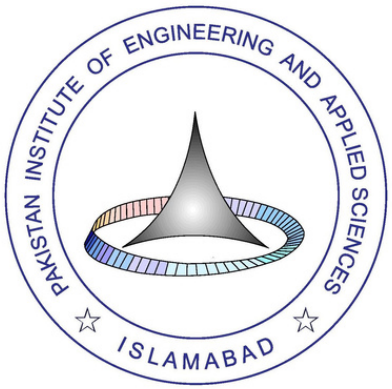



# Actuator Control

Pakistan Institute of Engineering and Applied Sciences

**Objectives:**

· Understand Actuator Control: Learn how to control physical devices using digital signals.
· Relay Operation: Interface a relay module with the WeMos D1 Mini to switch appliances (on/off control).
· LED Signaling with PWM: Control an LED's brightness using Pulse Width Modulation (PWM).
· Hands-on Experience: Gain practical skills in wiring, coding, and debugging actuator circuits in IoT applications.

**Required Components:**

· WeMos D1 Mini (ESP8266)
· Relay Module (compatible with 3.3V logic or with a transistor driver)
· LED (any color) plus a 220Ω current-limiting resistor
· (Optional) Potentiometer (to adjust PWM value dynamically)
· Breadboard and Jumper Wires
· USB cable for programming

# Task 1: Relay Control for Switching Appliances

## Arduino Code: (.ino file)

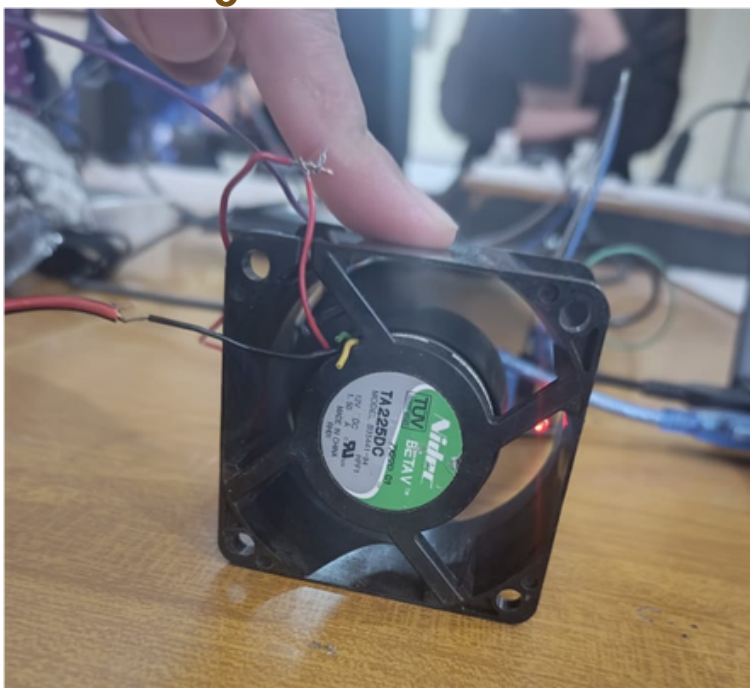
```
#define RELAY_PIN D2  // Relay control connected to digital pin D2

void setup() {
    pinMode(RELAY_PIN, OUTPUT);
    Serial.begin(115200);
    Serial.println("Relay Control Initialized");
}

void loop() {
    // Turn relay ON for 5 seconds, then OFF for 5 seconds
    digitalWrite(RELAY_PIN, HIGH);
    Serial.println("Relay ON");
    delay(5000);

    digitalWrite(RELAY_PIN, LOW);
    Serial.println("Relay OFF");
    delay(5000);
```

