REAL TIME SYSTEM AND INTERNET OF THINGS
FINAL PROJECT REPORT

ENVIRONMENTAL MONITORING SYSTEM

GROUP 17:

FABSESYA MUHAMMAD PI	2206829433
IRFAN YUSUF KHAERULLAH	2206813290
SHARIF FATIH ASAD MASYHUR	2206063014
RAJA YONANDRO RUSLITO	2206059553

COMPUTER ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING

UNIVERSITAS INDONESIA

DEPOK

2024/2025

PREFACE

We present this project report entitled Environmental Monitoring System which was developed as a project from the IoT and Real Time System lab. This report is designed to fulfill the final assignment requirements of the IoT and Real Time System practicum in the odd semester of 2024/2025, Faculty of Engineering, Department of Electrical Engineering, University of Indonesia. Without the help and cooperation of various parties, this final report would not have been able to be compiled and completed. Therefore, we would like to express our deepest gratitude to all parties who have helped, especially:

1. Mr. Fransiskus Astha Ekadiyanto as the lecturer in charge of the IoT and Real Time System course, odd semester 2024/2025.
2. The Digital Lab assistants who have helped and guided us during the IoT and Real Time System practicum.
3. Parents and fellow practicum participants who have provided support, both in the form of materials and ideas.

We state that this section is far from perfect. Constructive criticism and comments are highly expected to help improve the quality of this work. Thus, this section can be a useful reference for students who want to better understand the results of the practicum that has been achieved.

In short, the authors believe that the final project report of IoT and Real Time Systems lab can make a positive contribution to the development of knowledge and practical understanding in related fields. We hope that this section can be a guide for better understanding and application of knowledge in the context of everyday life.

Depok, December 03, 2024

Group 17

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Poor environmental conditions, particularly air quality and temperature, can have significant impacts on human health and ecosystem health. However, in many places, real-time environmental monitoring is still quite difficult. A robust, easy-to-use system that can provide early warnings can counteract the lack of public awareness of potential environmental events, such as air or temperature extremes.

Existing monitoring systems are often quite large, difficult to dismantle, and require complex infrastructure. In this situation, there is a need to develop an Internet of Things-based solution that can provide real-time environmental data, integrate with visualization platforms, and provide alerts if environmental parameters affect the battery. This is important to guide risk mitigation and escalation strategies, both at the individual and group levels.

Furthermore, the lack of responsive monitoring systems often slows down decision-making regarding environmental risk mitigation. Without data that is accessible in real time, individuals and organizations cannot quickly adjust their actions to protect health and safety. By integrating IoT technologies, communication protocols such as MQTT, and visualization platforms such as Blynk, innovative solutions are needed that not only provide accurate environmental information, but also facilitate effective data monitoring and management, both through physical devices such as LCDs and through mobile applications.

1.2 PROPOSED SOLUTION

To address the issue of a sensitive and responsive environmental monitoring system, this project demonstrates the development of an IoT-based environmental monitoring system using ESP32. The system is designed to monitor Air Quality Index (AQI) and water quality in real time using dedicated sensors. The data will be analyzed and visualized through the Blynk platform, allowing users to monitor environmental conditions using their smartphones.

The project also integrates the MQTT communication protocol to ensure efficient and transparent data transmission between sensors, servers, and applications. As an additional feature, an LCD will be used to silently display data on the screen, while actuators will provide alerts if environmental parameters indicate a safe battery. The solution is designed to provide high accessibility, making it easier for individuals and organizations to protect the surrounding environment, and quickly implement mitigation measures.

With a cost-effective IoT-based approach, the system offers a practical, real-time, and integrated solution to mitigate environmental risks while raising public awareness of the importance of healthy air quality and temperature.

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. The system must integrate with the Blynk platform, displaying AQI, humidity and temperature values on a mobile app interface.
2. Alerts should be activated via LED indicators or in-app notifications if environmental parameters exceed predetermined thresholds.
3. Data from the sensors must be transmitted via MQTT protocol without significant delays.
4. Data must also be displayed on the connected LCD screen in real-time and have a clear and readable LCD display for real-time data.
5. Sensor reading that is fully functional and does well in the presence of local variations in air quality.
6. A servo motor must automatically operate in accordance with the surrounding conditions.
7. LED indicators should operate with diligence and provide visual indications of the air quality status.

