Project Report:
Plant Monitor
University of Basel
Computer Architecture (62025-1)
Anton Bilokon

Basel - 2026

Abstract

This project presents an automated plant monitoring and irrigation system designed for indoor plant care. The system integrates an ESP32 microcontroller with environmental sensors (SHT31 for temperature and humidity, capacitive soil moisture sensor), an SSD1306 OLED display, and a water pump controlled via relay. In this project it is successfully implemented continuous monitoring of environmental conditions with local display, automated irrigation based on soil moisture thresholds, and remote control capabilities through Discord bot integration using MQTT protocol. The system operates reliably with configurable moisture thresholds (dry <20%, moist 20-60%, wet >60%) and includes safety features such as pump cooldown periods to prevent overwatering.

---

1. Introduction

This project presents an automated plant monitor that is able to irrigate the indoor plants. The system has a soil moisture sensor that serves as an information source for the added water pump. Which in combination provides a possibility to provide plant care when the soil gets dry. There are remote capabilities presented. The communication between controller and discord bot enables users to effectively turn on/off pump service, see the status message and exercise manual watering service.

Key features include continuous monitoring of temperature, humidity, and soil moisture levels, with data visualization on a local OLED display

1.1 Contributions

It is claimed to be achieved the following:

- **OLED driver:** Developed a low-level driver (oled_ssd1306.c/h) for the SSD1306 chip with I2C communication, pixel-level control, and ASCII rendering capabilities
- **Sensor integration:** SHT31 digital sensor and analog capacitive soil moisture sensor was integrated with proper calibration
- **Irrigation logic:** Implemented threshold-based automatic watering with safety features including 3-second pump activation and 10-second cooldown periods
- **Remote control system:** Established MQTT-based communication enabling Discord bot commands (!water, !pump_on, !pump_off, !status, !plant) for remote monitoring and control
- **Local display:** Continuous updating of sensor readings on OLED screen for immediate visual feedback
- **Modular software design:** Organized code structure separating concerns (main logic, display driver, connectivity) for maintainability and future expansion

2. Methodology and System Design

2.1 Hardware Architecture

The main logical unit of the system is ESP32 microcontroller, the choice of the microcontroller motivated by the financial factors, ability to provide WIFI compatibility.

Sensors:

- SHT31 sensor provides digital temperature and humidity readings via I2C (SDA: GPIO21, SCL: GPIO19)
- Capacitive soil moisture sensor outputs analog values read through ADC pin GPIO34, calibrated with air reference (4095) and water reference (2200)

Output and Control:

- SSD1306 OLED (128x64) connected via I2C displays sensor data locally
- 2.5-6V micro water pump controlled through GPIO5-connected relay module
- Power supply (5V for pump/sensors, 3.3V for ESP32)

2.2 Software Architecture

The software follows a modular design pattern:

Main Application (main.cpp): Contains the core control loop, sensor reading functions, pump logic, and coordination between modules. Implements both automatic (threshold-based) and manual (command-triggered) watering modes.

Display Driver (oled_ssd1306.c/h): Low-level driver directly interfaces with SSD1306 chip registers. Provides initialization routines, pixel manipulation, and text rendering using a 5x7 ASCII font.

Connection Module (connectToWifi.cpp/h): Manages WiFi connection establishment and MQTT protocol implementation for the communication with the Discord bot server. Implements callback functions to handle incoming commands.

3. Implementation

3.1 OLED Driver Development

The main challenge when developing Plant Monitor was to implement information rendering on the OLED display. Successful implementation required studying the SSD1306 datasheet and using it to understand initialization sequences, addressing modes, and pixel mapping, the direct manipulation of the SSD1306 chip. As well as implementation of I2C protocol, which nowadays just used in abstraction (e.g. <Wire.h> library. The major challenges, aside from ensuring correct display circuit setup were:

- Wrong command sequences that stalled preparation of the display for usage.
- Incorrect pixel mapping
- Complex character encoding bitmap

Solution of these challenges substituted a comprehensive understanding of SSD1306 datasheet, as well as numerous initialization tests.

3.2 Sensors Integration

Temp/humidity sensor integration: The digital sensor was integrated with the help of Adafruit SHT31 library that provides some level of abstraction that eases the sensor integration.

