Practical IoT (Internet of Things)

BSCIS – DCIS, PIEAS

Lab 03: Actuator Control

Objective

- ***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

Introduction

Actuator control is a fundamental aspect of IoT, enabling microcontrollers to interact with the physical world. In this lab:

- Relays will be used to control high-power devices (e.g., turning a lamp on/off).
- LEDs serve as indicators and can also demonstrate variable brightness using PWM, illustrating how analog behavior is achieved with digital signals.

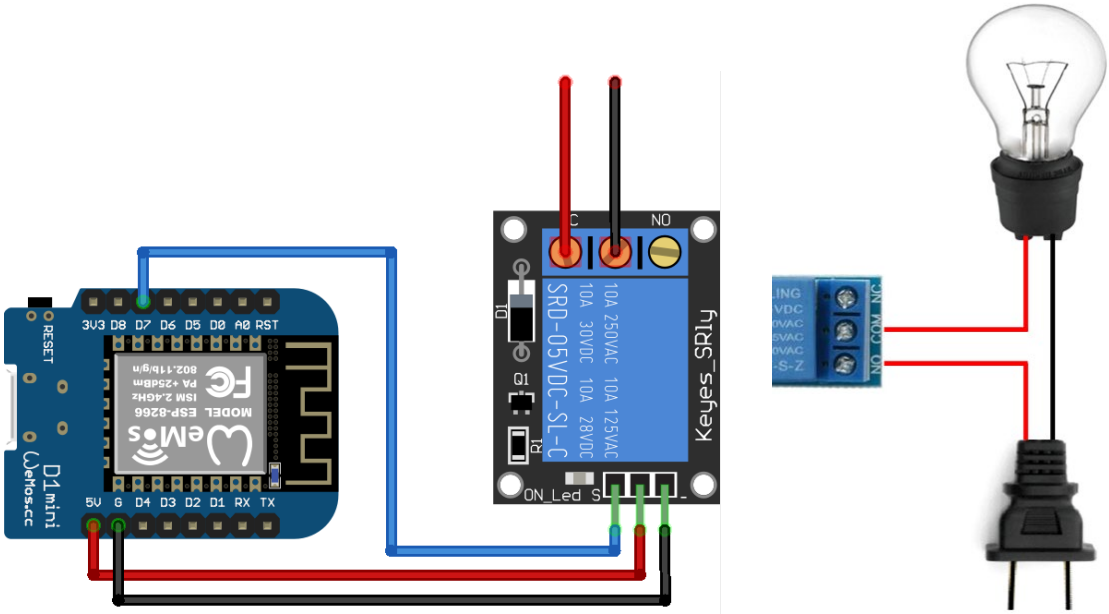
Part 1: Relay Control for Switching Appliances

Circuit Diagram:

- Relay Module Wiring:
    - VCC: Connect to the WeMos D1 Mini's 5V pin (if your relay module supports 3.3V, use that instead).
    - GND: Connect to Ground.
    - IN (Control Signal): Connect to a digital pin on the WeMos D1 Mini (e.g., D2).
- Appliance Side (Simulated):
    - For safety, you may use a simple LED (through a separate circuit) to simulate switching.

- Note: In a real application, the relay's contacts would be used to switch mains voltage (ensure proper isolation and safety practices).

Please note that We will be using 12V instead of AC Mains for your safety.



Sample Code: Relay Control

```
#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);
}
```

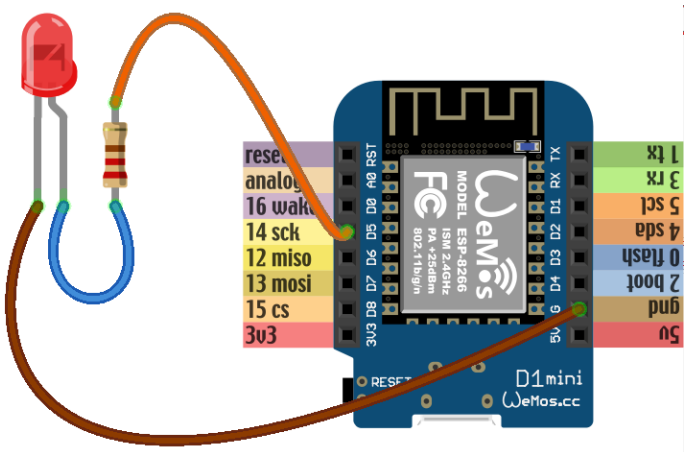
Explanation:

- The relay is controlled using a simple digital output.
- The code toggles the relay every 5 seconds, which can be observed via the Serial Monitor.

