PORTABLE BATTERY-POWERED WEATHER MONITORING SYSTEM

Final Project

Electronic Circuit 2

Group 4 members:

Mirza Ananta Hermawan - 2206822212

Arya Fadhlurrahman Mulia - 2206822010

Farizi Yufli Alrantisi - 2206821595

ELECTRICAL ENGINEERING - FACULTY OF ENGINEERING

UNIVERSITY OF INDONESIA

2024

CHAPTER I: BACKGROUND

A. Background Design

The “Portable Battery-Powered Weather Monitoring System” is designed to provide real time environmental data such as temperature, humidity, and light levels in a portable and battery powered form factor. This device can be useful in many applications including agriculture, where monitoring the environmental conditions can help in managing crop growth efficiently, or in urban areas where such data can contribute to smart city solutions.

This project could also be useful for outdoor event planning, where knowing the weather conditions can help ensure events run smoothly, or even in educational contexts to provide practical experiences with real time data collection and analysis. The portability aspect and easy factor due to it being powered by battery allows the device to be used in remote or changing locations, making it versatile and adaptable to different user needs.

CHAPTER II: DISCUSSION

A. Schematic Design

After multiple attempts at trial and error and after doing our research, for our Portable Battery-Powered Weather Monitoring System project we decided to use these components

1. **ESP32 (bare chip)**: We chose this microcontroller for its Wi-Fi and bluetooth capabilities. ESP32 is also more capable than other microcontrollers like the arduino uno due to its higher processing power and built-in connectivity options.
2. **DSC6083:** This component is needed for ESP32 bare chip. This oscillator offers better stability and reliability compared to traditional quartz oscillators. This oscillator provides a precise clock source for the ESP32, which is crucial for tasks requiring accurate timing such as data logging and time-stamping.

3. **I2C 3 Lines 12 Character LCD:** This display was chosen for its simplicity and effectiveness in showing real-time data through a standardized I2C communication interface, which simplifies the wiring and integration with the ESP32.
4. **CEP-1116 Buzzer:** We use the CEP-1116 because can generate loud enough sounds for audible alerts.
5. **YL-83 (Rain Sensor):** We used YL-83 because it has simple analog output that can easily interface with the ESP32 and its sensitivity to rain detection. This sensor is straightforward to implement compared to more complex sensors like capacitance-based or optical rain sensors, however we could not find the symbol and footprint for this sensor anywhere so we instead made a custom mounting for the sensor.
6. **BH1750 (Ambient Light Sensor):** We use this to provide digital output directly proportional to the lux level of ambient light, making it more accurate and easier to integrate than analog light sensors. This sensor uses I2C for communication, which reduces the number of required connections and simplifies the PCB design.
7. **DHT22 (Temperature and Humidity Sensor):** Chosen over the DHT11 due to its better accuracy and wider operating range.
8. **RTC D53231:** This real-time clock is important for maintaining accurate timekeeping, especially in data logging applications.
9. **LM317L508 Voltage Regulator:** For this project, it is configured to output a steady 3.3V, ensuring that all the components in the system receive the correct operating voltage. Since all components used in the project, including the ESP32, sensors, and display, operate optimally at 3.3V, this regulator plays a critical role in system stability and reliability. It prevents voltage fluctuations that could lead to erratic behavior or damage to sensitive components.
10. **MINI-SPDT-SW Switch:** we used this for the on-off switch.
11. **BC2AAPC Battery Holder:** It was chosen because it supports common AA batteries, which are readily available and provide sufficient power for the system.

