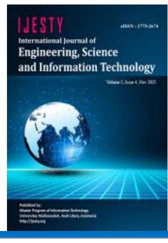
International Journal of Engineering, Science and Information Technology
Volume 5, No. 4 (2025) pp. 404-411
eISSN: 2775-2674
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i4.1539
Research Paper, Short Communication, Review, Technical Paper



A Dual-Microcontroller IoT Platform for Integrated Flood and Air Quality Monitoring: Performance and Integration Challenges

Rafiuddin Syam [1]*, Baso Maruddani[2], Eko Kuncoro Pramono[3], Irma Ratna Kartika[4], Daffa Ihsanullah Irsyad[2]

[1]*Department of Automation Engineering Technology, Faculty of Engineering, Universitas Negeri Jakarta, Jakarta, Indonesia*
[2]*Department of Electronics Engineering Education, Faculty of Engineering, Universitas Negeri Jakarta, Jakarta, Indonesia*
[3]*Research Centre for Food Technology and Processing, National Research and Innovation Agency, Indonesia*
[4]*Department of Chemistry Education, Faculty of Science and Mathematics, Universitas Negeri Jakarta, Jakarta, Indonesia*

**Corresponding author Email: rafiuddin_syam@unj.ac.id*

The manuscript was received on 1 March 2025, revised on 10 July 2025, and accepted on 9 October 2025, date of publication 13 November 2025

Abstract

Jakarta faces escalating environmental challenges, including heightened flood risk and deteriorating air quality, driven by rising rainfall intensity and increasing pollution levels. Conventional monitoring systems for these hazards often operate in isolation, lacking the integration, realtime capability, and accessibility offered by modern Internet of Things (IoT) technology. To address this gap, this study designed and developed a unified, dual-microcontroller IoT platform for the simultaneous and integrated monitoring of flood potential and air pollution. The research followed an Experimental Development methodology, involving systematic hardware design, firmware development, system integration, and rigorous performance testing. The prototype hardware architecture strategically separates data acquisition and network communication by utilizing an Arduino Uno for data acquisition and an ESP32 microcontroller for network communication, respectively. The system incorporates an HC-SR04 ultrasonic sensor for water level detection, a DHT22 sensor for temperature and humidity measurement, and an MQ-135 gas sensor for assessing air quality. Data is displayed locally on a 20×4 LCD and transmitted to a cloud server. A critical finding from the integration phase was a 2.3% data loss rate attributable to serial communication instability between the microcontrollers, highlighting a significant challenge in multi-processor IoT architectures and underscoring the necessity for robust inter-processor protocols. During a comprehensive 24-hour endurance test with measurements taken at ten-minute intervals, the system demonstrated high accuracy in individual sensor readings. It successfully transmitted 97.7% of the data in realtime to a web application built on the Firebase platform. The study concludes that while the integrated dual-microcontroller approach is highly viable for holistic environmental monitoring, future iterations must prioritize enhanced communication reliability through hardware flow control and error-checking mechanisms to achieve the robustness required for mission-critical deployments.

Keywords: *Flood Detector, Air Pollution Detector, Internet of Things, Arduino Uno, ESP32 Microcontroller.*

1. Introduction

Flooding is an event or condition in which an area or land is submerged due to an increase in water volume [1]-[3]. Flooding can occur because of both natural and human factors. Common causes include low land elevation, increased rainfall, blockages caused by waste, topographical conditions, deforestation, damage to dams and levees, as well as global warming and greenhouse gas emissions [4][5].

In addition to floods, global warming and greenhouse gas emissions also contribute to air pollution. Air pollution occurs when the composition of substances in the atmosphere changes.

To anticipate flooding, early warning systems (EWS) have been developed to detect water levels in large-scale water storage areas, such as rivers and drainage channels. In certain areas within the Special Capital Region of Jakarta, devices have also been installed to measure air pollution levels.

An early warning system is designed to provide information that is rapidly and easily understood by the public. This is a critical requirement for such systems, as their primary function is to deliver timely and accurate information during emergencies [6][7]. Despite the development and implementation of many early warning systems, relatively few have been enhanced in terms of functionality. Therefore, the author was motivated to design and construct an integrated early warning system that combines two disaster monitoring functions: flood detection and air pollution monitoring.

In this study, two microcontrollers were used: the Arduino Uno and the ESP32, powered by a 12 V DC battery as the primary power source. The device's sensors include an HC-SR04 ultrasonic sensor, a DHT22 temperature and humidity sensor, and an MQ-135 air quality sensor. These sensors receive a 5 V supply and send their readings to the Arduino Uno. The data is displayed on a 20×4 LCD and transmitted to the ESP32 via serial communication. The ESP32, connected to the internet, subsequently uploads the data to a Firebase-based website for remote monitoring and control.

