EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 6 No. 1 (2025) 90-102
https://publisher.uthm.edu.my/periodicals/index.php/eeee

Real-time Flood Monitoring Alert System with Lora and IoT

Nurliyana Latifah Yahya[1], Mariyam Jamilah Homam[1]*

[1] *Department of Electrical Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, Malaysia*

*Corresponding Author: mariyam@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2025.06.01.013

Article Info

Received: 23 January 2025
Accepted: 23 March 2025
Available online: 9 May 2025

Keywords

Real-Time Flood Monitoring, ESP32 Microcontroller, ThingSpeak IoT Platform, Ultrasonic sensor, Temperature and humidity sensors, Water flow sensor

Abstract

This project focuses on developing a Real-Time Flood Monitoring and Alert System using LoRa and IoT technologies to enhance disaster preparedness in flood-prone areas, with the primary objective of continuously monitoring water levels, temperature, humidity, and water flow to provide timely alerts that mitigate flood risks. The methodology involves deploying an ultrasonic sensor, temperature and humidity sensors, and a water flow sensor connected to an ESP32 microcontroller, with data transmitted via LoRa to the ThingSpeak IoT platform for real-time monitoring. Testing at the UTHM pond yielded two key assessments: the monitored water levels, temperature, humidity, and water flow after rainfall, and the real-time data display on an OLED screen, confirming the system's functionality in providing immediate information to users. During moderate rainfall, the system successfully recorded a water level range of 25 cm to 26 cm, a temperature range of 27.73°C to 27.57°C, a humidity range of 87% to 92.31%, and a water flow range of 0 (an indication of no flow or movement detected). These results show how well the system detects environmental changes and how it may improve local authorities' ability to make decisions, eventually enhancing community safety in flood-prone areas. In conclusion, the project demonstrates a flood monitoring system, and it is suggested that the sensor network be expanded in the future and that mobile applications be used to increase user accessibility.

1. Introduction

Water is a fundamental requirement in everyday life, but excessive and inaccessible water conditions can lead to floods, particularly in Malaysia during the northeast monsoon season. Floods, often classified as disasters, can result from activities such as overflowing rivers and intense rainfall. These events highlight the need for effective flood monitoring systems to better prepare for floods and reduce flood-related hazards.

This project aims to develop a monitoring system for sensing water levels in flood-prone areas. Effective early warning systems for impending floods present significant challenges for disaster preparedness and response. Several studies have explored IoT-based flood monitoring solutions to address these challenges. For instance, Zahir et al. [1] proposed a smart IoT flood monitoring system, while Gomathy et al. [2] emphasized the importance of IoT-based flood detection management. Hassan et al. [3] highlighted the role of IoT systems for water level monitoring, and Chari et al. [4] explored an IoT-based flood alerting system using Raspberry Pi.

Despite advancements, limitations persist in existing solutions. Issues such as limited weather forecast information, inconsistent internet connectivity, SMS character limits, sensor malfunctions, power outages, and poor weather conditions impacting sensor accuracy have been documented [5], [6]. Additionally, the absence of

GPS sensors and GSM connectivity poses challenges for reliable location tracking and warning response times [7]. This project addresses these gaps by leveraging IoT and LoRa technologies to create a robust flood monitoring and alerting system.

The objectives of this project include developing a monitoring system for sensing water levels in flood-prone areas, collecting data from sensors, and analyzing water levels, flow, temperature, and humidity for the use of precautionary measures. The scope of the study includes testing and monitoring the UTHM pond, using the ThingSpeak application to visualize water levels, water flow, temperature, and humidity values in flood-prone areas, and collecting continuous readings from all sensors. Notifications will be triggered when any reading exceeds predetermined thresholds.

2. Methodology

Fig. 1 shows the process for completing this study, which is illustrated in the flowchart above. The procedure begins with the initialised power source, which can be battery or solar-powered. The ESP32 microcontroller on the transmitter side is powered on and initialized. The water flow sensor, ultrasonic sensor, temperature sensor, and humidity sensor are then sequentially read by ESP32. Once all sensor data is gathered, the ESP32 uses the LoRa communication protocol to send this data.

An additional ESP32 microcontroller is initialized on the receiving end to prepare to receive the data. The ESP32 shows the data on an OLED screen for instant viewing after receiving data over LoRa. Furthermore, an IoT platform, Think Speak, receives data and uses it for additional analysis, storage, and remote monitoring. After all, the data has been successfully transmitted, received, presented, and stored.