Soil Moisture sensor: Due to the capacitive nature of the sensor some test cases had to be adjusted and general understanding that sensor will provide correct data, but it will require some time. While during development such behaviour was confusing, it plays no big role for the end user.

3.3 Watering System Safety

Preventing overwatering required implementing multiple safety mechanisms:

- Cooldown timer: 10 second between the pump activations
- Work duration limiting: 3 seconds of simultaneous pump work, delivering approximately 100 ml of water per cycle.
- Enable/disable pump: initial state *disable,* it is possible to enable and disable automatic pump service, as well as using manual settings.

3.4 Remote Communication

Plant monitor Communication is implemented with the help of MQTT protocol. Firstly, ESP32 establishes the connection to WIFI with the help of <WiFi.h> library and uses <PubSubClient.h> to setup MQTT communication and subscribe to the data channel. Wifi disconnections are handled with continuous reconnections.

4. Results

4.1 Functional Testing

Based on the functional requirements to the project, next results were achieved:

- Continuous monitoring: The system runs reliably for extended periods (tested up to 72 hours continuously), reading sensors and updating the OLED display.

- Automated irrigation: Threshold-based watering activates correctly when soil moisture drops below 20%.

- Remote control: All five Discord commands (!water, !pump_on, !pump_off, !status, !plant) function as specified with response times under 5 seconds.

- Local display: OLED shows clear, readable information including temperature (°C), humidity (%), soil moisture (%), and pump status.

4.3 Match with Original Goals

The implemented system meets all stated objectives from the project proposal:

- Monitor environmental conditions (temperature, humidity, soil moisture)
- Provide visual feedback on local OLED display
- Enable remote monitoring and control
- Automate irrigation based on soil moisture measurements

Additional features implemented beyond original scope include the comprehensive Discord bot command set and detailed status reporting.

5. Conclusions

5.1 Creator

As this was an individual project, all aspects were completed by Anton Bilokon, including hardware assembly, software development (main logic, OLED driver, connectivity module), calibration, testing, and documentation.

5.2 Critical Self-Evaluation

Strengths:

- Modular code design, maintenance, documentation and scalability.
- OLED driver implementation, work done with "low-level" code.
- Safety features (cooldown, duration limits), failure prevention.
- Real world practical use.

What can be improved:

- Error handling, more Logging. For now it takes more time to identify the source of the error, if one occurs in the code. Logging: past sensors data is not stored.
- Physical housing for electronics is absent and can make the whole system inoperable faster.

5.3 Lessons Learned

1. Low-level driver development provided an important understanding of how hardware communicates with one another.
2. Calibration is critical for analog sensors; empirical testing in target conditions is necessary for reliable operation.
3. Safety mechanisms must be thought of at the first stages of project development and fully introduced on the design stage.
4. Modular design, eases the future work and organises projects structurally, useful even for 1 person projects.

5.4 Future Development

- Light intensity monitoring using photoresistor or lux sensor to optimize growing conditions
- Multiple sensor nodes communicating with central controller for larger grow spaces
- Data logging to SD card or cloud storage for trend analysis
- Email/SMS alerts for critical conditions requiring immediate attention
- Protective enclosure for electronics

5.5 AI - Disclosure

This report was developed with the help of AI - tools, for better wording and structure consistency. The original structure of the report was suggested by the lecture evaluators. During the project development AI tools were used exclusively during the extreme debug cases.

References

Libraries:

- Adafruit SHT31 Library: https://github.com/adafruit/Adafruit_SHT31
- PubSubClient (MQTT): https://github.com/knolleary/pubsubclient
- Adafruit GFX Library: https://github.com/adafruit/Adafruit-GFX-Library

Documentation:

- SSD1306 OLED Datasheet: https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf
- SHT31 Sensor Datasheet: https://sensirion.com/media/documents/213E6A3B/63A5A569/Datasheet_SHT3x_DIS.pdf
- ESP32 Technical Reference: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

Hardware Components:

- ESP32
- SSD1306 OLED Display (128x64)
- SHT31 Temperature/Humidity Sensor
- Soil Moisture Sensor v1.2
- 2.5-6V Micro Water Pump
- Relay Module

Project Repository:

- GitHub: https://github.com/spaceTony9/ca_grow_box_project

Resources:

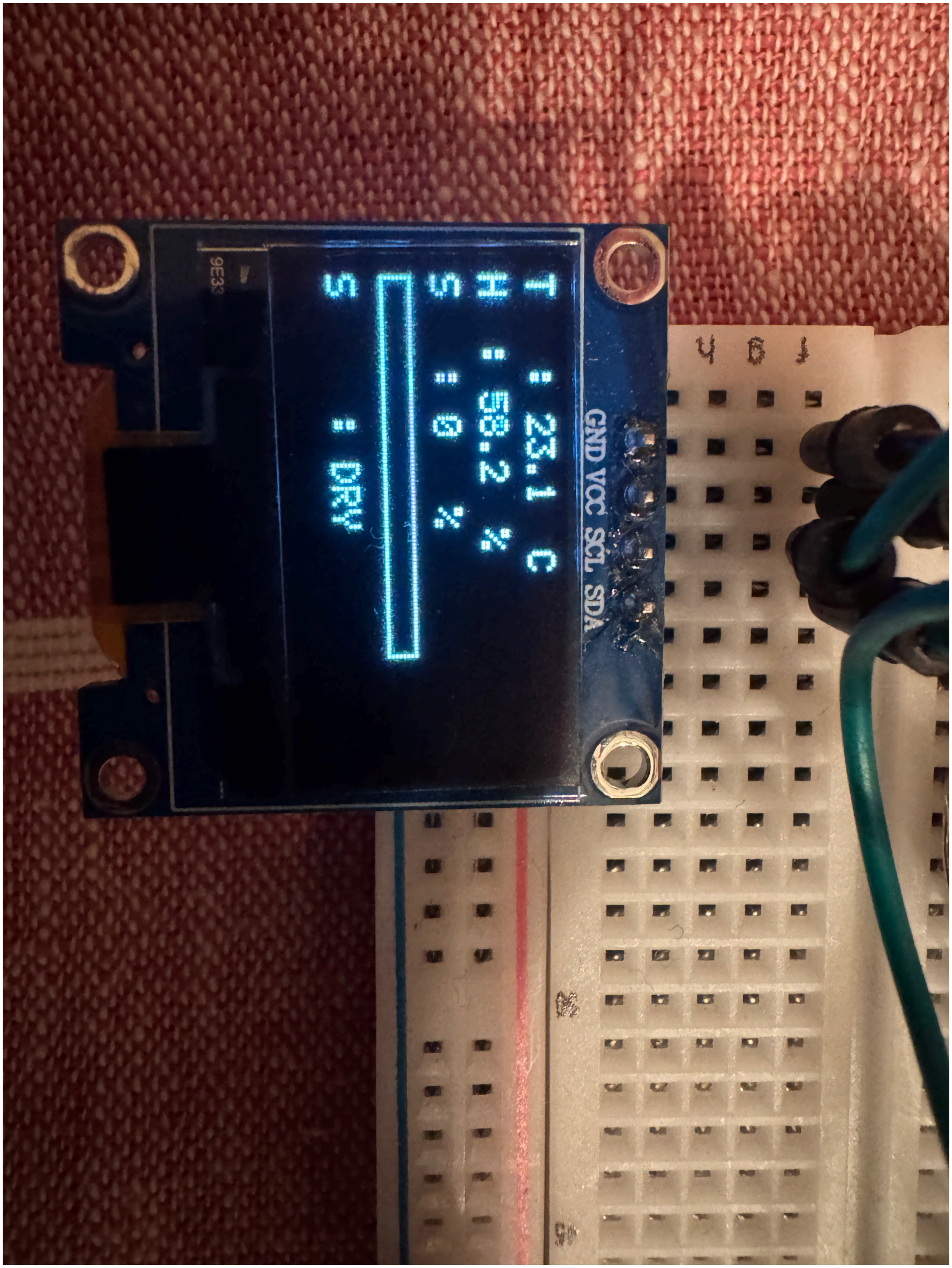
- MQTT Protocol Documentation: https://mqtt.org/
- Discord.py Bot Framework: https://discordpy.readthedocs.io/
- I2C Protocol Overview: https://www.nxp.com/docs/en/user-guide/UM10204.pdf