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Role 1	Report Writing	Fabsesya Muhammad P. I.
Role 2	Coder and Assembling Real Circuit	Irfan Yusuf Khaerullah
Role 3	Create Project Presentation and Assists in Assembling Reports	Sharif Fatih Asad Masyhur
Role 4	Idea Concepting and Report Writing	Raja Yonandro Ruslito

Table 1.1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

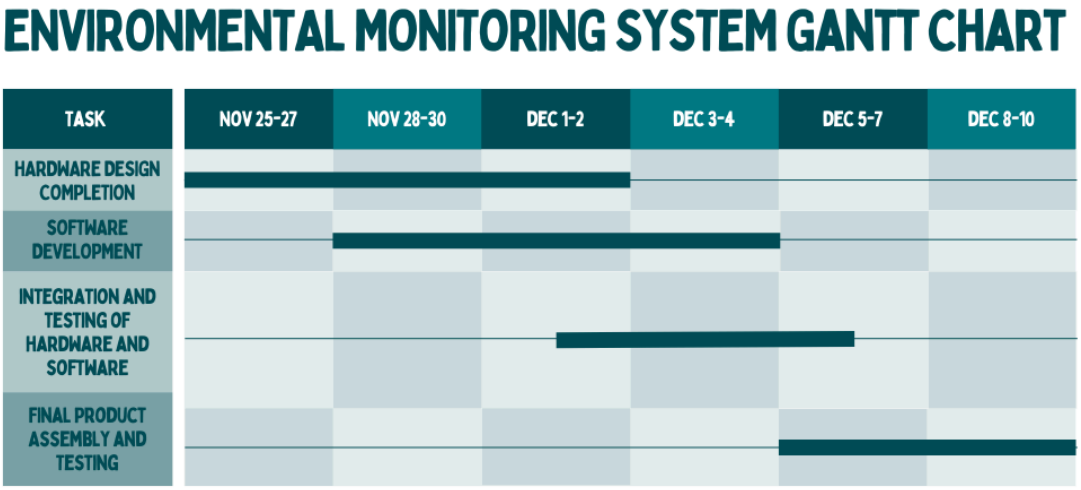


Fig 1.1. Gantt Chart

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

Our hardware design includes an ESP32, a DHT11 temperature and humidity sensor, an MQ135 gas sensor, an air quality index (AQI) sensor, an LCD display, a servo motor, LED indicators, and jumper cables. The connection uses multiple communication protocols to integrate the components efficiently.

The DHT11 sensor is connected to the ESP32 for data transmission, the MQ135 sensor and AQI sensor are analog sensors connected to the ADC pins on the ESP32 for air quality monitoring. Both sensors measure pollutant levels and contribute to calculating the Air Quality Index (AQI). The LCD display uses an I2C communication interface, with SDA and SCL pins connected to GPIO pinout on the ESP32 responsible to control or read the state of a specific pin which handles incoming and outgoing digital signals.

The ESP32 serves as the central processing unit responsible for collecting data from the sensors and processing it. The system is designed to monitor air quality and temperature in real-time. If the AQI or temperature exceeds predefined thresholds, the ESP32 triggers visual and audible alerts.

The Servo motor is connected to a PWM GPIO pin on the ESP32 and can be used to translate the controller's control signal into the motor output shaft's rotational angular displacement or angular velocity. LED indicators are connected to digital output pins to provide real-time visual feedback on air quality status.

Lastly, the system uses the MQTT protocol to send real-time data to the cloud, allowing visualization and control through the Blynk application. The ESP32 is powered via a USB connection, while all components are integrated on a protoboard for organization and shared connections.

2.2 SOFTWARE DEVELOPMENT

The software development for this project involves programming the ESP32 microcontroller to collect, process, and communicate real-time environmental data using various sensors and peripherals. The software is implemented in a single .ino file and integrates with cloud platforms such as Blynk and MQTT for monitoring and control.