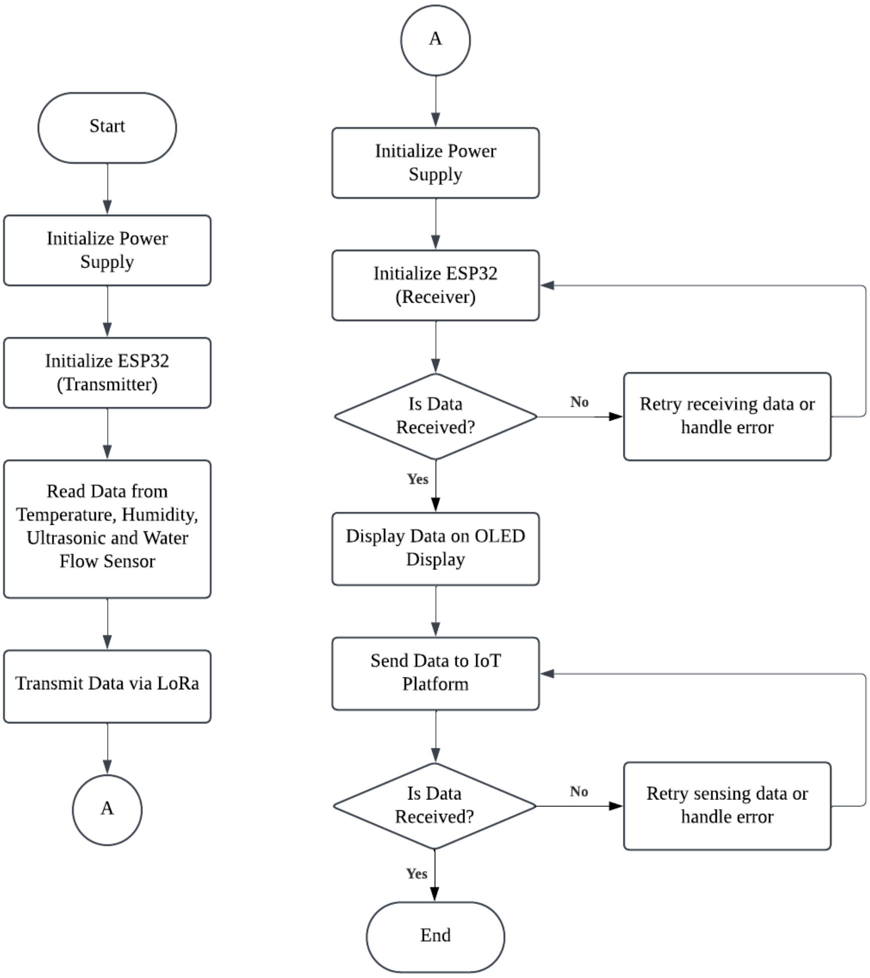


Fig. 1 *General Flowchart of this project*

Fig.2 shows the block diagram of the system that detects floods by observing natural factors such as humidity, temperature, water level, and flow level. It uses various sensors to collect data on these variables. The AHT20+BMP280 sensor monitors temperature and humidity changes. The water flow sensor consists of a plastic valve body, a water rotor, and a hall-effect sensor. The HC-SR04 Ultrasonic Range Finder Distance Sensor measures distances using ultrasonic waves. All sensors are connected to an ESP32 microcontroller for data processing and storage. The system also has a Wi-Fi feature for easy access to the system and its data over the Internet of Things (IoT), as shown in Fig. 3. The system's functionality ensures accurate and timely detection of potential floods.