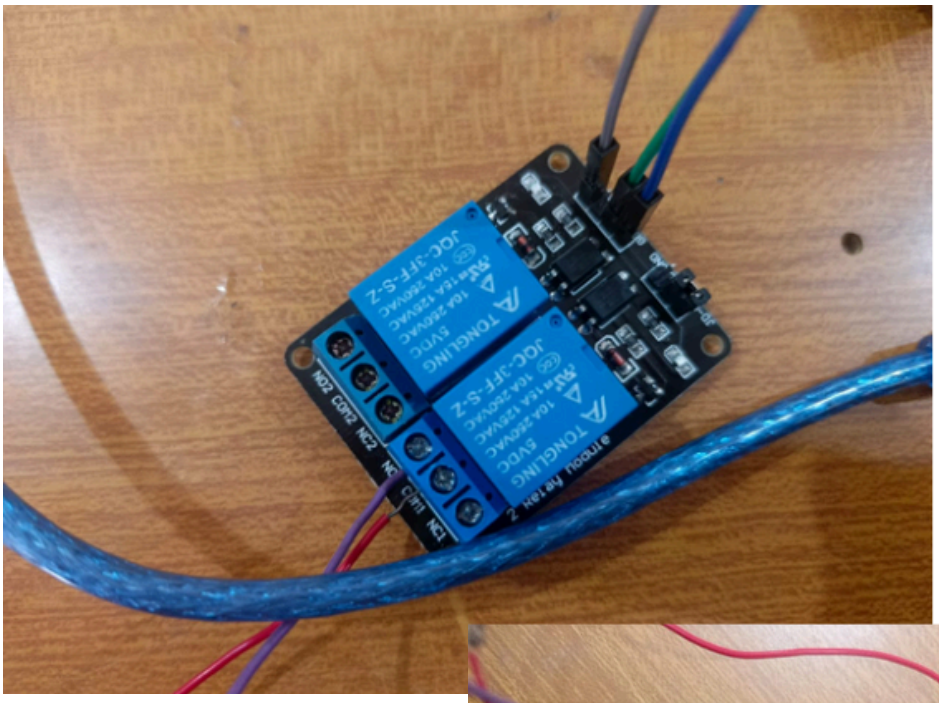
tput

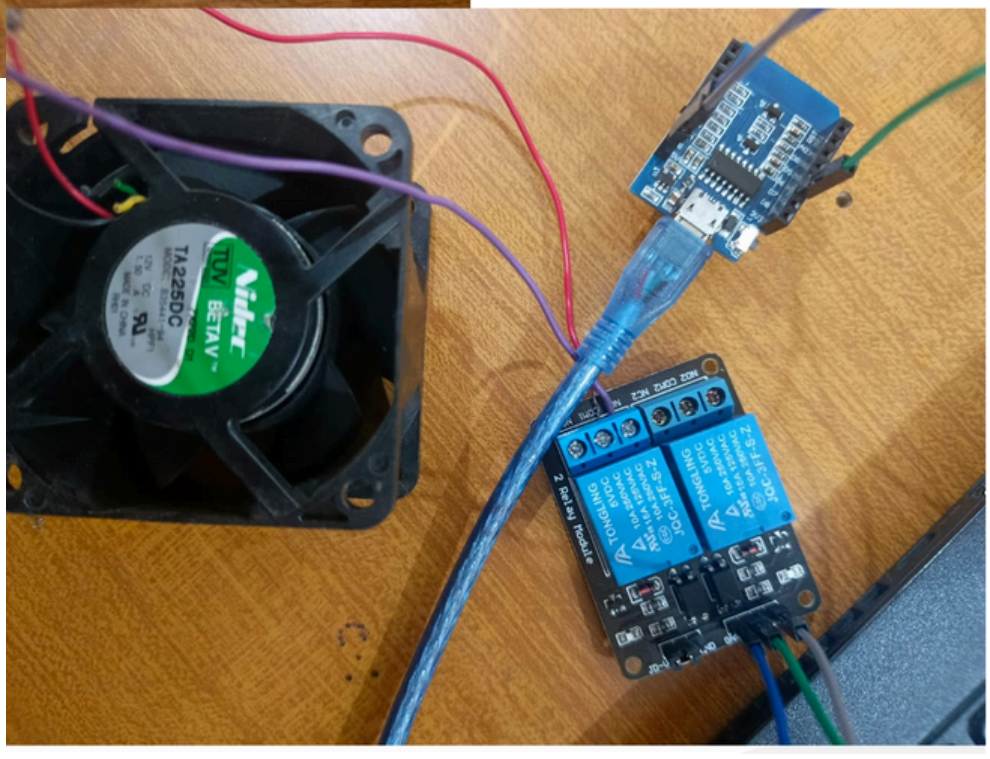
Output Serial Monitor

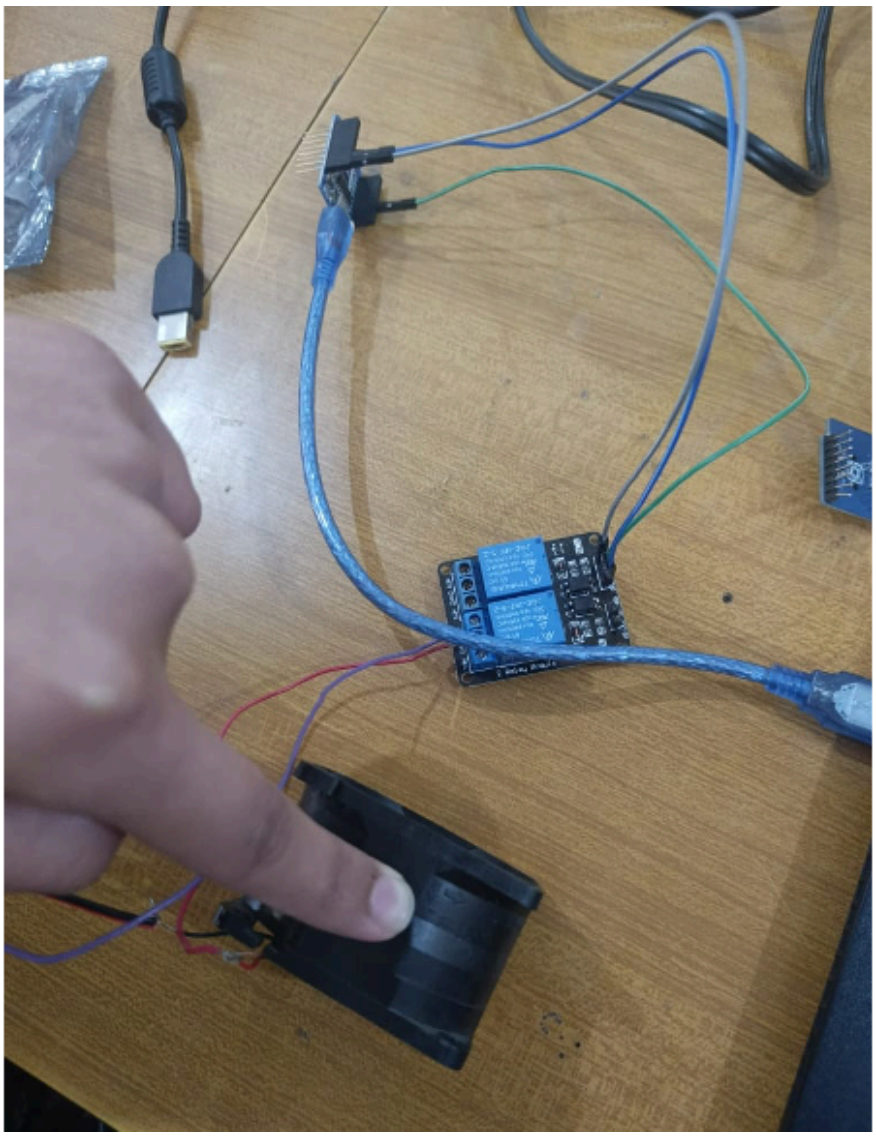
```
Writing at 0x00024000... (76 %)
Writing at 0x00028000... (84 %)
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 272592 bytes (200690 compressed) at 0x00000000 in 4.9 seconds (effective 442.7 kbit/s)...