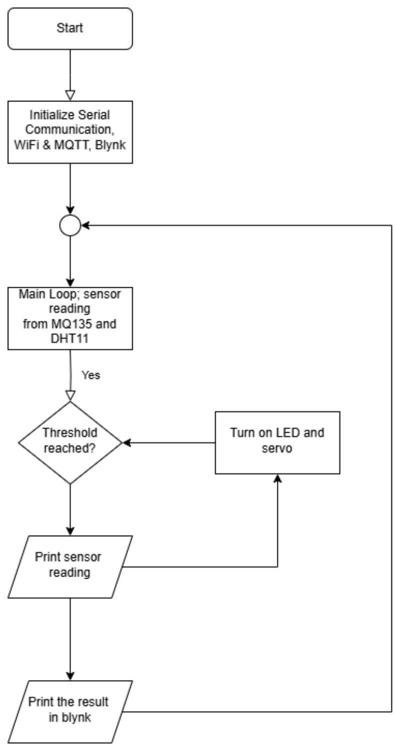


Fig 2.1. Flowchart of the Program

The process starts with the initialization of crucial components including serial communication for debugging, Wi-fi for network connectivity, and the MQTT and Blynk platforms for cloud-based data exchange. After initialization the system enters a loop where it continuously reads data from the MQ135 sensor and DHT11 temperature. The sensor data is then compared against predefined threshold values. If any parameter exceeds its threshold, the system triggers alerts. These alerts include activating an LED indicator and controlling a servo motor, which could perform specific tasks such as opening ventilation. The system logs sensor simultaneously reads to the serial console for debugging and sends the data to the Blynk platform, enabling users to monitor environmental conditions remotely and in real time. The loop continuously monitors and repeats the process to maintain a responsive and dynamic system.

2.3 HARDWARE AND SOFTWARE INTEGRATION

Finally, the hardware and software will be integrated by uploading the program into ESP32 and connecting all the necessary components in place and configuring the communication protocols. The Blynk template for the integration with WiFi connection. Once the hardware and software are ready, the integration phase focuses on establishing communication between the components, and testing the system's functionality.

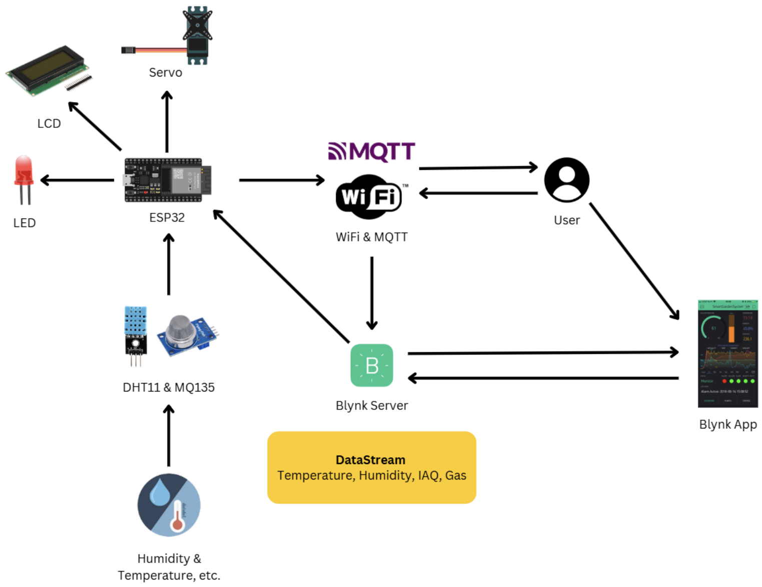


Fig 2.2. System Schema

Thorough documentation is crucial, encompassing hardware assembly, software implementation, calibration procedures, and user instructions. This serves as a valuable resource for effectively using and maintaining the air quality monitoring device. Seamless integration of hardware and software components ensures smooth operation and delivers accurate and dependable air quality measurements.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

During the testing phase of our IoT project, "Environmental Monitoring System," we conducted a series of tests to ensure the program and circuit operated in line with established criteria and expectations. Using the Blynk 2.0 application, the ESP32 connected to the WiFi network using the specified SSID and password. Once connected, the LCD 20x4 screen illuminated and displayed Temperature, Humidity, and Gas values. The LED and servo were configured to activate under specific conditions. On the Blynk mobile app, the values for Temperature, Humidity, and Gas were displayed, accompanied by a status message indicating air quality based on the sensor readings.