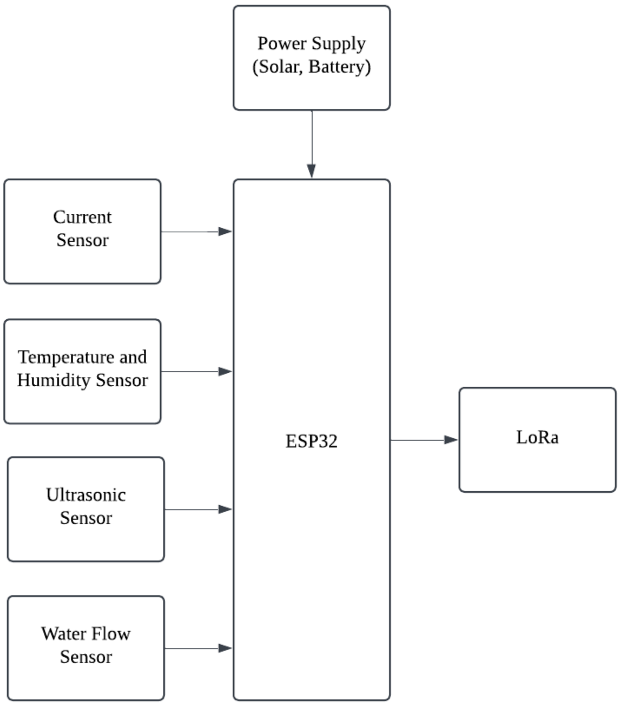


Fig. 2 *Block Diagram of Transmitter*

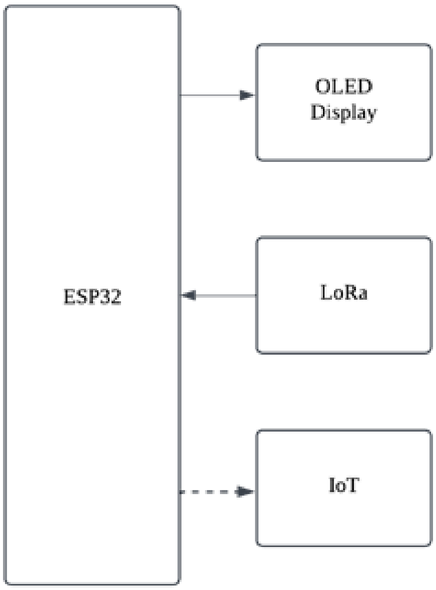


Fig. 3 *Block Diagram of Receiver*

The project uses an ESP32 microcontroller to measure water levels, water flow, temperature, and humidity, as shown in Fig. 4. An ultrasonic sensor is connected to the ESP32, and a water flow sensor collects data on water flow direction. The output is displayed on an OLED screen through VCC, GND, SDA, and SCL pins. The AHT20+BMP280 measures temperature and humidity through pins 3.3V, GND, SDA, and SCL. The INA219 DC sensor is connected to the ESP32. The receiver uses an ESP32 microcontroller, and an OLED screen displays the outputs. The ESP32 is integrated with pins 3.3V, GND, SDA, and SCK for accurate readings, as shown in Fig. 5.

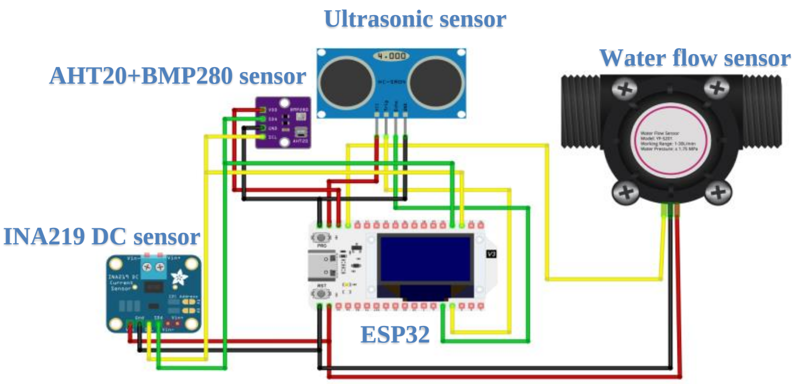


Fig. 4 *Circuit design of transmitter*

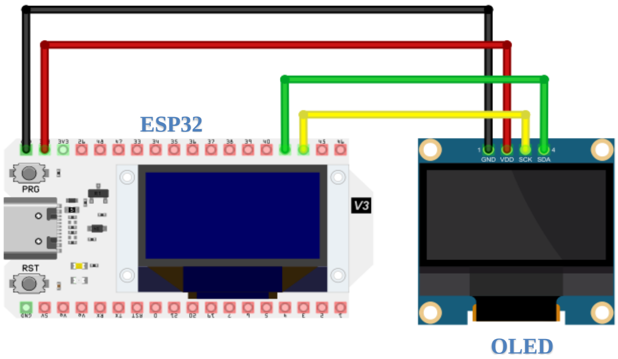


Fig. 5 *Circuit design of receiver*

Fig. 6 shows the design prototype for the transmitter. It aims to monitor the water level in a flood-prone area. The system uses ultrasonic sensor to measure water level, AHT20+BMP280 to measure temperature and humidity, and a water flow sensor to determine the flow direction. The inside of the transmitter prototype has the sensors and a battery that are connected to the microcontroller, and on the side of the prototype, there is an antenna that enables wireless communication.