Hash of data verified.
```

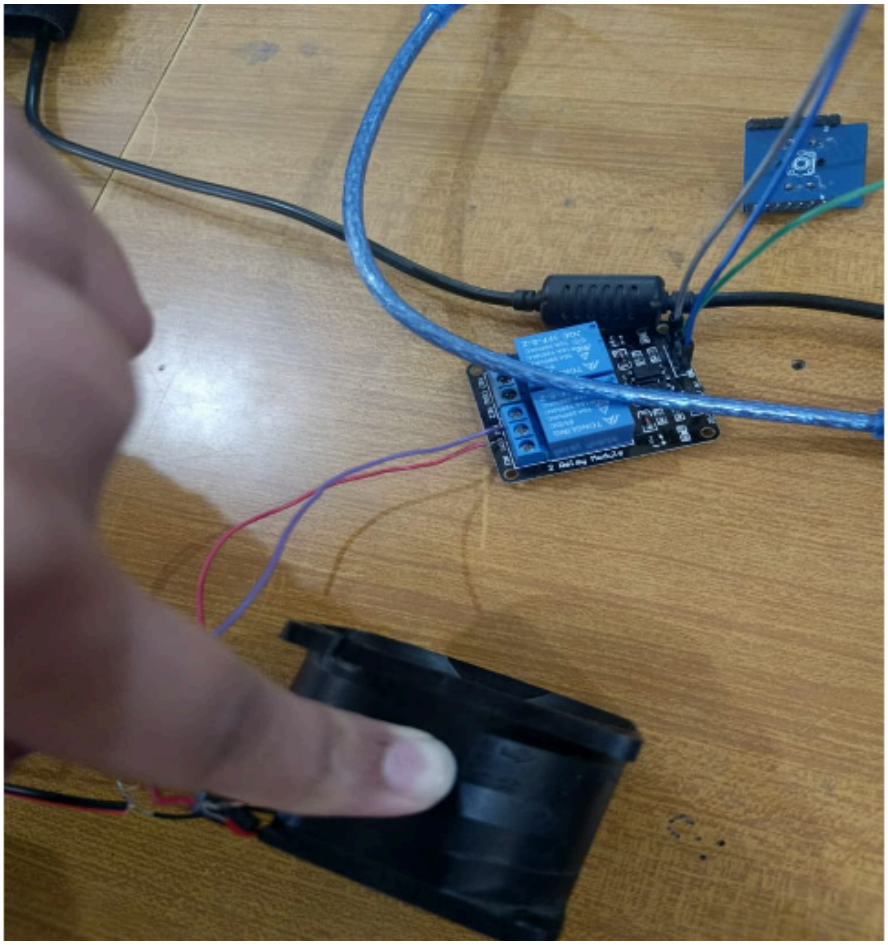
## Circuit Diagram:







## Short Description:

**Purpose and Working of Relays and PWM:**
A relay is an electrically operated switch that allows a low-voltage signal to control a higher-voltage circuit. In this experiment, a WeMos D1 Mini control a relay to turn an FAN circuit on and off.

