Comprehensive PDF Report

1.Project : ESP32 Web Control with NeoPixel, OLED, and DHT11

22 -ntu-cs-1335

2. Table of Contents

- List of sections with page numbers.

3. Introduction

- Brief overview of the project.
- Objectives of the project.
- Key components used:
  - ESP32 microcontroller.
  - NeoPixel LED.
  - OLED display (SSD1306).
  - DHT11 temperature and humidity sensor.

4. Hardware Setup

- List of components:
  - ESP32 microcontroller.
  - NeoPixel LED.
  - OLED display (SSD1306).
  - DHT11 sensor.
  - Breadboard and jumper wires.
- Wiring diagram:
  - NeoPixel: Connected to GPIO 48.
  - OLED: SCL to GPIO 9, SDA to GPIO 8.

  - DHT11: Connected to GPIO 4.
- Pin configurations:
  - NeoPixel: Pin 48.
  - OLED: SCL = Pin 9, SDA = Pin 8.
  - DHT11: Pin 4.

5. Software Setup
- Software environment: MicroPython.
- Libraries used:
  - `network`: For WiFi connectivity.
  - `socket`: For web server functionality.
  - `neopixel`: For controlling the NeoPixel.
  - `machine`: For GPIO and I2C.
  - `dht`: For reading DHT11 sensor data.
  - `ssd1306`: For OLED display.
- Installation instructions:
  - Install MicroPython on ESP32.
  - Upload necessary libraries.

6. Code Explanation
- WiFi Configuration
  ```python
  SSID = "NTU FSD"
  PASSWORD = ""
  sta = network.WLAN(network.STA_IF)
  sta.active(True)
  sta.connect(SSID, PASSWORD)
  ```

```
- NeoPixel Control
  ```python
  def set_neopixel(r, g, b):
      np[0] = (r, g, b)
      np.write()
  ```
- OLED Display
  ```python
  def display_text(text):
      oled.fill(0)
      oled.text(text, 0, 10)
      oled.show()
  ```
- DHT11 Sensor
  ```python
  def get_sensor_data():
      dht_sensor.measure()
      temp = dht_sensor.temperature()
      hum = dht_sensor.humidity()
      return temp, hum
  ```
- Web Server
  - The `web_page` function generates an HTML page with interactive controls.

7. Web Interface
   - Features:
  ```

    - RGB sliders for NeoPixel control.
    - Temperature and humidity gauges.
    - Text input field for OLED display.
  - Design:
    - HTML, CSS, and JavaScript for interactivity.
  - Screenshots:
    - Include images of the web interface.

8. Testing and Results
  - Testing steps:
    - Verify WiFi connectivity.
    - Test NeoPixel color changes.
    - Display text on the OLED.
    - Read and display sensor data.
  - Sample outputs:
    - NeoPixel colors.
    - OLED text.
    - Sensor readings.
  - Challenges:
    - Sensor errors.
    - WiFi disconnections.
  - Solutions:
    - Error handling in code.
    - Stable power supply.

9. Conclusion
  - Summary:

- Successful integration of ESP32 with NeoPixel, OLED, and DHT11.
- Web interface for control and monitoring.

- Learnings:
- Web server implementation.
- Sensor integration.

- Future improvements:
- Add more sensors.
- Enhance web interface.

10. Appendix

- Full code listing.
- References:
- Datasheets for ESP32, NeoPixel, OLED, and DHT11.
- Tutorials or guides used.
- Links to libraries and tools.

- Integration of ESP32 with NeoPixel, OLED, and DHT11.
- Web-based control and monitoring.

- Objectives:
- Control NeoPixel color via web interface.
- Display text on OLED.
- Monitor temperature and humidity.

Hardware Setup

- Components:
- ESP32 microcontroller.
- NeoPixel LED.

  - OLED display (SSD1306).
  - DHT11 sensor.
  - Breadboard and jumper wires.
- Wiring:
  - NeoPixel: GPIO 48.
  - OLED: SCL = GPIO 9, SDA = GPIO 8.
  - DHT11: GPIO 4.

Code Explanation
- **WiFi Configuration**:
  - Connects ESP32 to WiFi.
- **NeoPixel Control**:
  - Sets NeoPixel color using RGB values.
- **OLED Display**:
  - Displays text on OLED.
- **DHT11 Sensor**:
  - Reads temperature and humidity.
- **Web Server**:
  - Generates HTML page with controls.

Web Interface
- Features:
  - RGB sliders for NeoPixel.
  - Temperature and humidity gauges.
  - Text input for OLED.
- Design:
  - HTML, CSS, and JavaScript.

Testing and Result

- Testing:
  - WiFi connectivity.
  - NeoPixel color changes.
  - OLED text display.
  - Sensor data reading.