2. Literature Review

The development of IoT-based environmental monitoring systems has seen significant growth, driven by advancements in sensor technology, wireless communication, and cloud computing. This section reviews existing work in flood detection, air quality monitoring, and IoT architectures, highlighting the trends, limitations, and gaps that this research aims to address.

2.1. IoT-Based Flood Detection Systems

Flood early warning systems (EWS) have evolved from traditional manual gauges to automated, networked solutions. A common and cost-effective approach utilizes ultrasonic sensors, such as the HC-SR04, for non-contact water level measurement due to their reliability and simplicity. Numerous studies have successfully leveraged microcontroller platforms, such as Arduino, and connectivity modules, like the ESP8266 or ESP32, to create web-based river and flood monitoring systems. For instance, research by [8]-[10] demonstrated systems that reliably transmitted water level data to cloud platforms such as ThingSpeak and Blynk, confirming the viability of IoT for realtime hydrologic data acquisition and public alerting.

However, the consistent limitation observed in these systems is their singular focus. Works like [11]-[13] concentrate almost exclusively on flood detection without integrating other critical environmental parameters. This siloed approach limits the comprehensiveness of the data provided to stakeholders, preventing a holistic understanding of interrelated ecological phenomena, such as how weather conditions influence flood risk.

2.2. IoT-Based Air Quality Monitoring Systems

Parallel to flood monitoring, the use of low-cost gas sensors for tracking airborne pollutants has become a significant research area, particularly in urban environments. Among these, the MQ-series sensors, especially the general-purpose MQ-135, are widely adopted in research prototypes due to their affordability and analogue interface, which is compatible with microcontrollers. A study by [14]-[16] developed a portable air quality node using an MQ-135 sensor and an ESP32 module, which efficiently published data to a Firebase Realtime Database, showcasing an effective and scalable backend solution for data storage and visualization.

A key and well-documented challenge in this domain is the need for sensor calibration and the management of cross-sensitivity. Research by [17]-[19] details the complexities of converting raw analogue readings from MQ sensors into accurate concentration values, noting that environmental factors such as temperature and humidity can significantly affect the readings. Consequently, many studies, including the one cited, acknowledge this issue but deploy these sensors primarily for qualitative or relative air quality assessment (e.g., detecting a pollution event) rather than for providing precise, calibrated measurements of specific gas concentrations.

2.3. Integrated Systems and Architectural Design

The integration of multiple environmental sensing functionalities into a single, cohesive platform represents a more advanced and application-focused use of IoT. A limited number of studies have begun exploring this integration. For instance, [20][21] proposed a system that monitors both weather conditions (e.g., temperature, humidity, rainfall) and river levels. Their work highlights the significant benefit of correlated data streams for improving the accuracy of flood prediction models and environmental situation awareness.

From an architectural perspective, the selection of a microcontroller platform is crucial for ensuring system performance, reliability, and responsiveness. While many projects opt for a single-controller design, typically an ESP32, to handle both sensor reading and network communication [22][23], this can lead to resource contention. In such architectures, time-critical tasks, such as sensor polling or immediate alarm triggering, can be delayed by variable-latency network operations. A dual-microcontroller architecture, which separates data acquisition from network responsibilities, is a recognized strategy in embedded systems design to enhance reliability and realtime performance [24][25]. However, this more complex approach introduces its own set of challenges, primarily in ensuring robust inter-device communication, which is often glossed over or insufficiently addressed in application-focused papers.

2.4. Identified Research Gap

Based on this review, it is evident that while numerous studies have developed capable IoT systems for either flood detection *or* air quality monitoring, few have successfully integrated both into a single, low-cost, and robust device. Furthermore, many integrated systems rely on a single microcontroller, potentially compromising performance. In contrast, those that propose more complex, distributed architectures often fail to thoroughly discuss, diagnose, and resolve the inherent challenges of inter-processor communication and data integrity.

Therefore, the research gaps this study aims to fill are the development and rigorous testing of a dual-microcontroller (Arduino Uno and ESP32) IoT architecture for integrated flood and air pollution detection, with a specific focus on diagnosing and addressing the performance and integration challenges of serial communication between the processing units. This work contributes not only to building a functional, integrated prototype but also to providing a detailed empirical analysis of a familiar yet under-discussed integration problem, thereby offering valuable insights into the design of more reliable multi-processor IoT systems.