Initially, testing was performed without the Blynk interface to ensure all data collected by the DHT11 and MQ135 sensors appeared accurately and consistently in the serial monitor. This step verified that the data was being correctly received and processed. The results were successfully displayed on the serial monitor.

```
Message arrived on topic: env-monitor/output. Message: Environment monitored.
21:41:03       Mutex taken
21:41:03       Gas: 16.00      Temperature: 24.80°C    Humidity: 48.00%
21:41:04       Temperature is within normal range.
21:41:04       Air quality is GOOD - Servo deactivated
21:41:04       Mutex given
--------------------------------
21:41:08       Mutex taken
21:41:08       Gas: 15.00      Temperature: 24.80°C    Humidity: 48.00%
21:41:09       Temperature is within normal range.
21:41:09       Air quality is GOOD - Servo deactivated
21:41:09       Mutex given
--------------------------------
21:41:13       Mutex taken
21:41:13       Gas: 16.00      Temperature: 24.80°C    Humidity: 48.00%
21:41:14       Temperature is within normal range.
21:41:14       Air quality is GOOD - Servo deactivated
```

Fig 3.1. Serial Monitor Testing Result

Next, we tested the LCD 20x4 screen to confirm proper communication between the ESP32, DHT11, MQ135, and the display using the I2C protocol. Once the sensor readings were verified for accuracy, we proceeded to test the functionality of the LED indicators and servo. The LEDs were programmed to activate under predefined conditions, while the servo emitted a sound when the air quality fell below a specified threshold, categorized as "Poor." Following these initial tests, we integrated the system with Blynk 2.0. The data obtained from the DHT11 and MQ135 sensors were displayed on the serial monitor, LCD screen, and the

Blynk app. It was critical to ensure that all three outputs displayed consistent and accurate values. Additionally, we closely monitored the timing delay between updates on the LCD and the Blynk platform to ensure that any discrepancies were negligible. The status message on the Blynk interface was cross-verified to match the air quality condition accurately, providing users with reliable and immediate information.

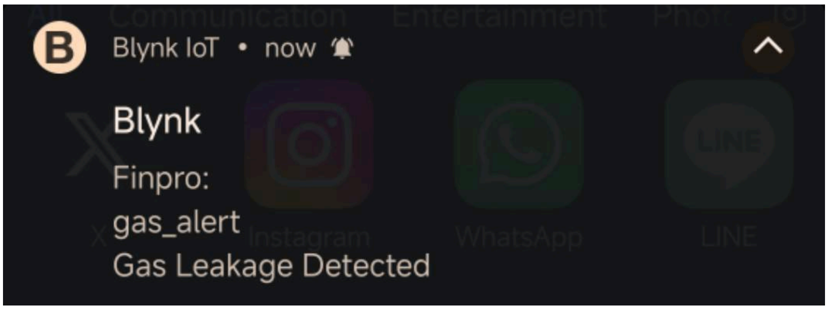


Fig 3.2. Blynk Application Notification

The final phase focused on ensuring seamless synchronization of data across the serial monitor, LCD 20x4 screen, and the Blynk interface. By minimizing data synchronization delays and ensuring the alignment of air quality indicators, users could access timely and accurate updates about their environment through the Blynk app. This process aimed to enhance user understanding and facilitate informed decision-making regarding indoor air quality conditions.

