

LAB2: IoT Webserver with LED, Sensors, and LCD Control

1. Overview

In this lab, students will design an ESP32-based IoT system with MicroPython that integrates a web interface and an LCD display. The system will allow users to control an LED, read sensors, and send custom messages to the LCD through a webserver.

This lab emphasizes interaction between web UI and hardware, giving students practice in event-driven IoT design.

2. Learning Outcomes (CLO Alignment)

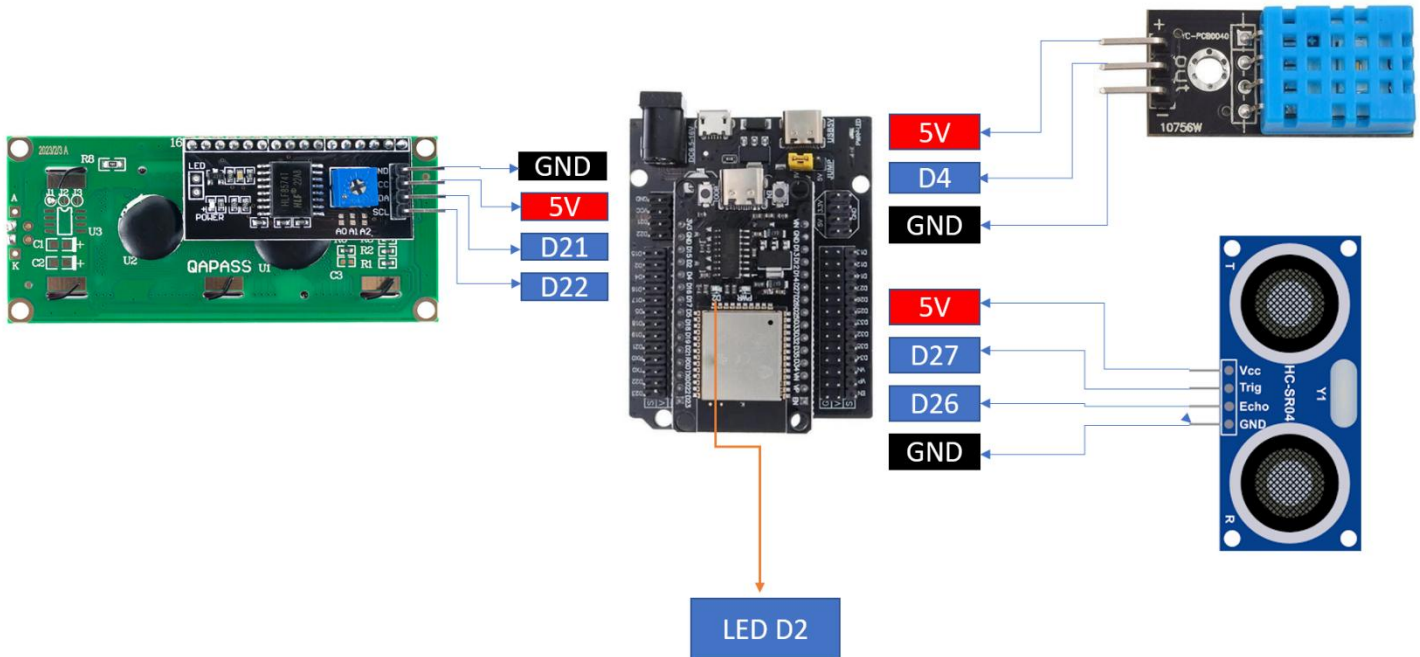
By the end of this lab, students will be able to:

- Implement a MicroPython webserver to serve HTML controls.
- Control an LED from the web page.
- Read data from DHT11 and ultrasonic sensors and expose it on the webserver.
- Use web buttons to selectively show temperature and distance on an LCD (I²C).
- Send custom text from a textbox to display on the LCD.
- Document wiring, interface behavior, and system operation.

3. Equipment

- ESP32 Dev Board (MicroPython firmware flashed)
- DHT11 sensor (temperature/humidity)
- HC-SR04 ultrasonic distance sensor
- LCD 16×2 with I²C backpack
- Breadboard, jumper wires
- USB cable + laptop with Thonny
- Wi-Fi access

4. Wiring



5. Tasks & Checkpoints

Task 1 - LED Control (15 pts)

- Add two buttons (ON/OFF) on the web page.
- When clicked, LED on GPIO2 should turn ON or OFF.
- Evidence: short video showing button click → LED changes.

Task 2 - Sensor Read (15 pts)

- Read **DHT11** temperature and **ultrasonic distance**.
- Show values on the web page (refresh every 1-2 seconds).
- Evidence: screenshot of web page with sensor values.

Task 3 - Sensor → LCD (20 pts)

- Add two buttons:

- **Show Distance** → writes distance to LCD line 1.
- **Show Temp** → writes temperature to LCD line 2.
- Evidence: photo of LCD showing correct sensor values after button clicks.

Task 4 - Textbox → LCD (20 pts)

- Add a textbox + “Send” button on the web page.
- User enters custom text → LCD displays it (scroll if >16 chars).
- Evidence: short video of typing text in browser → appears on LCD.

Task 5 - Documentation (30 pts)

- README.md with:
 - Wiring diagram/photo
 - Setup instructions (Wi-Fi, running server)
 - Usage instructions (LED control, sensor buttons, textbox → LCD)
- Evidence: GitHub repo with source code, screenshots, and demo video.

6. Submission & Academic Integrity

Submit a private GitHub repo (add instructor as collaborator). Include:

- Source code (main.py + LCD helper files)
- README.md with wiring diagram/photo and usage instructions
- Screenshots of web page + LCD output
- Short demo video (60–90s) showing:
 - LED ON/OFF from browser
 - Temperature/distance displayed on LCD via buttons
 - Text typed into browser textbox displayed on LCD