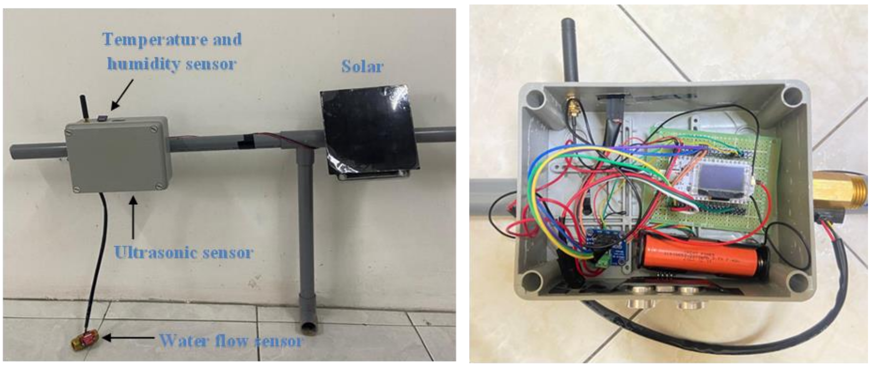


Fig. 6 *Prototype design of transmitter*

Fig. 7 shows the design prototype for the receiver. It displays real-time data on OLED and alerts or acts when the water level is high. Next, Fig. 6 shows the inside of a prototype receiver box. The ESP32 microcontroller was responsible for receiving and processing data from the transmitter. An OLED display is for data visualization and an antenna is to enable wireless communication.

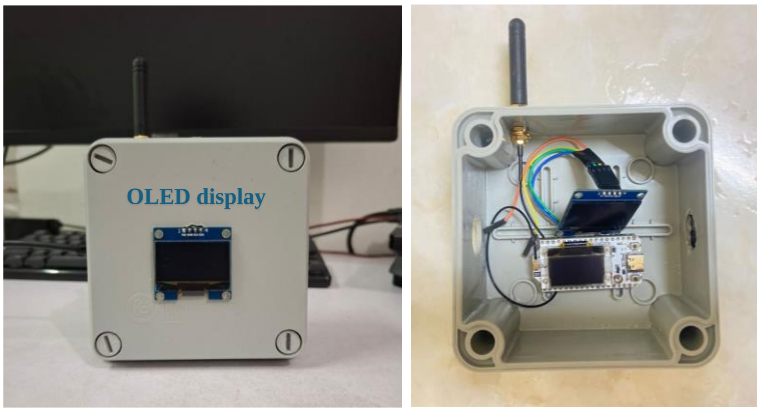


Fig. 7 *Prototype design of receiver*

Fig. 8 shows the prototype at UTHM pond (in front of FPTV). The setup consists of a sensor connected to a junction box positioned on a PVC pipe, and the prototype is placed near the water. Furthermore, this configuration has enabled the prototype to monitor water levels and evaluate its performance in a real-world environment.

Fig. 8 *Location the prototype at UTHM pond (in front of FPTV)*

3. Result and Discussion

Testing and experiments were conducted at UTHM pond (in front of FPTV) to create a variety of environments, ensuring comprehensive data collection and analysis. By utilizing different settings, researchers aimed to observe how certain conditions influenced the outcome, thereby gaining a deeper understanding of the subject under investigation.

3.1 Dashboard

ThingSpeak is essentially a dashboard application used for this work. It provides a simple and effective way to collect, visualize, and analyse real-time and historical data from respective sensors and devices. Furthermore, it allows to create the custom graphs and widgets to monitor system performance, set up alerts for critical events, and access data remotely from any internet-connected device. ThingSpeak cloud-based storage and interaction with programs like MATLAB provide fast data management and advanced analysis, resulting in a powerful yet simple solution for this project.