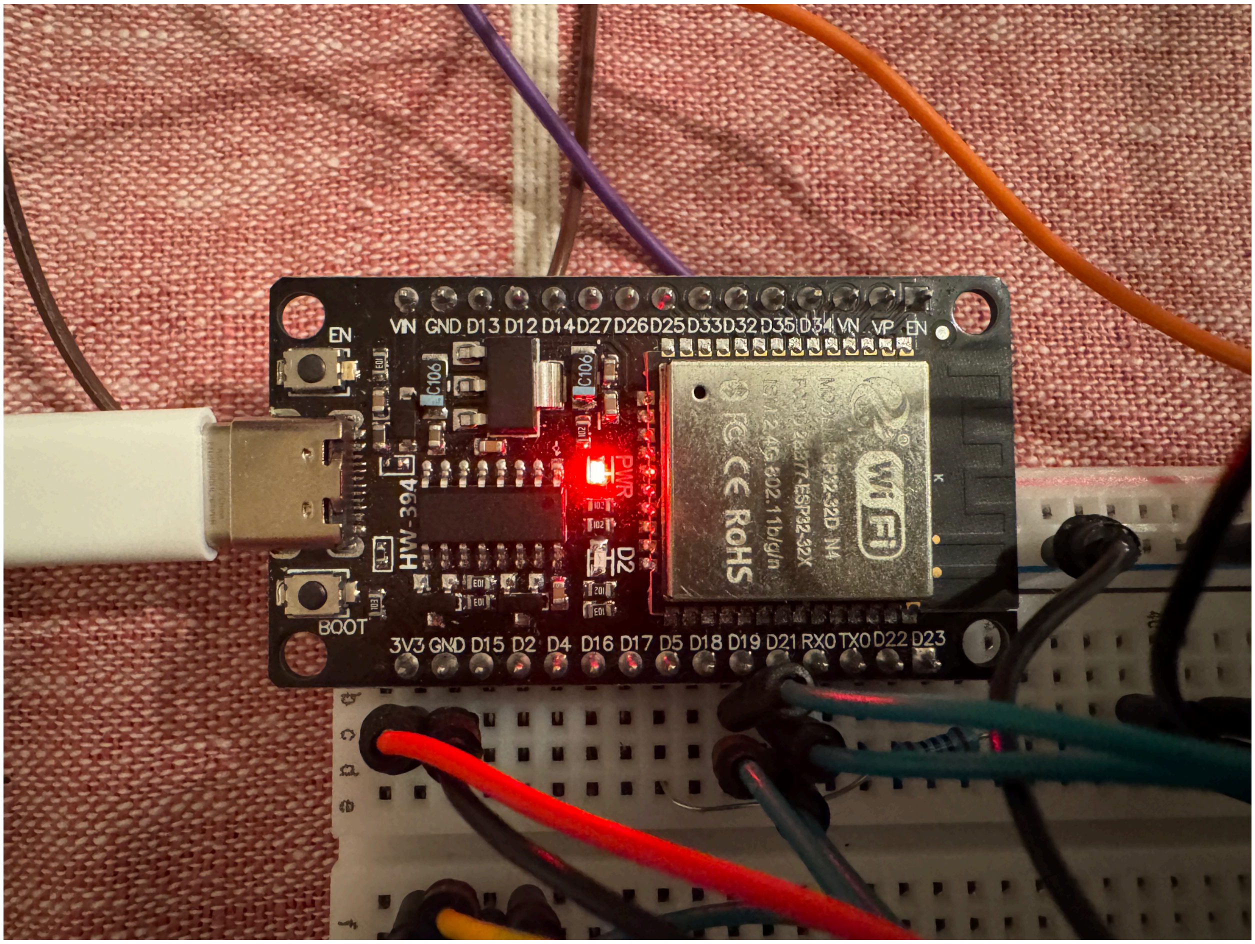
Appendix A: Hardware Connection Diagram