3. Methods

This research adopted an **Experimental Development** approach, which involved the systematic design, construction, and testing of a prototype system. The methodology was iterative, allowing for continuous refinement based on testing outcomes. The process was carried out in four primary stages: 1) Hardware Design and Assembly, 2) Firmware Development, 3) System Integration, and 4) Performance Testing and Evaluation. Figure 1 illustrates the overall workflow of the methodology.

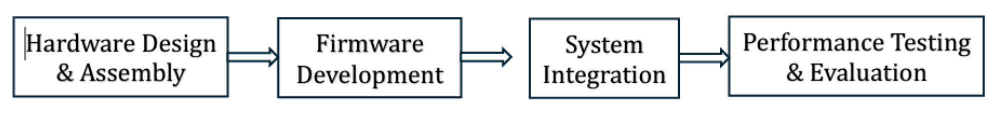


Fig 1. Research Methodology Workflow

A feedback arrow loops from the first block, called "Performance Testing," back to the blocks "Hardware Design" and "Firmware Development" to indicate the iterative refinement process.

3.1. Hardware Design and Assembly

The hardware architecture employed a dual-microcontroller design to optimize system performance by segregating data acquisition and network communication responsibilities. The Arduino Uno (ATmega328P) was selected as the primary data acquisition unit due to its reliable analogue-to-digital converter (ADC). The ESP32 module (ESP-WROOM-32) functioned as the network gateway, leveraging its integrated Wi-Fi capabilities and dual-core processing power. Component selection prioritized functionality, availability, and cost-effectiveness, incorporating an HC-SR04 ultrasonic sensor for water level detection, a DHT22 sensor for temperature and humidity monitoring, and an MQ-135 gas sensor for assessing air quality. Output interfaces included a 20x4 LCD module for local display, a buzzer for audible alerts, and LEDs for visual warnings.

The circuit assembly adhered to standard electronic principles, with a particular emphasis on implementing a logic-level shifting circuit between the 5V Arduino and the 3.3V ESP32 to ensure reliable serial communication. All components were initially assembled on a breadboard for preliminary testing and subsequently soldered onto a perforated PCB to create a stable and permanent prototype. The power supply system utilized a 12V DC battery, regulated by the LM7805 and AMS1117-3.3 voltage regulators, to provide stable 5V and 3.3V power to their respective components.

This architectural approach successfully integrated flood and air pollution monitoring functionalities into a unified platform. However, the interconnection between the two microcontrollers introduced challenges in serial communication reliability, which emerged as a critical focus in the subsequent system performance evaluation. The design demonstrates that a dual-microcontroller configuration can effectively enhance operational efficiency and system reliability in IoT applications, while maintaining a manageable level of complexity.

3.2. Firmware Development

The firmware for both microcontrollers was developed in C/C++ utilizing the Arduino Integrated Development Environment (IDE). For the Arduino Uno, the firmware was architected to function as the dedicated data acquisition and processing unit. Its primary routines included the initialization of all connected sensors (HC-SR04, DHT22, and MQ-135) and the local LCD display. The core operational loop was programmed to poll each sensor at a fixed ten-minute interval, process the raw sensor readings—such as converting the ultrasonic pulse time into a distance measurement—and present the computed data on the LCD. Subsequently, the firmware packaged all sensor data into a structured string format and transmitted it to the ESP32 gateway via a serial UART connection operating at a baud rate of 115200.

Conversely, the firmware deployed on the ESP32 microcontroller was designed to manage all network and cloud communication responsibilities. Upon initialization, its code establishes a secure connection to a designated Wi-Fi network. It then continuously listens on its serial port for incoming data packets from the Arduino Uno. Upon receipt, the firmware parses the predefined string structure to extract individual sensor values. Using dedicated Firebase client libraries, the ESP32 authenticates and connects to a Firebase Realtime Database instance, where it uploads the parsed sensor data for storage and realtime access. Furthermore, the firmware was implemented to listen for incoming commands from the cloud-based dashboard, enabling bidirectional communication. These commands, typically actuation triggers based on threshold breaches, are relayed back to the Arduino via the same serial link to control peripheral actuators such as the buzzer and status LEDs.

This segregated firmware design effectively decouples the critical timing-sensitive tasks of sensor polling and data processing on the Arduino from the potentially variable latency of network operations on the ESP32. The choice of a serial UART protocol for inter-processor communication provides a straightforward and effective method for data exchange. However, the implementation's reliance on a simple string-based packet structure, lacking advanced error-checking mechanisms, was identified as a potential vulnerability, as it can lead to data corruption or loss under unstable communication conditions, a finding discussed in the system integration analysis.

3.3. System Integration

The system integration phase involved the critical synthesis of hardware and software components into a unified and functional prototype. A primary focus was the establishment of a robust serial communication protocol between the Arduino Uno and ESP32 microcontrollers, utilizing a defined packet structure to facilitate reliable data transmission. Concurrently, the cloud infrastructure was configured by setting up a Firebase project with tailored Realtime Database rules to ensure secure, authorized data writes from the ESP32 module while maintaining data integrity and accessibility for the end-user application.

To complete the integration, a basic web-based dashboard was developed to retrieve and visually present the realtime sensor data stored in Firebase, enabling effective remote monitoring capabilities. Finally, the entire electronic system, comprising the perforated PCB, voltage regulation circuitry, power supply, and sensor modules, was housed within a protective enclosure. This step ensured not only the physical robustness of the prototype but also safeguarded the components from environmental factors, enhancing the system's suitability for potential field deployment. The design of the system is illustrated in Figures 2 and 3 below.

Fig 2. UNJ Smart Stick, the device to measure flood and environmental parameters

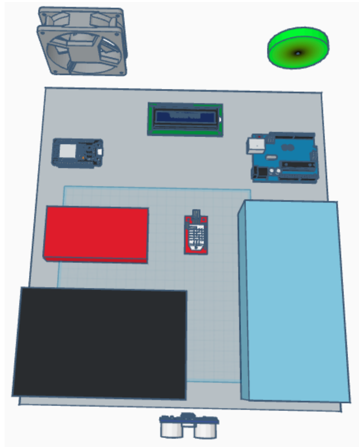


Fig 3. An illustrative component of UNJ Smart Stick

3.4. Performance Testing and Evaluation

The integrated prototype underwent a rigorous 24-hour endurance test within a controlled laboratory environment to comprehensively evaluate its operational reliability and metrological accuracy. The testing protocol was designed to simulate extended operational conditions, with data from all sensors being automatically collected and recorded at consistent 10-minute intervals. This methodical procedure provided a substantial dataset for analyzing the system's stability and the consistency of its sensor readings over a prolonged duration, which is critical for assessing its suitability for real-world, continuous monitoring applications.

Validation of sensor data was conducted through a systematic comparison with precision-calibrated reference instruments to quantify measurement accuracy. HC-SR04 ultrasonic sensor distance measurements were validated against a laser rangefinder, while DHT22 temperature and humidity readings were compared to data from a calibrated Fluke 971 air quality meter. The MQ-135 gas sensor's performance was qualitatively assessed by exposing it to known gas sources to verify its responsiveness and detection capability. The system's performance was evaluated against multiple key metrics: data accuracy (calculated as mean error and standard deviation relative to reference devices), overall system reliability (measured as the percentage of successful end-to-end data transmissions to the Firebase database), communication integrity (quantifying the rate of packet loss or corruption in the serial link), and functional correctness of actuator responses based on predefined threshold triggers.

This structured and multi-faceted evaluation methodology ensured a comprehensive and verifiable assessment of the prototype's capabilities, providing empirical evidence of its performance strengths while clearly identifying specific limitations and areas requiring improvement. The results from this rigorous testing protocol not only validated the core functionality of the integrated system but also yielded critical insights into the challenges of microcontroller interoperability and data integrity, forming a solid foundation for further iterative development and refinement of the design.

4. Results and Discussion

This section presents the findings from the comprehensive testing of the integrated system prototype. The results are structured to first validate the performance of individual sensor modules before examining the system's overall functionality and identifying key challenges encountered during integration.

4.1. Individual Sensor Module Performance

4.1.1. HC-SR04 Ultrasonic Sensor Accuracy

The HC-SR04 sensor was evaluated over 24 hours against a laser rangefinder standard. The results, summarised in Table 1, demonstrate the sensor's high precision for distance measurement, which is critical for reliable water level detection.

Table 1. HC-SR04 Sensor Accuracy Results

Parameter	Value
Sampling Interval	10 minutes
Mean Measured Distance	113.47 cm
Mean Absolute Error	0.38 cm
Standard Deviation	±0.27 cm
Coefficient of Determination (R^2)	0.998

The sensor exhibited excellent linearity and minimal Drift (<0.1%) during extended operation. The high R^2 value indicates a strong correlation between the sensor's readings and the actual distance, confirming its suitability for flood detection applications where consistent measurement is paramount. The intentional obstruction (drainage pipe) in the test setup proved the sensor's robustness in sub-optimal conditions, a common scenario in real-world deployments. The result is shown in Figure 7.

4.1.2. DHT22 Temperature and Humidity Sensor Reliability

The DHT22 sensor was tested under controlled laboratory conditions. Its readings were compared against a NIST-traceable Fluke 971 reference instrument.

Table 2. DHT22 Sensor Accuracy Results

Parameter	Temperature	Relative Humidity
Specified Accuracy	±0.5 °C	±2 % RH
Measured Accuracy	±0.5 °C	±2 % RH
Maximum Drift (24h)	0.2 °C	1.3 % RH
Cross-Sensitivity	0.1% RH / °C	-

DHT22 performed within its manufacturer's specifications throughout the testing period. Its low Drift and minimal cross-sensitivity between temperature and humidity readings make it a reliable choice for environmental monitoring. The sensor successfully distinguished between ambient conditions and the slightly elevated temperature inside the device enclosure, providing accurate data for both the external environment and the system's internal state.

4.1.3. MQ-135 Air Quality Sensor Response

The MQ-135 sensor's response to various gases was qualitatively assessed. The sensor exhibited distinct changes in its analogue output when exposed to carbon dioxide (exhaled breath), alcohol, and ammonia vapours, confirming its ability to detect a range of airborne pollutants.

Discussion: While the MQ-135 is effective as a qualitative indicator of air quality deterioration, its output is influenced by temperature, humidity, and the presence of multiple gases. This cross-sensitivity is a known limitation of low-cost metal-oxide semiconductor (MOS) sensors. Therefore, the data from this sensor is best interpreted as a general Air Quality Index (AQI) rather than as a precise measurement of the concentration of a specific gas.

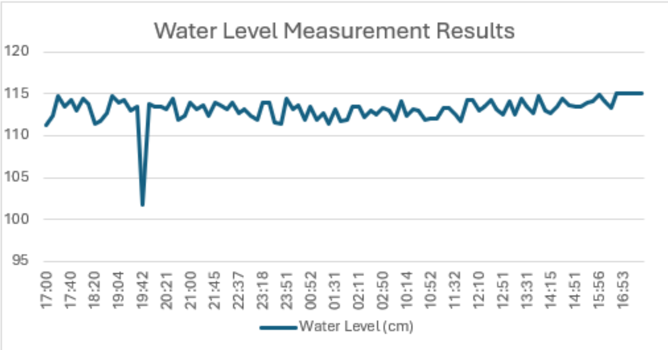


Fig 4. HC-SR04 Ultrasonic Sensor results

4.2. System Integration and Operational Challenges

The integration of the sensor modules with the dual-microcontroller architecture and cloud platform revealed a critical challenge.

4.2.1. Serial Communication Reliability

Over a continuous 24-hour operational assessment, the integrated system demonstrated a high rate of successful data transmission, successfully delivering 97.7% of all acquired data points to the cloud-based Firebase database. However, a detailed analysis revealed that 2.3% of the total readings—corresponding to 23 out of 1008 data points—were recorded as null values, specifically manifesting as zero parts per million (ppm) readings from the MQ-135 sensor. A subsequent diagnostic investigation conclusively determined that these anomalies were not attributable to sensor malfunctions but rather originated from intermittent failures within the serial communication (UART) interface linking the Arduino Uno and the ESP32 microcontrollers.

The root cause analysis identified three primary technical vulnerabilities contributing to the communication breakdown. First, despite the implementation of a logic-level shifter, transient instabilities between the 5V (Arduino) and 3.3V (ESP32) logic domains were hypothesized to occasionally corrupt data packets during transmission. Second, the absence of hardware flow control mechanisms (CTS/RTS) meant that the ESP32, when preoccupied with resource-intensive network operations, could not signal the Arduino to halt transmission, resulting in buffer overflows and consequent packet loss. Third, the communication protocol employed a simple string-based packet structure that lacked robust error-checking features, such as cyclic redundancy check (CRC) checksums, thereby preventing the ESP32 from detecting corrupted data and requesting retransmission.

This finding carries significant implications for embedded systems design, as it underscores a frequently underestimated challenge in multi-microcontroller architectures. The results effectively shift the focus from individual sensor performance to the critical importance of designing resilient communication protocols for ensuring data integrity. The identified shortcomings provide clear direction for future iterations, emphasizing the need for enhanced electrical isolation, hardware flow control implementation, and sophisticated packet framing with error detection to achieve the reliability required for mission-critical environmental monitoring applications.

4.2.2. Cloud Integration and Dashboard Performance

For 97.7% of successful transmissions, the Firebase Realtime Database and the web dashboard performed flawlessly. Data was updated in near-realtime (latency of < 2 seconds), and the dashboard provided a straightforward, accessible interface for monitoring environmental conditions remotely.

Discussion: The successful integration with Firebase demonstrates the practicality of using low-cost, cloud-based solutions for IoT applications. It provides a scalable and user-friendly backend, eliminating the need for complex private server infrastructure.

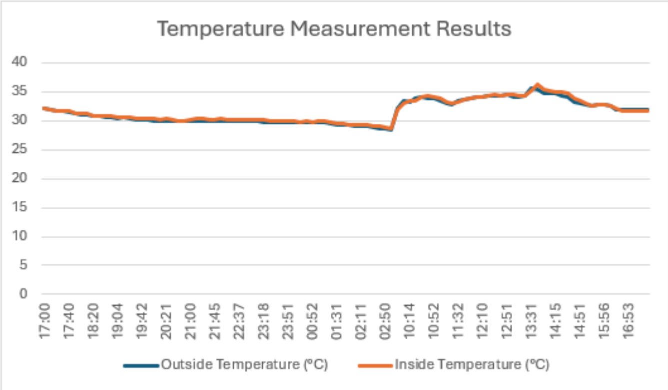


Fig 5. DHT22 Sensor results

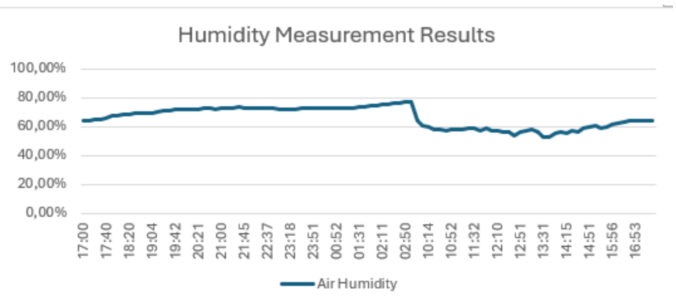


Fig 6. DHT22 Sensor Humidity results

4.3. Overall System Functionality

Despite the identified intermittent serial communication issue, the core functional objectives of the integrated monitoring system were successfully validated during the testing process. The Arduino Uno demonstrated robust performance in simultaneously reading and processing data from all three sensor modules—HC-SR04, as shown in Figure 4, DHT22 in Figures 5 and 6, and MQ-135 in Figures 7 and 8—confirming its efficacy as a dedicated data acquisition unit. Concurrently, the ESP32 reliably managed Wi-Fi connectivity and cloud synchronization tasks, successfully transmitting many data packets to the Firebase database. All local actuators, including the LCD, buzzer, and status LEDs, function correctly when activated by predefined threshold triggers, providing both local and remote alert capabilities. Furthermore, the system's operation is powered solely by a 12V DC battery confirms its energy autonomy and practical potential for deployment in field locations lacking permanent power infrastructure.

The developed prototype thus successfully serves as a proof of concept for a low-cost, integrated environmental monitoring system. The encountered communication challenge does not invalidate the overarching architectural design; instead, it provides a clear and valuable direction for targeted future improvement. This process of identifying and diagnosing system limitations through iterative testing is a fundamental and expected aspect of engineering design refinement. The insights gained into the critical need for robust inter-device communication protocols significantly contribute to the development methodology for reliable multi-processor IoT systems, establishing a solid foundation for subsequent hardware and firmware enhancements aimed at achieving higher technology readiness levels.