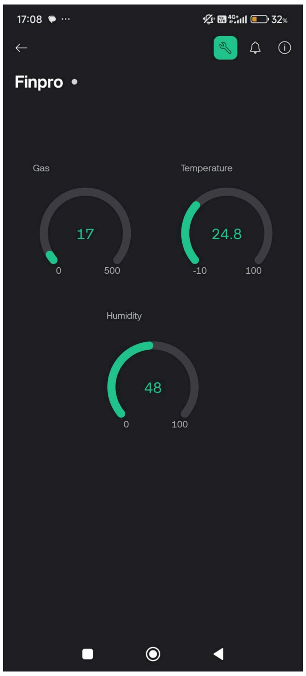


Fig 3.3. Blynk Result

This integration testing phase emphasized synchronizing data output across the serial monitor, LCD screen, and Blynk interface to ensure consistency and real-time accuracy. Particular attention was given to reducing delays in data synchronization between the local display and the Blynk platform, guaranteeing that users received timely and consistent updates on air quality status through the Blynk mobile app. Additionally, ensuring that the status message displayed on Blynk accurately reflected the actual IAQ condition was a priority to improve user understanding and support informed decisions about the surrounding air quality.

3.2 RESULT

The results from the three tests conducted on the original circuit using the Blynk platform matched the predefined criteria and expectations. These tests thoroughly evaluated the system's functions and responses to confirm optimal performance and seamless real-time integration with Blynk.

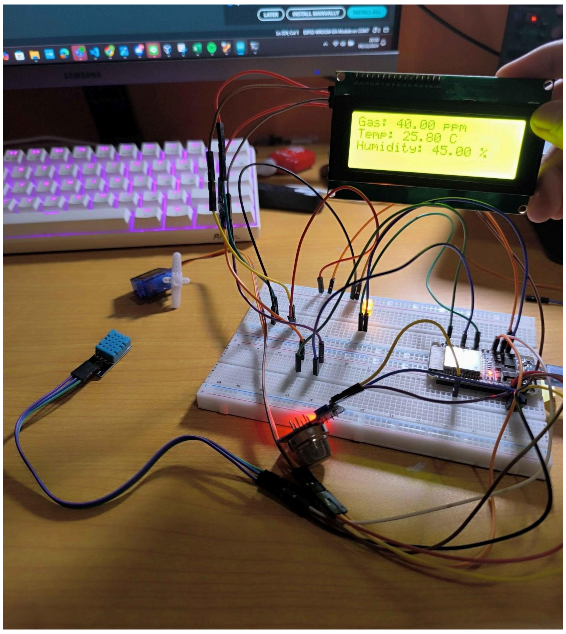


Fig 3.4. Final Result

Based on the detailed testing results, the IoT project *"Environmental Monitoring System"* successfully combined hardware and software to create an intelligent environmental monitoring solution. The system efficiently collects data from the DHT11 and MQ135 sensors, displaying it promptly on the LCD screen and user devices through Blynk.

Additionally, the system incorporates LED indicators that respond to air quality conditions and a servo motor that activates when poor air quality is detected, providing users with timely alerts to help mitigate potential health risks.

3.3 EVALUATION

The evaluation phase focused on analyzing the functionality, user experience, and overall effectiveness of the *"Environmental Monitoring System"* after thorough testing and feedback collection.

The system reliably provided real-time, precise measurements of temperature, humidity, and gas levels. The integration of the ESP32, DHT11 and MQ135 sensors, LCD display, LED indicators, servo motor, and the Blynk platform showcased its stability and performance. The system's ability to respond to predefined conditions, such as activating LED signals and controlling the servo based on air quality thresholds, demonstrated its reliability.

The Blynk mobile application offered a seamless and straightforward user experience. It allowed users to access vital environmental data with clear graphics and concise updates on air quality. Synchronization between the local display and Blynk ensured smooth monitoring across devices. The LED indicators and servo motor actions served as effective notifications, enabling users to react promptly to changes in air quality.

By efficiently collecting and displaying sensor data on both the LCD and Blynk platforms, the system ensured accurate and timely monitoring. The correspondence between the Blynk-generated messages and the actual air quality levels improved communication, helping users make informed decisions. The LED alerts and servo responses were crucial in notifying users about poor air quality, encouraging timely preventive actions.