Part 2: LED Signaling with PWM

Circuit Diagram

- LED Wiring:
  - Anode (+): Connect to a PWM-capable digital pin (e.g., D5) through a 220Ω resistor.
  - Cathode (-): Connect to Ground.
- Optional (PWM Control with Potentiometer):
  - Connect a potentiometer to an analog input (e.g., A0) to adjust the brightness interactively.



Sample Code: LED PWM Control

```
#define LED_PIN D5  // LED connected to PWM-capable pin D5

void setup() {
  pinMode(LED_PIN, OUTPUT);
  Serial.begin(115200);
  Serial.println("PWM LED Control Initialized");
}

void loop() {
  // Ramp brightness up
  for (int brightness = 0; brightness <= 255; brightness++) {
    analogWrite(LED_PIN, brightness);
    delay(10); // Short delay for smooth ramping
  }
  // Ramp brightness down
  for (int brightness = 255; brightness >= 0; brightness--) {
    analogWrite(LED_PIN, brightness);
    delay(10);
  }
}
```

Explanation:

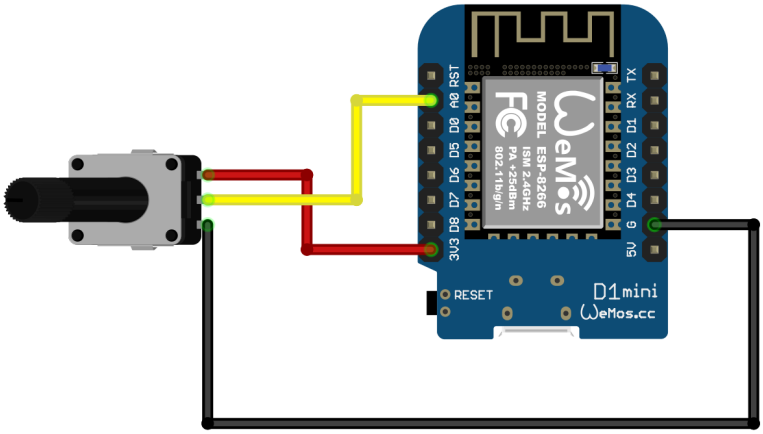
- PWM (Pulse Width Modulation) is used to vary the brightness of the LED.
- The analogWrite function changes the duty cycle on pin D5, simulating analog output with a digital signal.
- The LED brightness ramps up and then down, demonstrating smooth transitions.

Part 3: Advanced Task (Optional)

Dynamic PWM Control using a Potentiometer:

- ***Circuit:*** Connect one end of the potentiometer to 3.3V, the other to GND, and the wiper (middle pin) to the analog input (A0) on the WeMos D1 Mini.
- ***Code Modification:*** Read the potentiometer value using analogRead(A0) and map it to a PWM range (0–255) for the LED.

Please Note that Potentiometer Connection is Additioal to the LED connection done in previous part.



Example Code Snippet:

```
#define LED_PIN D5
#define POT_PIN A0

void setup() {
  pinMode(LED_PIN, OUTPUT);
  Serial.begin(115200);
}

void loop() {
  int potValue = analogRead(POT_PIN);
  // Map the potentiometer value (0-1023) to PWM range (0-255)
  int brightness = map(potValue, 0, 1023, 0, 255);
  analogWrite(LED_PIN, brightness);
  Serial.print("Brightness: ");
  Serial.println(brightness);
  delay(50);
}
```

Explanation:

This code reads the analog value from the potentiometer and maps it to a PWM output, allowing the user to adjust the LED brightness interactively.

Submission Requirements:

Submit a single PDF containing:

- **Your Arduino Code:** Submit Arduino sketches (.ino) for both the relay control and the PWM LED control (with optional potentiometer integration, if implemented).
- **Circuit Diagram:** Provide clear schematic diagrams (or photos) of your breadboard setup.
- **Short Description:**
    - The purpose and working of relays and PWM.
    - How the relay and LED circuits were assembled.
    - Challenges encountered and how they were addressed.
    - (Optional) Suggestions for further improvements or applications.

Conclusion:

- This lab demonstrates fundamental actuator control using a WeMos D1 Mini.
- Students gain hands-on experience with relays for switching and PWM for LED dimming.
- The optional potentiometer integration adds an extra layer of interactivity, showcasing the dynamic nature of IoT applications.

Grading Rubric:

Criterion	Points	Description
Relay Control Circuit	30	Accurate wiring and functional relay control, including clear schematic.
PWM LED Control	30	Effective use of PWM to control LED brightness, with smooth transitions.
Code Quality & Documentation	20	Well-commented code, clear explanations of digital vs. analog output.
Optional Tasks	20	Additional functionality (e.g., potentiometer integration for dynamic PWM) if implemented.
Total	100	