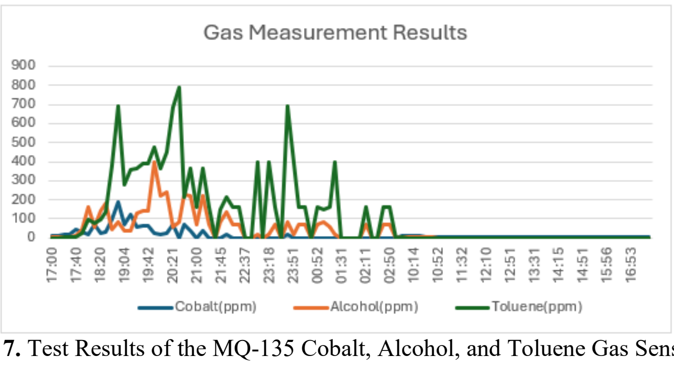


Fig 7. Test Results of the MQ-135 Cobalt, Alcohol, and Toluene Gas Sensors

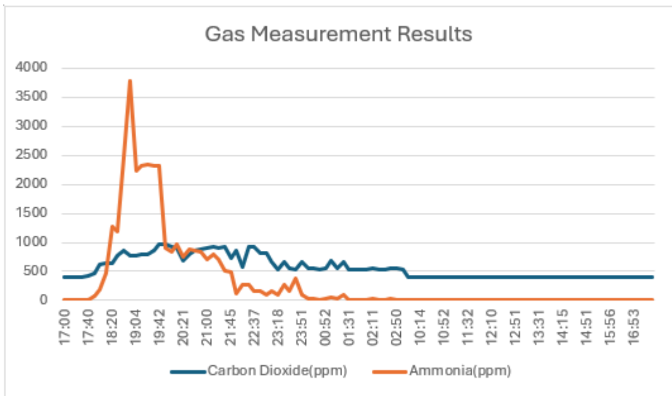


Fig 8. MQ-135 Carbon Dioxide and Ammonia Gas Sensor Test Results

5. Conclusion

This study successfully designed and constructed a prototype for an integrated flood and air pollution detection system based on IoT technology. The prototype effectively combined an Arduino Uno as a data acquisition unit and an ESP32 as a network gateway, integrating an HC-SR04 ultrasonic sensor, a DHT22 temperature and humidity sensor, and an MQ-135 air quality sensor. The system demonstrated core functionality in reading environmental data, processing it locally, and transmitting it to a cloud-based Firebase database for realtime monitoring via a web dashboard.

The results confirmed the high reliability and accuracy of the individual sensor modules, with the HC-SR04 and DHT22 performing within their specified tolerances. However, the integration process revealed a significant challenge in the form of intermittent serial communication failures between the microcontrollers, resulting in a 2.3% data loss rate. This underscores the critical importance of robust communication protocols, including hardware flow control and error-checking mechanisms, in multi-processor embedded systems.

While the system achieved its primary objective of proving the concept of an integrated monitoring device, the identified communication issue currently limits its reliability for mission-critical deployments. Therefore, the prototype is best classified at a Technology Readiness Level (TRL) of 4-5, representing validation in a controlled environment. Future work must focus on implementing hardware fixes—such as galvanic isolation and improved level-shifting circuits—and software enhancements, including CRC checksums and a watchdog timer, to elevate the system to a higher TRL suitable for field testing. This research provides a valuable foundation and a clear roadmap for developing more robust, low-cost, and integrated environmental monitoring solutions.

Acknowledgements

The authors would like to express their profound gratitude to the LPPM Universitas Negeri Jakarta, RIIM Competition for Research and Innovation (Risnov) - BRIN with contract number: 127/IV/KS/07/2025 dan B/968/UN39.14/HK.07/VIII/2025 and the Indonesia Endowment Fund for Education (LPDP) for their financial support and facilitation, which were instrumental in the completion of this work.

References

[1] W. Handayani, U. E. Chigbu, I. Rudiarto, and I. H. S. Putri, "Urbanization and increasing flood risk in the northern coast of Central Java—Indonesia: An assessment towards better land use policy and flood management," *Land*, vol. 9, no. 10, p. 343, 2020. doi: 10.3390/land9100343.

[2] H. Setiyono, A. N. B. Bambang, M. Helmi, and M. Yusuf, "Effect rainfall season on coastal flood in Semarang City, Central Java, Indonesia," *Int. J. Health Sci.*, vol. 6, no. S1, pp. 7584–7595, 2022. doi: 10.53730/ijhs.v6nS1.6618.

[3] D. A. Retnowati, R. Virtriana, and A. Riqqi, "Potential losses in rice production due to flooding on the northern coast of West Java, Indonesia," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 1472, no. 1, 012028, 2025. doi: 10.1088/1755-1315/1472/1/012028.

[4] T. Satoshi and L. Charlotte, "Deforestation and its role in climate change and soil degradation," 2025.
M. Leon, G. Cornejo, M. Calderón, E. González-CarriónE, H. Florez, "Effect of Deforestation on Climate Change: A Co-Integration and Causality Approach with Time Series," *Sustainability*. 2022; 14(18):11303. https://doi.org/10.3390/su141811303

[5] M. Rahman, "Climate change and environmental degradation: A serious threat to global security," *Eur. J. Social Sci. Stud.*, vol. 8, 2023. doi: 10.46827/ejsss.v8i6.1493.

[6] B. Haupt, "The use of crisis communication strategies in emergency management," *J. Homeland Secur. Emerg. Manage.*, vol. 18, 2021. doi: 10.1515/jhsem-2020-0039.

[7] P. Charan, M. M. Siddiqui, V. Yadav, C. Pathak, Y. Narayan and Z. H. Khan, "AI-Enhanced Early Warning Systems for Natural Disaster Detection and Mitigation using Wireless Sensor Networks," *2024 Second International Conference Computational and Characterization Techniques in Engineering & Sciences (IC3TES)*, Lucknow, India, 2024, pp. 1-6, doi: 10.1109/IC3TES62412.2024.10877562.

[8] P. Rosyady, D. Yulianto, and F. Warsino, "IoT-based home water monitoring using Arduino," *Mobile Forensics*, vol. 3, pp. 75–84, 2021. doi: 10.12928/mf.v3i2.5517.