In conclusion, the evaluation affirmed that the *"Environmental Monitoring System"* successfully met its goals. It proved to be a dependable, user-friendly, and efficient tool for monitoring indoor environmental conditions. Its seamless combination of hardware, software, and user interface features makes it a practical and versatile solution for creating healthier and safer indoor spaces.

CHAPTER 4

CONCLUSION

The development of the air quality monitoring system is a significant advancement in tackling the challenges of air pollution. By integrating the MQTT, DHT11 and MQ135 sensors, ESP32 microcontroller, and Blynk platform, the system offers a complete solution for monitoring critical environmental factors while providing real-time accessibility through an intuitive interface.

The software has been thoughtfully designed for optimal performance, featuring a modular code structure, robust error-handling, and seamless integration with the Blynk platform. This enables users to effortlessly view air quality data, receive instant alerts, and personalize their monitoring experience using the Blynk application.

The integration of hardware and software components was achieved successfully, ensuring smooth operation and accurate air quality measurements. Extensive testing procedures were carried out to validate the system's reliability across various scenarios, and a thorough calibration process was defined to ensure consistent accuracy over time.

REFERENCES

[1] Digilab UI, “Module 9: IOT Platform - Blynk” *Digilabdte.com*, [Online]. Available: https://learn.digilabdte.com/books/realtime-system-iot/page/module-9-iot-platform-blynk [Accessed: 05-Dec-2024].

[2] Digilab UI, “Code Example | Module 9,” *Digilabdte.com*, 2024. Available: https://learn.digilabdte.com/books/realtime-system-iot/page/code-example [Accessed: 05-Dec-2024].

[3] GeeksforGeeks: GeeksforGeeks, "Introduction to Message Queue Telemetry Transport (MQTT) Protocol," GeeksforGeeks, [Online]. Available: https://www.geeksforgeeks.org/introduction-of-message-queue-telemetry-transport-protocol-mqtt/. [Accessed: 09-Dec-2024].

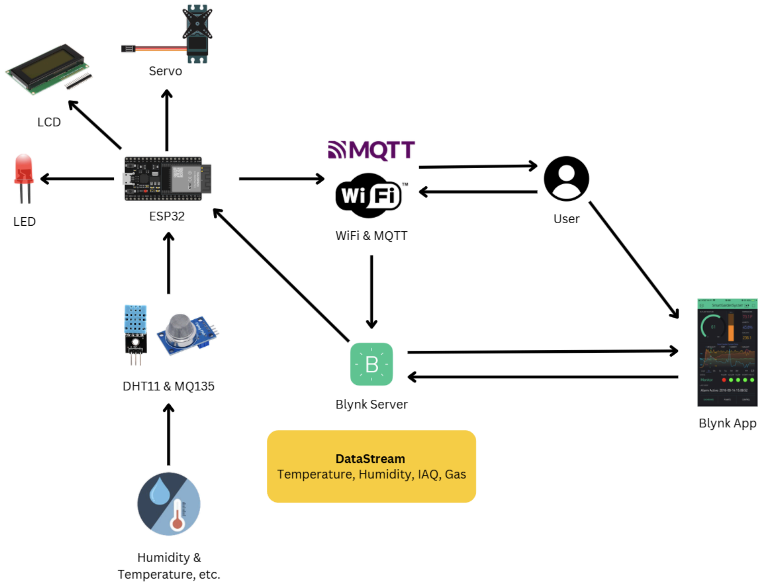
[4] Javatpoint: Javatpoint, "MQTT Protocol," Javatpoint, [Online]. Available: https://www.javatpoint.com/mqtt-protocol. [Accessed: 09-Dec-2024].

[5] Digilab UI, “Module 7 : WiFi, HTTP, and MQTT,” *Digilabdte.com*, 2024. [Online]. Available: https://learn.digilabdte.com/books/realtime-system-iot/chapter/module-7-wifi-http-and-mqtt [Accessed: 09-Dec-2024].

[6] Random Nerd Tutorials, “ESP32 Useful Wi-Fi Library Functions (Arduino IDE) | Random Nerd Tutorials,” Feb. 12, 2021. [Online]. Available: https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/ [Accessed: 09-Dec-2024].

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation