Wireless Power Distribution Unit (PDU) Documentation

## **Table of Contents**

1. Introduction

2. Components Required

3. Circuit Diagram

4. Assembly Instructions

5. Arduino Code

6. How It Works

7. Web Interface

8. Troubleshooting

9. Future Enhancements

### 1. Introduction

The Wireless Power Distribution Unit (PDU) project involves creating a system to control multiple electrical appliances remotely using a web interface. This is achieved using a NodeMCU (ESP8266), a 4x4 relay module, and an LCD screen. The NodeMCU connects to a WiFi network, allowing users to control the relays through a web browser.

### **2. Components Required**

- NodeMCU (ESP8266)

- 4x4 Relay Module

- 16x2 LCD Screen with I2C interface

- Jumper Wires

- Breadboard

- Power Supply for the Relay Module (if required)

### 3. Circuit Diagram

Connections:

1. Relay Module:

- VCC to 3.3V on the NodeMCU

- GND to GND on the NodeMCU

- IN1 to D1 on the NodeMCU

- IN2 to D2 on the NodeMCU

- IN3 to D3 on the NodeMCU

- IN4 to D4 on the NodeMCU

2. LCD Screen:

- VCC to 3.3V on the NodeMCU

- GND to GND on the NodeMCU

- SDA to D2 on the NodeMCU

- SCL to D1 on the NodeMCU

#### **4. Assembly Instructions**

1. Connect the relay module to the NodeMCU as per the connections listed above.
2. Connect the LCD screen to the NodeMCU.
3. Ensure all connections are secure and powered correctly.

##### **5. Arduino Code**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

// WiFi credentials

const char\* ssid = "your\_SSID";

const char\* password = "your\_PASSWORD";

// Create an instance of the server

ESP8266WebServer server(80);

// Initialize the LCD display

LiquidCrystal\_I2C lcd(0x27, 16, 2);

// Relay pin definitions

const int relayPins[] = {D1, D2, D3, D4};

bool relayStates[] = {LOW, LOW, LOW, LOW}; // Initial states for the relays

void setup() {

  // Initialize the LCD

  lcd.begin();

  lcd.backlight();

  lcd.print("Wireless PDU");

  // Initialize the relay pins

  for (int i = 0; i < 4; i++) {

    pinMode(relayPins[i], OUTPUT);

    digitalWrite(relayPins[i], relayStates[i]);

  }

  // Connect to WiFi

  WiFi.begin(ssid, password);

  lcd.setCursor(0, 1);

  lcd.print("Connecting...");

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

    lcd.print(".");

  }

  lcd.clear();

  lcd.print("Connected");

  lcd.setCursor(0, 1);

  lcd.print(WiFi.localIP());

  // Define server routes

  server.on("/", handleRoot);

  for (int i = 0; i < 4; i++) {

    server.on(String("/relay" + String(i+1) + "/on").c\_str(), [i]() { handleRelay(i, HIGH); });

    server.on(String("/relay" + String(i+1) + "/off").c\_str(), [i]() { handleRelay(i, LOW); });

  }

  // Start the server

  server.begin();

}

void loop() {

  // Handle client requests

  server.handleClient();

}

// Handle the root URL

void handleRoot() {

  String html = "<html>\

  <head>\

    <title>Wireless PDU</title>\

  </head>\

  <body>\

    <h1>Wireless PDU Control</h1>";

  for (int i = 0; i < 4; i++) {

    html += "<p>Relay " + String(i+1) + ": <a href=\"/relay" + String(i+1) + "/on\">ON</a> | <a href=\"/relay" + String(i+1) + "/off\">OFF</a></p>";

  }

  html += "</body>\

  </html>";

  server.send(200, "text/html", html);

}

// Handle relay control

void handleRelay(int relay, int state) {

  relayStates[relay] = state;

  digitalWrite(relayPins[relay], state);

  server.sendHeader("Location", "/");

  server.send(303);

}

**Steps to Upload Code:**

1. Open the Arduino IDE.

2. Install the required libraries:

- LiquidCrystal\_I2C

- ESP8266WiFi

- ESP8266WebServer

3. Replace `your\_SSID` and `your\_PASSWORD` with your WiFi credentials.

4. Select the appropriate board (NodeMCU 1.0) and port.

5. Upload the code to the NodeMCU.

##### **6. How It Works:**

* Initialization
  + The LCD display shows "Wireless PDU."
  + The relay pins are set to their initial states (OFF).
  + The NodeMCU connects to the specified WiFi network and displays the IP address on the LCD.
* Web Server
  + The root URL ("/") serves an HTML page with links to control each relay (turn ON or OFF).
  + Separate routes are defined for turning each relay ON ("/relayX/on") and OFF ("/relayX/off").
* Relay Control
  + When a relay control URL is accessed, the corresponding relay is switched ON or OFF, and the web page is refreshed.

##### **7. Web Interface**

1. Open a web browser and enter the IP address displayed on the LCD.

2. You will see a control panel with buttons to turn each relay ON or OFF.

#### **8. Troubleshooting**

* No WiFi Connection:
* Ensure the WiFi credentials are correct.
* Check if the WiFi network is operational.
  + Relays Not Switching:
  + Check the connections between the NodeMCU and the relay module.
  + Ensure the relay module is powered correctly.
    - LCD Not Displaying Correctly:
    - Verify the connections between the NodeMCU and the LCD.
    - Ensure the I2C address is correct.

#### **9. Future Enhancements**

* Add more relays to control additional devices.
* Implement a user authentication system for secure access.
* Add functionality to control relays using voice commands or mobile applications.
* Integrate with home automation systems like Home Assistant.

---

This documentation provides a comprehensive guide to creating a wireless PDU using a NodeMCU, a relay module, and an LCD screen. By following these instructions, you can successfully build and operate the system.

DOCUMENTATION BY

SHIVARAJ HIREMATH AND MALIKA NAUSHAD