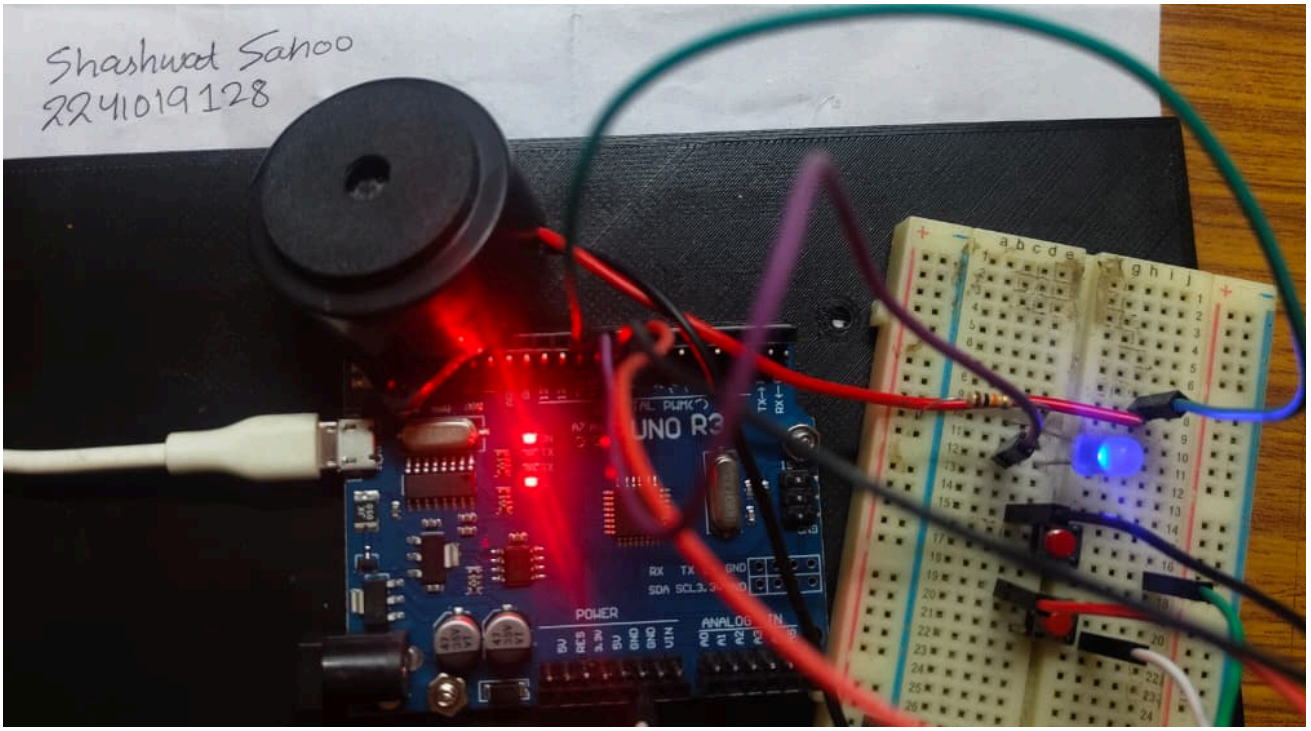


Figure 21: (Hardware Implementation based controlling LED brightness using two push buttons and generating a buzz sound at maximum and minimum brightness levels)

Conclusion:

Precautions:

Post Experiment Questionnaire:

1) A push button is connected to an Arduino board with an external pull-up resistor of 10kΩ. What will be the state of the button when it is pressed?

2) In the above problem, if the external pull-up resistor is changed to 4.7kΩ, what will be the state of the button when it is pressed?

3) A push button is connected to an Arduino board with an internal pull-up resistor enabled. What will be the state of the button when it is pressed?

4) In the above problem, if the internal pull-up resistor is disabled, what will be the state of the button when it is pressed?

5) A push button is connected to an Arduino board with an external pull-down resistor of 10kΩ. What will be the state of the button when it is pressed?

6) What is the maximum brightness level that can be achieved with PWM if the LED has a maximum current rating of 20mA and a forward voltage of 2V? (Assume a 5V power supply)

7) If the PWM duty cycle is set to 25%, what is the average current flowing through the LED if its resistance is 220 ohms? (Assume a 5V power supply)

(Signature of the Faculty)

Date: _____

(Signature of the Student)

Name: _____

Registration No.: _____

Branch: _____

Section _____