

Experiment – 9

Aim: Serial I/O - PWM can be done in this circuit. By providing different values of R, G, B in the led color function.

Components

- Arduino Uno R3
- Small Breadboard
- Jumper cable
- LED x1
- Oscilloscope

Theory

1. Arduino Uno R3

The Arduino Uno R3 is the main microcontroller in this circuit. It acts as the brain of the setup, executing the programmed code to control the LED. It provides both power and signal to the circuit, making it an essential component for automation and control projects.

2. Small Breadboard

The small breadboard is used for easy and temporary circuit connections without soldering. It allows the LED and other components to be connected securely while maintaining flexibility for modifications.

3. Jumper Cable:

Used to make electrical connections between the Arduino, breadboard, and components.

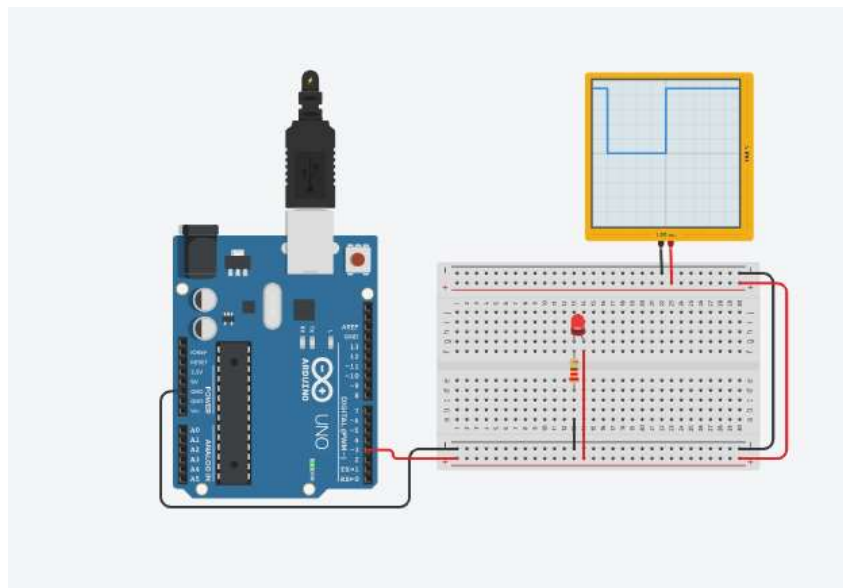
4. LED (Light Emitting Diode):

The LED emits light when current flows through it, serving as a visual indicator of the circuit's status and functionality.

5. Oscilloscope:

The oscilloscope visualizes electrical signals by displaying voltage waveforms over time, allowing for analysis of circuit behavior and performance.

Circuit Design



Procedure to Design a Light-Controlled Circuit using Arduino Uno R3 and LED

Step 1: Add Components to the Workspace

1. Search for Arduino Uno R3 and add it to the workspace.
2. Search for a small breadboard and place it next to the Arduino.
3. Search for an LED (Light Emitting Diode) and position it on the breadboard.
4. Search for a 10k Ω resistor to use with the LED.
5. Use jumper cables to connect the components properly.

Step 2: Build the Circuit Connections

LED Connections:

1. Connect the anode (+) of the LED to a digital pin (e.g., pin 3) on the Arduino.
2. Connect the cathode (-) of the LED to one terminal of the 10k Ω resistor.
3. Connect the other terminal of the 10k Ω resistor to GND on the Arduino.

Step 3: Code for LED Control

```
int led = 3;
int brightness = 0;
int fadeAmount = 5;
void setup() {
  pinMode(led, OUTPUT);
}
void loop() {
  analogWrite(led, brightness);
  brightness = brightness + fadeAmount;
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  delay(35);
}
```

Conclusion

In this project, we successfully implemented a circuit that utilizes the Arduino Uno R3 to control the brightness of an LED through Pulse Width Modulation (PWM). By varying the brightness value sent to the LED, we demonstrated how to create a smooth fading effect, showcasing the capabilities of the Arduino in automation and control projects. The use of a small breadboard and jumper cables facilitated easy connections, allowing for flexibility in circuit design.