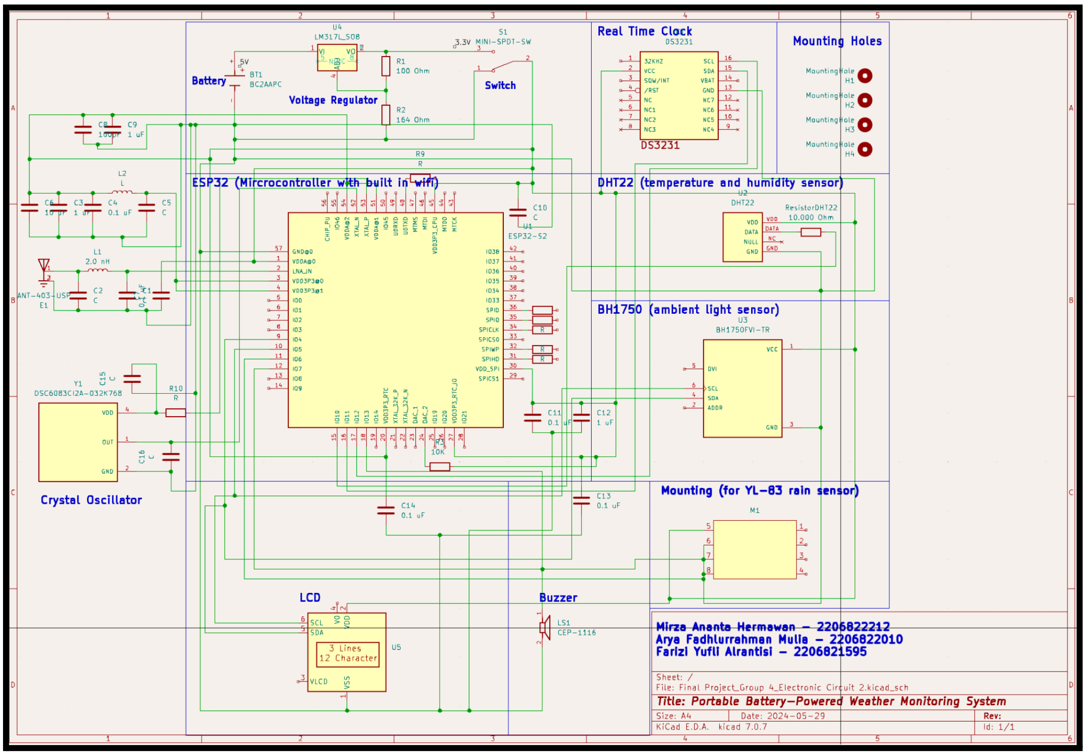


Figure 1: The schematic in KiCad

B. PCB Layout

In the PCB layout all the components are strategically placed to minimize the wiring distance and reduce potential interference.

1. **Central placement of ESP32:** The ESP32 chip is centrally located to minimize the length of connections to other peripheral components, reducing signal degradation and interference. It's surrounded by essential passive components (resistors and capacitors) required for its operation, ensuring stability and reliability.
2. **Power management:** The LM317L508 voltage regulator is positioned near the power input from the battery holder (BC2AAPC). This placement helps in immediate voltage regulation of the incoming power supply, ensuring that all components receive a consistent 3.3V as needed. Close to it, various filtering capacitors are placed to smooth out any voltage fluctuations.
3. **Sensor placement:**
   a. DHT22: located near the ESP32 to ensure short and direct signal paths for accurate temperature and humidity readings, minimizing the effects of electrical noise on the sensor's analog signal.

b. BH1750 light sensor: also close to the ESP32 connected via I2C, which benefits from shorter communication lines to reduce potential communication errors and interference.

c. YL-83 rain sensor mounting: also clode to the ESP32 to ensure accurate measurements and minimize noise.

4. **Display and user interface:**

a. LCD display (U5): Positioned on the top of the board for easy viewing. The display uses an I2C interface, reducing the number of wires needed, which simplifies the routing and reduces clutter on the PCB.

b. Buzzer (LS1): Located on the side for effective sound dispersal without obstruction, which is critical for alerting users to specific environmental thresholds being exceeded.

5. **Connectivity:** antenna (ANT-403-USP): Placed at the edge of the board for clear signal transmission and reception, which is crucial for maintaining the integrity of the communication capabilities of the ESP32.

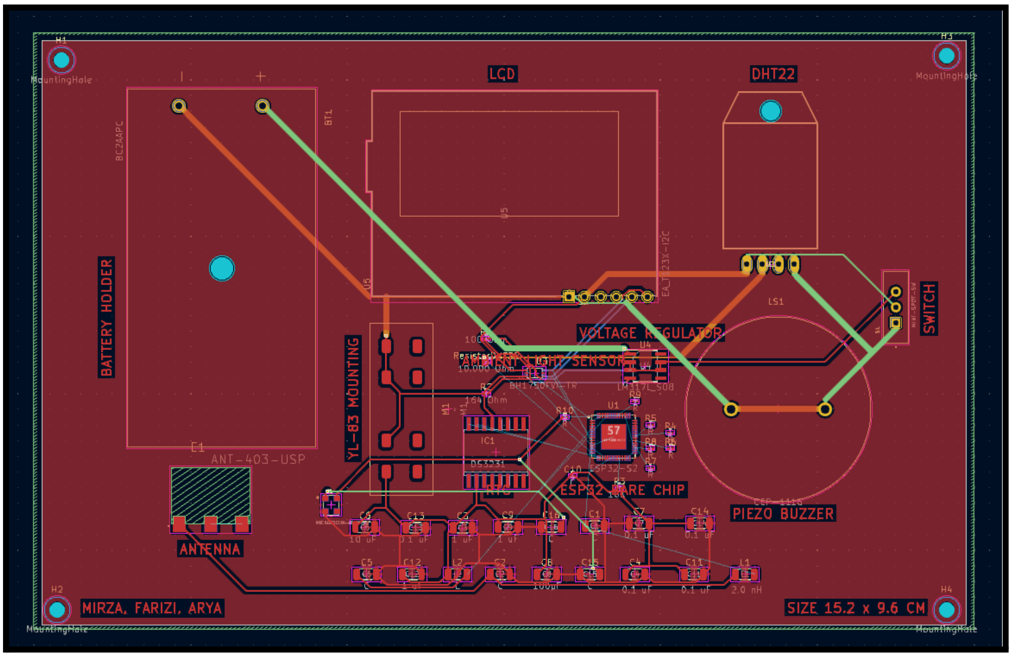
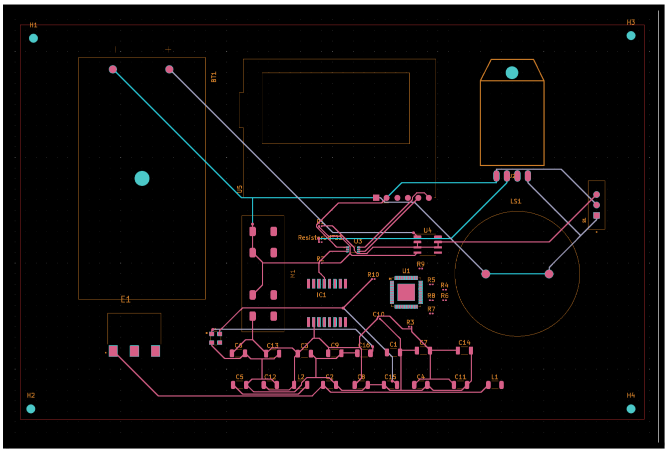


Figure 2: The PCB layout



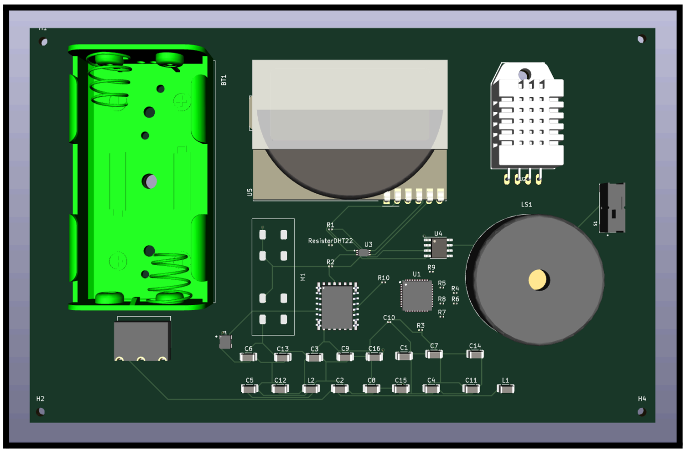


Figure 3: The Gerber file **Figure 4: 3D PCB**

C. Design Constraint

One of the most difficult challenges in the PCB design was managing the connections for the ESP32 bare chip, which required additional external resistors, capacitors, and a stable power supply to function correctly. This was quite complex due to the need for precise placement and routing of these small components, while keeping the design compact and efficient. Furthermore, we encountered difficulties with the YL-83 rain sensor due to the lack of available symbols and footprints. To address this, we created a custom mounting that securely holds the sensor and connects it to the main PCB via designated points. This solution not only resolved the placement issue but also allowed for optimal sensor positioning without compromising the PCB's integrity and layout.

D. Conclusion

The "Portable Battery-Powered Weather Monitoring System" demonstrates the possibility of integrating many sensors into a compact and user-friendly format. Despite the complexities caused by using an ESP32 bare chip, which made us use numerous passive components, and challenges such as creating a custom mounting solution for the YL-83 rain sensor, the project successfully culminated in a robust design. These challenges were met with creative solutions that not only ensured the functionality and reliability of the system but also enhanced our group's skills in PCB design and problem solving. As a result, this system is a proof of the adaptability required in modern electronics design and offers a scalable model for future developments in portable environmental monitoring technologies.

REFERENCES

[1] “ESP32-S2 Hardware Design Guidelines,” SparkFun Electronics. Available: https://cdn.sparkfun.com/assets/5/2/8/9/4/esp32-s2_hardware_design_guidelines.pdf.

[2] “Interfacing DS3231 RTC with Arduino and DIY Digital Clock,” Circuit Digest. Available: https://circuitdigest.com/microcontroller-projects/interfacing-ds3231-rtc-with-arduino-and-diy-digital-clock.

[3] “Complete ESP32 Course - #1 Getting Started,” YouTube, uploaded by Rui Santos. Available: https://www.youtube.com/watch?v=S_p0YV-JlfU&t=8894s.

[4] “ESP32 with DHT11 DHT22 Temperature and Humidity Sensor using Arduino IDE,” Random Nerd Tutorials. Available: https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/.

[5] “ESP32 with BH1750 Ambient Light Sensor,” Random Nerd Tutorials. Available: https://randomnerdtutorials.com/esp32-bh1750-ambient-light-sensor/.