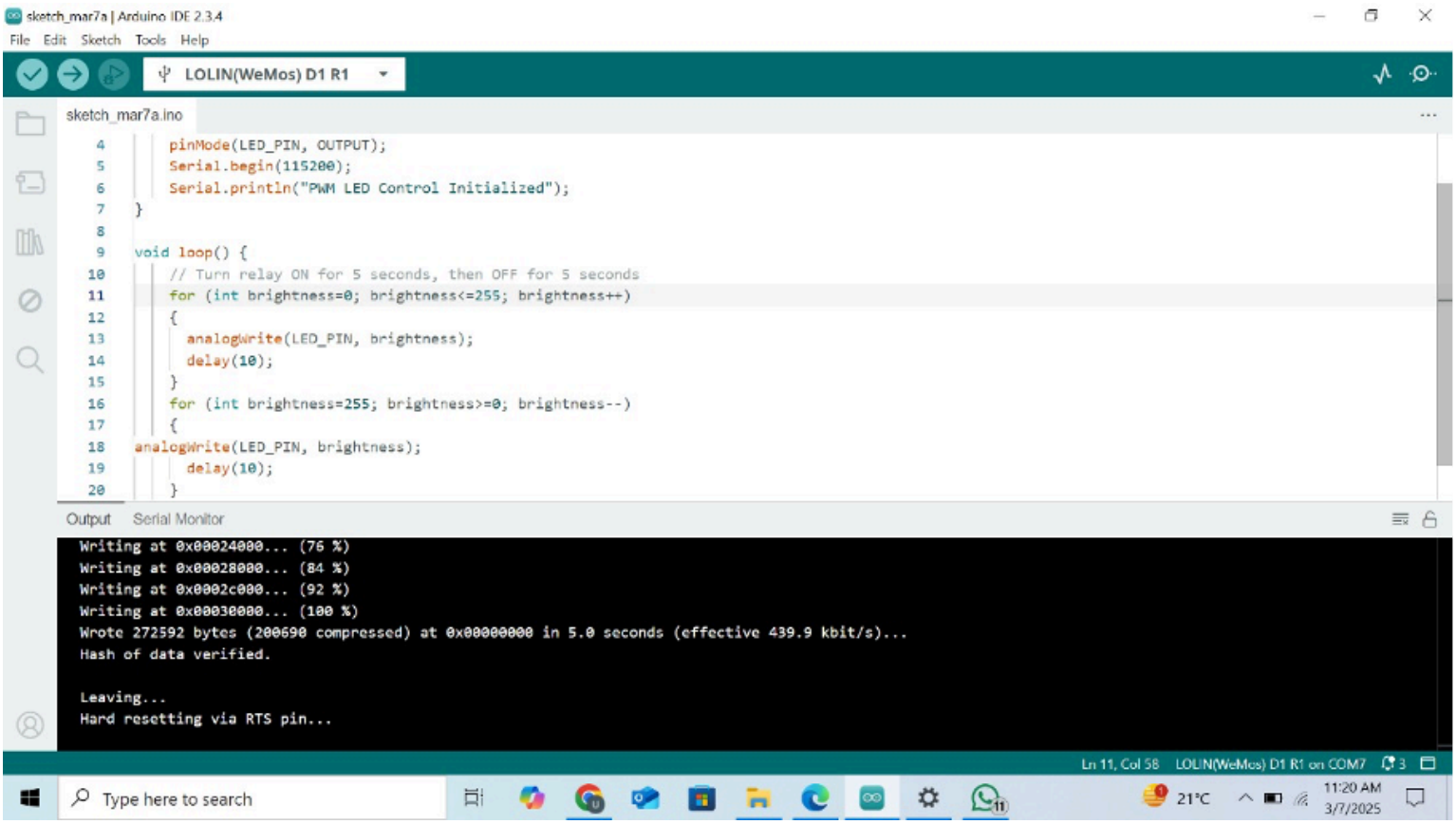
**How the Relay and LED Circuits Were Assembled:**
The relay module was connected to the WeMos D1 Mini:
VCC → 5V, GND → Ground, IN → D2 (control signal).
The relay's NO (Normally Open) contacts switched a 12V-powered FAN circuit instead of AC mains for safety.

**Challenges Encountered and Solutions:** No challenges faced in this task.

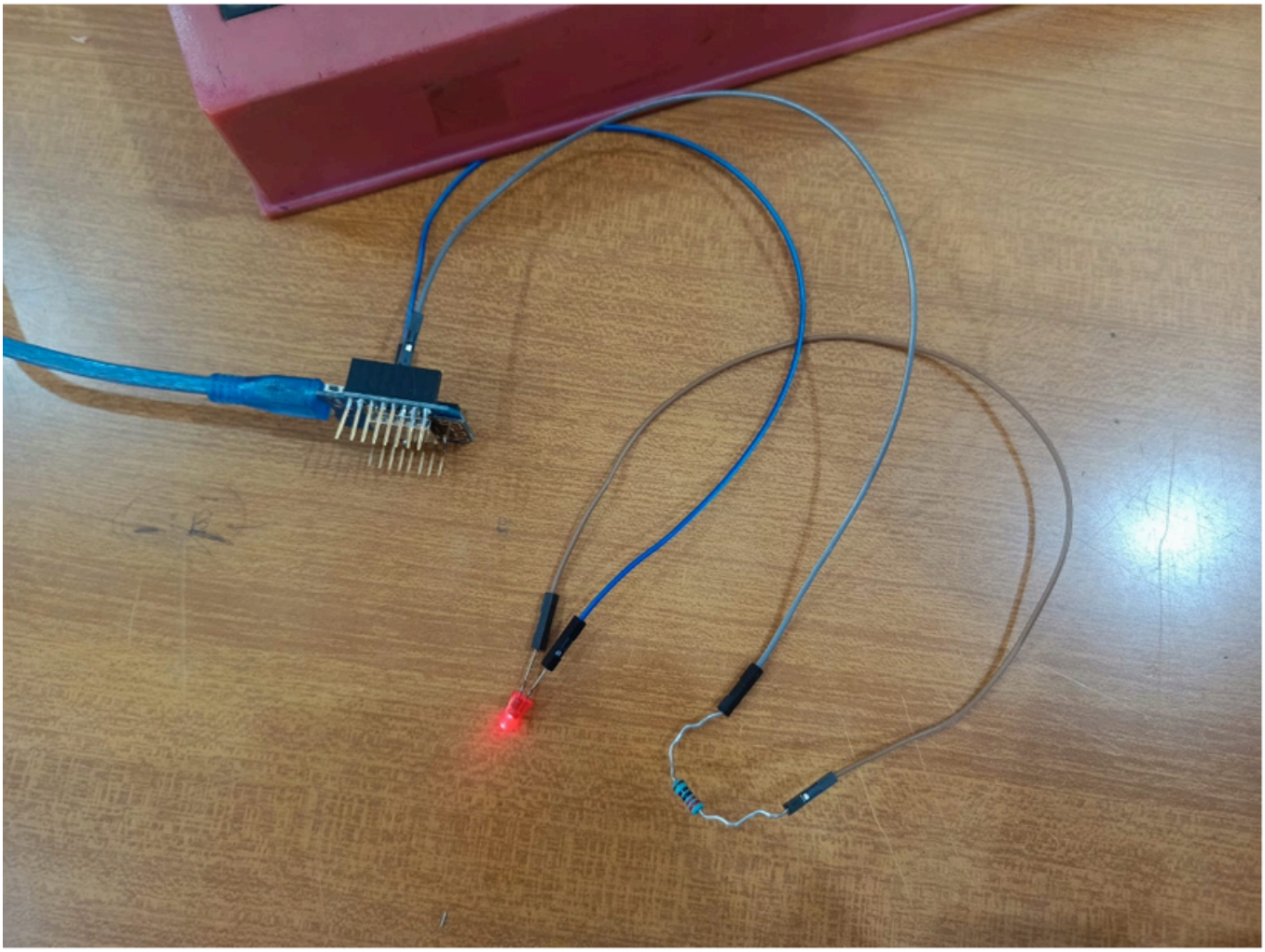
**Suggestions for Improvement:**
1. Instead of manually toggling the relay, integrating WiFi control using platforms like Blynk or MQTT would make the system smarter and allow remote operation.
2. Adding a push button or motion sensor could make the relay more versatile, allowing it to respond to real-world triggers rather than just timed commands.
3. To improve electrical isolation, an optocoupler could be used between the WeMos D1 Mini and the relay. This would protect the microcontroller from potential voltage spikes and enhance overall system reliability.

# Task 2: LED Signalling with PWM

## Arduino Code: (.ino file)



## Circuit Diagram:



## Short Description:

**Purpose and Working of PWM:**

PWM (Pulse Width Modulation) is used to control the brightness of an LED by rapidly switching the signal between HIGH and LOW at different duty cycles. This creates a dimming effect without changing the voltage. In this task, the WeMos D1 Mini gradually increased and decreased the LED's brightness using PWM.