[9] P. A. Rosyady, D. Yulianto, and F. Warsino, "IoT-based home water monitoring using Arduino," *Mobile Forensics*, vol. 3, no. 2, pp. 75–84, 2021. doi: 10.12928/mf.v3i2.5517.

[10] T. Andriani *et al.*, "Design of flood early detection system using WeMos D1 Mini ESP8266 IoT technology," *J. Phys. Sci. Eng.*, vol. 4, pp. 67–73, 2020. doi: 10.17977/um024v4i22019p067.

[11] A. Tarpanelli, A. C. Mondini, and S. Camici, "Effectiveness of Sentinel-1 and Sentinel-2 for flood detection assessment in Europe," *Nat. Hazards Earth Syst. Sci.*, vol. 22, pp. 2473–2489, 2022. doi: 10.5194/nhess-22-2473-2022.

[12] S. Jarrett and D. Hölbling, "Spatial evaluation of a natural flood management project using SAR change detection," *Water*, vol. 15,

no. 12, p. 2182, 2023. doi: 10.3390/w15122182.
[13] [A. Tarpanelli, A. C. Mondini, and S. Camici, “Effectiveness of Sentinel-1 and Sentinel-2 for flood detection assessment in Europe,” *Nat. Hazards Earth Syst. Sci.*, vol. 22, no. 8, pp. 2473–2489, 2022. doi: 10.5194/nhess-22-2473-2022.
[14] S. M. Abiduzzaman, H. Mansor, T. Gunawan, and R. Ahmad, “Realtime outdoor air quality monitoring system,” in *Proc. ICSIMA*, 2021, pp. 140–145. doi: 10.1109/ICSIMA50015.2021.9526332.
[15] M. G. Arkhan and Z. R. S. Elsi, “Air quality monitoring system based Internet of Things,” *Brilliance*, vol. 4, no. 2, pp. 669–673, Nov. 2024.
[16] A. Janarthanan, A. Paramarthalingam, A. Arivunambi and P. M. D. R. Vincent, "Realtime indoor air quality monitoring using the Internet of Things," *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, Kannur, India, 2022, pp. 99-104, doi: 10.1109/ICICICT54557.2022.9917990.
[17] F. Gonibala *et al.*, “Toward an advanced gas composition measurement device for chemical reaction analysis,” *Bul. Ilm. Sarjana Tek. Elektro*, vol. 5, pp. 525–538, 2023. doi: 10.12928/biste.v5i4.9249.
[18] M. A. Sugianto and Z. Zulfikar, “Design of a carbon monoxide gas measurement tool using microcontroller-based MQ-7 gas sensor in households,” *Newton: Netw. Inf. Technol.*, vol. 3, no. 3, pp. 36–47, 2024. doi: 10.32764/newton.v3i3.4881.
[19] M. Kumar *et al.*, “Air quality monitoring using MQ135 gas sensor and Arduino Uno,” *Int. J. Latest Technol. Eng. Manage. Appl. Sci.*, vol. 14, no. 5, pp. 1097–1101, 2025. doi: 10.51583/ijltemas.2025.140500119.
[20] M. Wajid *et al.*, “Flood prediction system using IoT & artificial neural network,” *VFAST Trans. Softw. Eng.*, vol. 12, pp. 210–224, 2024. doi: 10.21015/vtse.v12i1.1603.
[21] H. R. Arante *et al.*, “Development of a secured IoT-based flood monitoring and forecasting system using genetic-algorithm-based neuro-fuzzy network,” *Sensors*, vol. 25, p. 3885, 2025. doi: 10.3390/s25133885.
[22] T. Chai, D. Kim, and S. Shin, “Efficient internet of things communication system based on near-field communication and long range radio,” *Sensors*, vol. 25, no. 8, p. 2509, 2025. doi: 10.3390/s25082509.
[23] M. E. Gavira *et al.*, “Characterization and performance evaluation of ESP32 for realtime synchronized sensor networks,” *Procedia Comput. Sci.*, vol. 237, pp. 261–268, 2024. doi: 10.1016/j.procs.2024.05.104.
[24] H. Turkmanović, M. Karličić, V. Rajović, and I. Popović, “High performance software architectures for remote high-speed data acquisition,” *Electronics*, vol. 12, no. 20, p. 4206, 2023. doi: 10.3390/electronics12204206.
[25] H. Hasani, F. Freddi, R. Piazza, and F. Ceruffi, “A wireless data acquisition system based on MEMS accelerometers for operational modal analysis of bridges,” *Sensors*, vol. 24, no. 7, p. 2121, 2024. doi: 10.3390/s24072121.