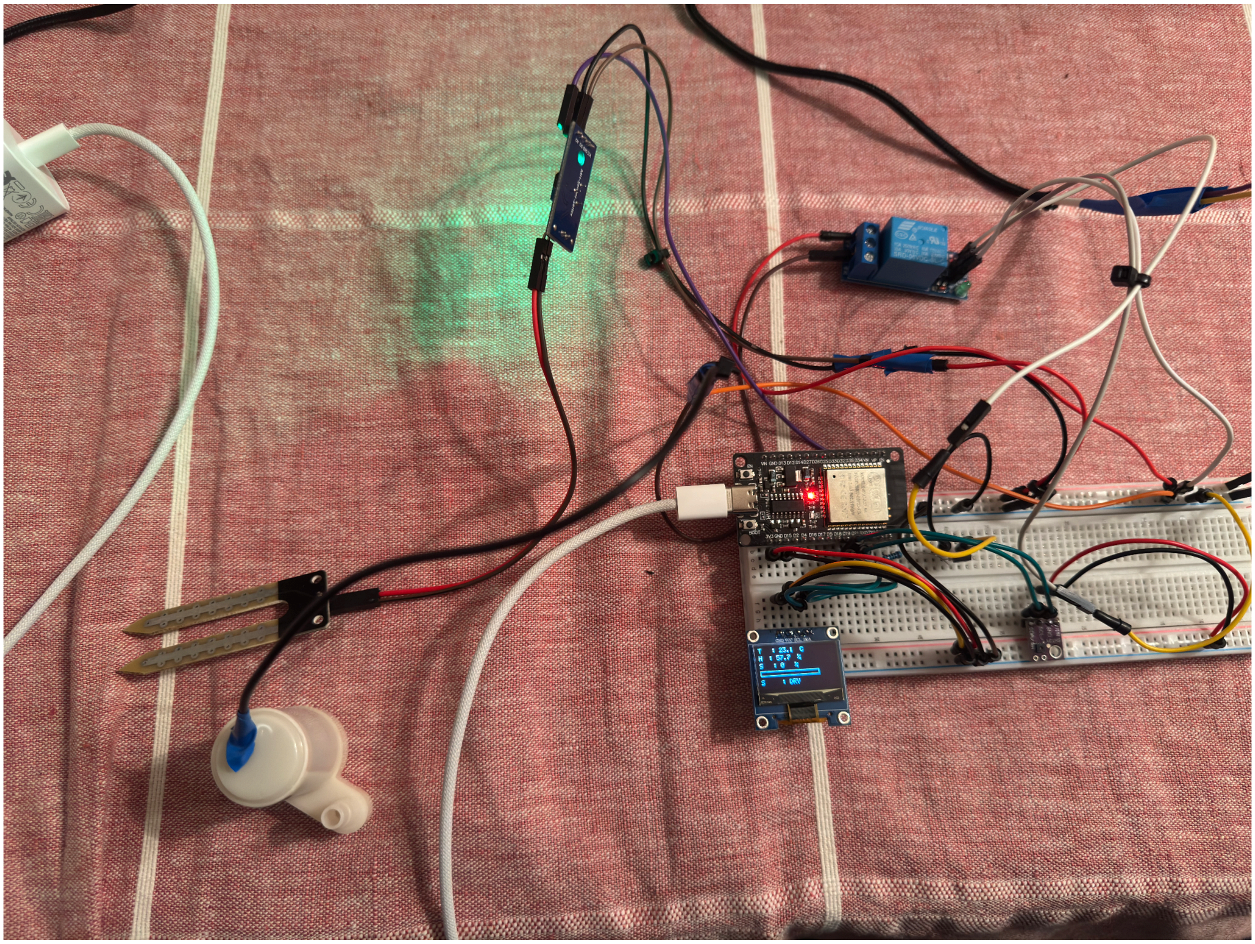
ESP32 Pin Connections:

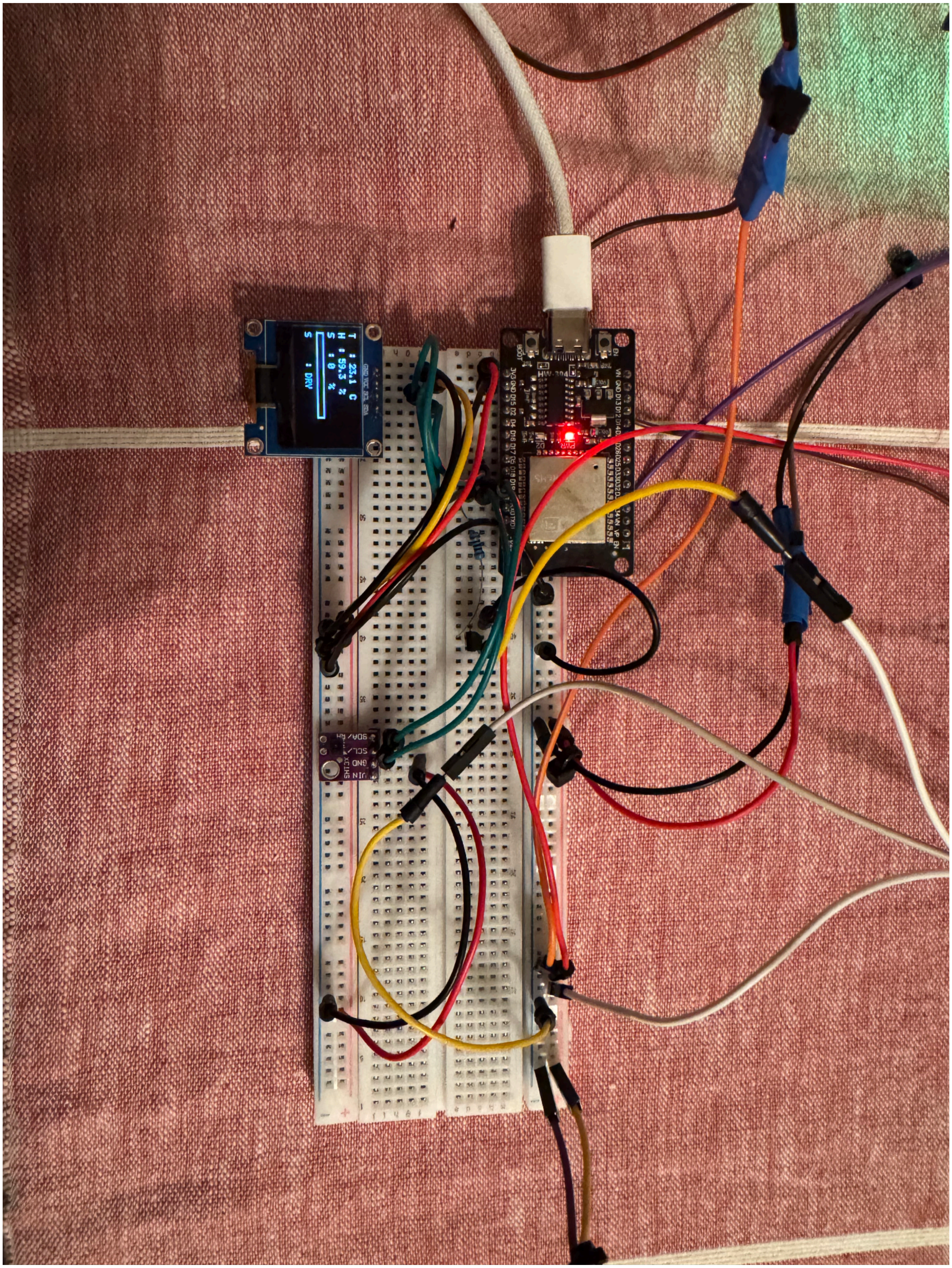
Component	ESP32 Pin	Purpose
SHT31 VIN	3.3V	Power supply
SHT31 GND	GND	Ground
SHT31 SDA	GPIO 21	I2C data
SHT31 SCL	GPIO 19	I2C clock
OLED VCC	3.3V	Power supply
OLED GND	GND	Ground
OLED SDA	GPIO 21	I2C data
OLED SCL	GPIO 19	I2C clock
Soil VCC	5V	Power supply
Soil GND	GND	Ground
Soil AO	GPIO 34	Analog output
Relay Signal	GPIO 5	Pump control
Relay VCC	5V	Power supply
Relay GND	GND	Ground

Appendix C: System Photos


T : 23.1 C
H : 58.2 %
S : 0 %
S : DRY
GND VCC SCL SDA


VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
EN
BOOT
HW-394
PWR
D2
3V3 GND D15 D2 D4 D16 D17 D5 D18 D19 D21 RX0 TX0 D22 D23
RoHS


T : 23.1 C
H : 57.7 %
S : 0 %
S : DRY


T : 23.1 C
H : 59.3 %
S : 0 %
S : DRY
GND VCC SCL SDA
SDA
SCL
GND
VIN