**How the LED Circuit Was Assembled:**

The LED was connected to a PWM-capable digital pin on the WeMos D1 Mini.
A resistor was used to limit current and protect the LED.
The WeMos was powered via USB, and the LED was controlled using Arduino code.

**Challenges Encountered and Solutions:** No challenges faced in this task.

**Suggestions for Improvement:**

- We can use multiple LEDs to create a cool fading effect across different colors, making the project visually appealing.
- We can add a light sensor to make the LED adjust brightness automatically based on the surrounding light—useful for energy-saving applications.
- Or we can integrate WiFi control to adjust brightness remotely using an app, turning it into a smart lighting system.

# Task 3: Dynamic PWM Control using a Potentiometer

## Arduino Code: (.ino file)

## Output: The brightness is changing by rotating the knob of potentiometer

Serial Monitor × Output

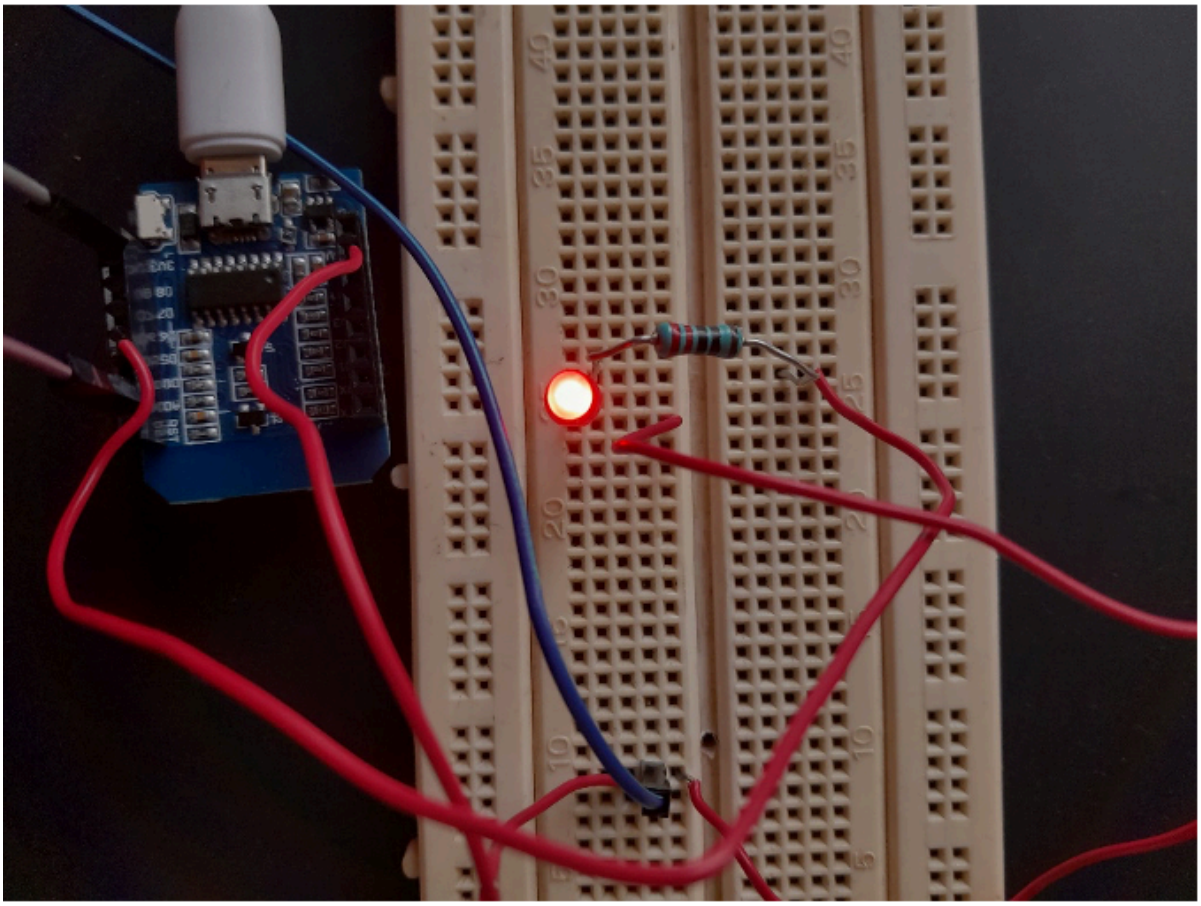
Message (Enter to send message to 'LOLIN(WeMos) D1 R1' on '/dev/cu.usbserial-14110')

```
14:56:18.104 -> Brightness: 255
14:56:18.234 -> Brightness: 255
14:56:18.330 -> Brightness: 252
14:56:18.428 -> Brightness: 207
14:56:18.525 -> Brightness: 200
14:56:18.622 -> Brightness: 200
14:56:18.721 -> Brightness: 200
14:56:18.821 -> Brightness: 200
14:56:18.921 -> Brightness: 199
14:56:19.021 -> Brightness: 175
14:56:19.121 -> Brightness: 126
14:56:19.220 -> Brightness: 76
14:56:19.319 -> Brightness: 26
14:56:19.419 -> Brightness: 3
14:56:19.518 -> Brightness: 4
14:56:19.614 -> Brightness: 4
14:56:19.744 -> Brightness: 3
14:56:19.843 -> Brightness: 4
14:56:19.942 -> Brightness: 4
14:56:20.039 -> Brightness: 4
```

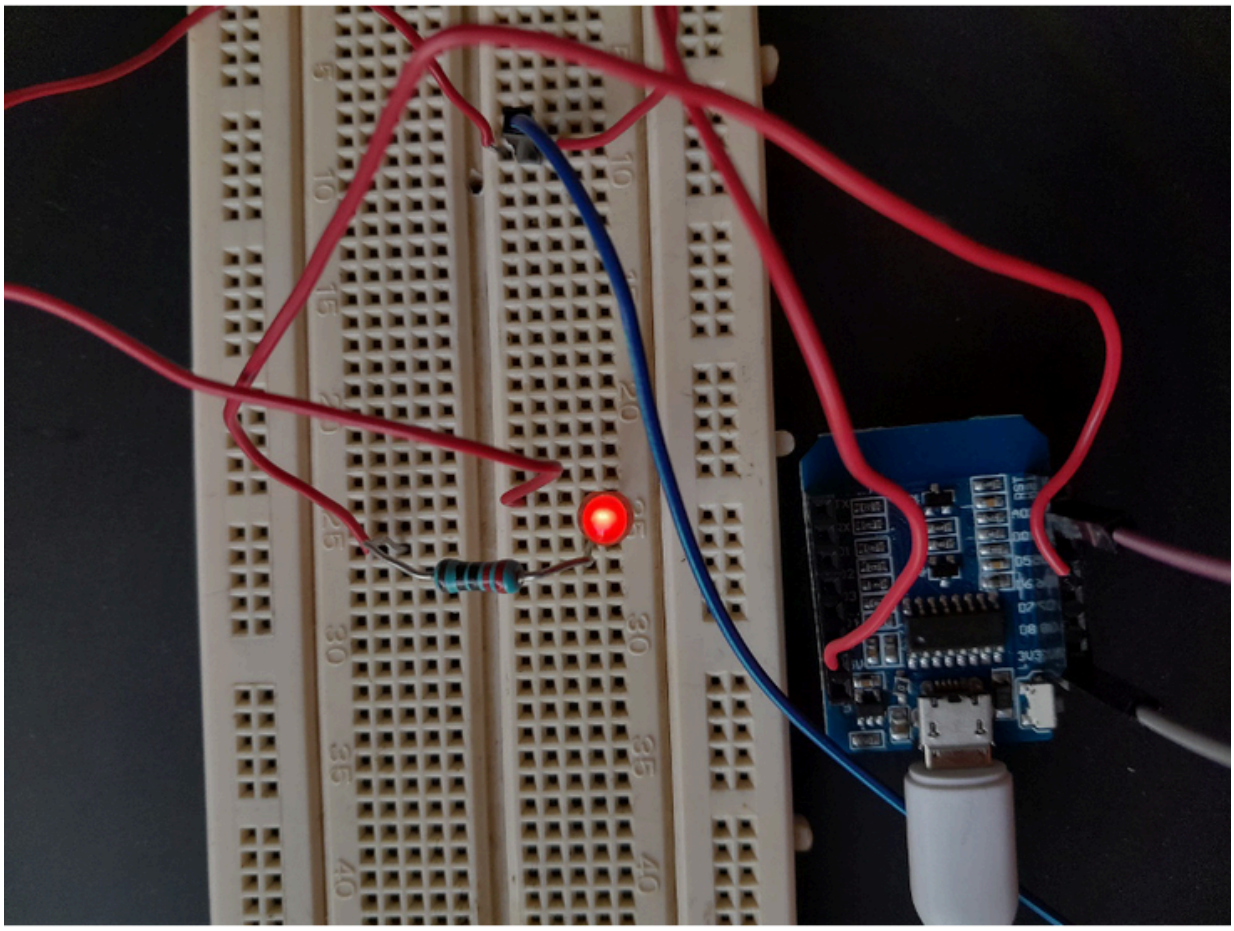
lexing: 54/87

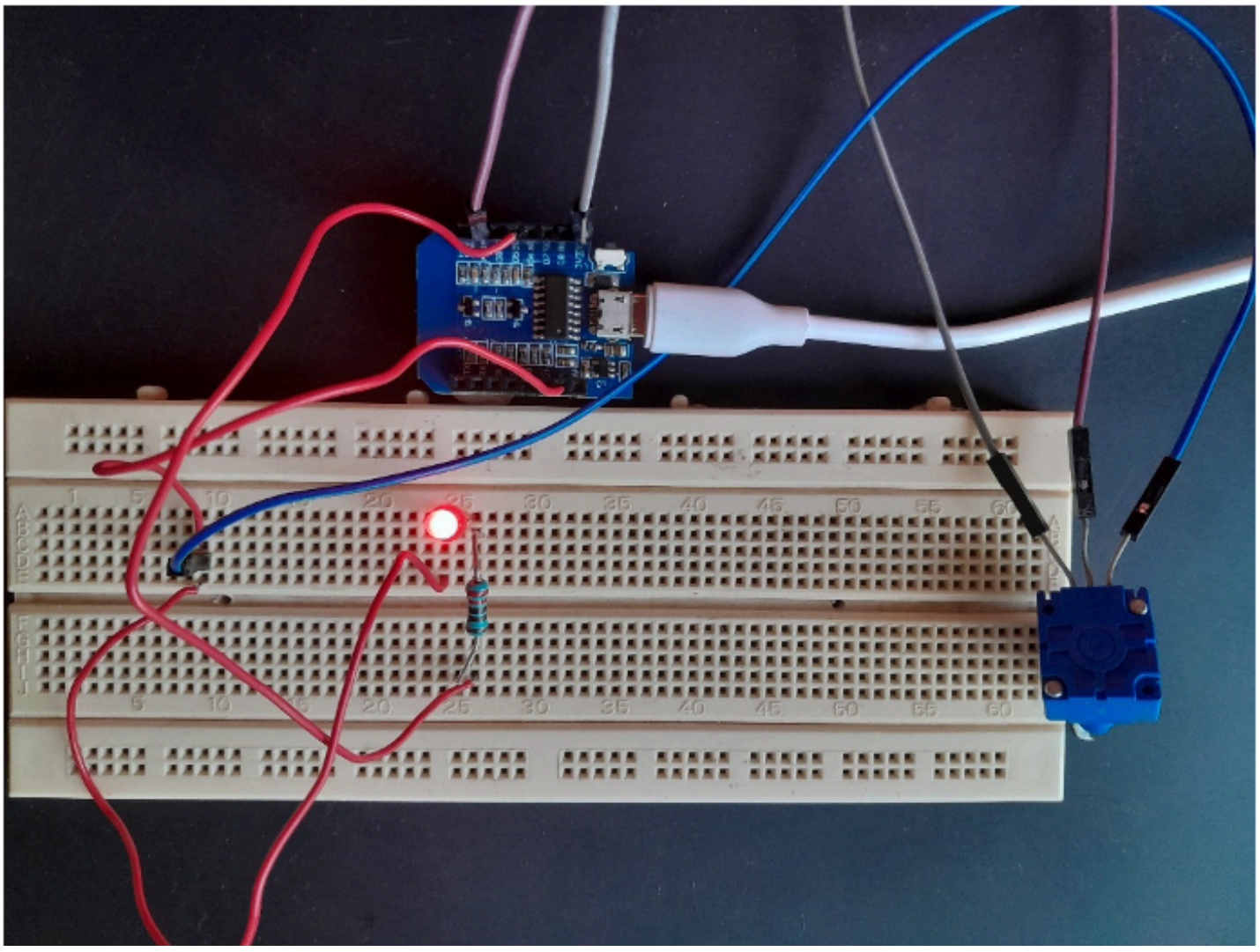
## Circuit Diagram:

**HIGH BRIGHTNESS**



# LOW BRIGHTNESS





# Short Description:

**Purpose and Working of PWM:**
In this lab, we worked on controlling the brightness of an LED using a potentiometer and PWM (Pulse Width Modulation) signal from the ESP8266. The potentiometer works like a variable resistor — when we rotated its knob, the voltage on its middle pin changed. This changing voltage is sent to the A0 pin of the ESP8266, where the microcontroller reads it using the analogRead() function.
Once the ESP knows this voltage, we used the map() function to convert that voltage reading into a brightness value between 0 and 255. This brightness value controlled how much time the LED stayed ON or OFF using PWM. In short, the more we turn the potentiometer, the brighter the LED glows.

**How the Circuit Was Assembled**

- One side of the potentiometer was connected to 3.3V.
- Other side was connected to GND.
- The middle pin (wiper) was connected to A0 pin of ESP8266.
- The LED was connected to D5 pin through a resistor to protect it from too much current.

**Challenges and Solutions:** No challenges faced in this task.

**Suggestions for Improvement (Optional)**

- Instead of directly adjusting LED brightness, we could smooth the potentiometer reading using averaging to reduce sudden flickers.
- This same setup could be used to control motor speed, servo position, or even fan speed.

**END**