3.1.1 ThingSpeak

Fig. 9 shows the ThingSpeak channel setup, which allows to be configured up to 8 fields to monitor specific data streams, such as temperature, humidity, and other sensor readings. It can name the channel, add a description, and enable or disable fields based on the project requirements. However, the channel has some limitations, its free accounts allow for a maximum of channels. The data will update every 20 seconds and it will be stored for up to one channel before being overwritten.

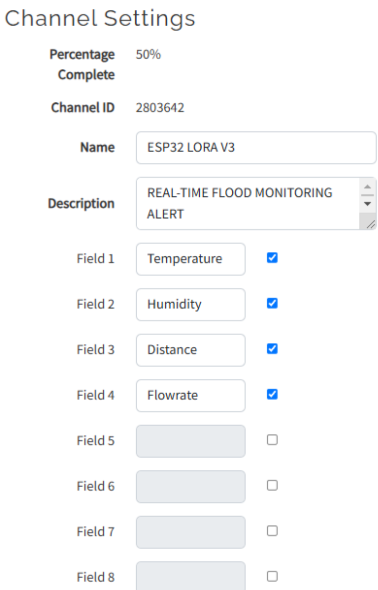


Fig. 9 *ThingSpeak Channel Setup*

Fig. 10 shows the field charts in ThingSpeak to display the real-time and historical data from the specific field. It also allows to monitor trends and analyze patterns over time. The chart typically includes a time-based x-axis to show timestamps of data entries, and a y-axis represents the data values for the selected parameters. The chart will update every 20 seconds automatically when a new date is added, ensuring that the latest information is always visible.

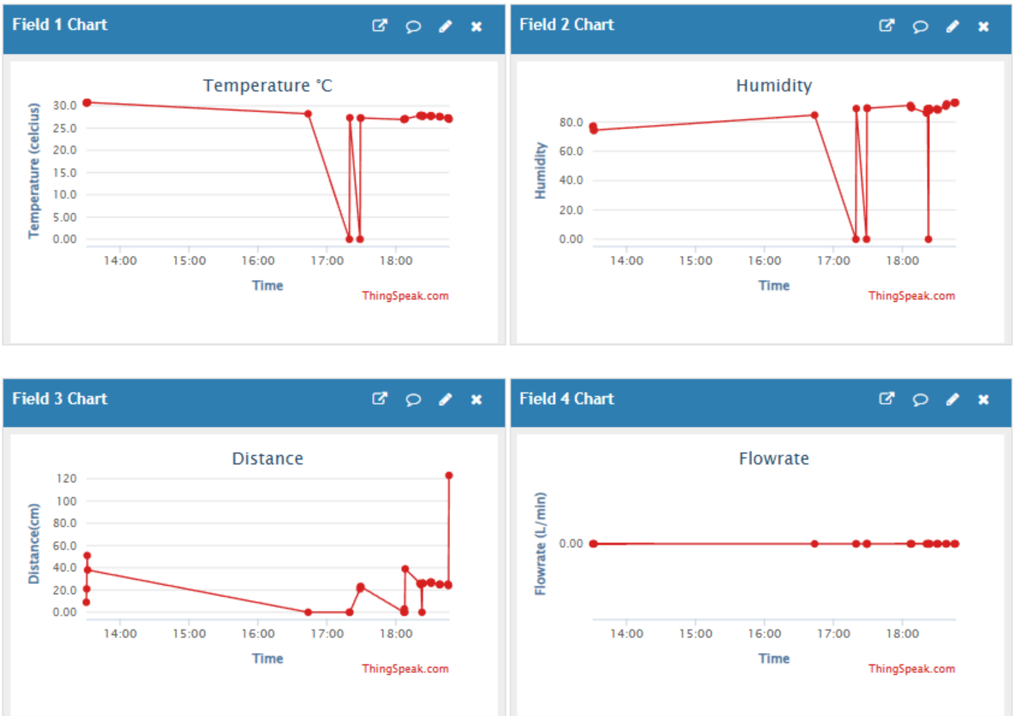


Fig. 10 *Field chart of ThingSpeak*

3.2 Testing and Experiment

Testing and experiments were conducted at UTHM pond (in front of FPTV). By utilizing different settings, researchers aimed to observe how certain conditions influenced the outcome, thereby gaining a deeper understanding of the subject under investigation.

3.2.1 Testing 1: Monitoring and Collecting Data After Rain Using ThingSpeak

The system was tested for its ability to collect data and send it to ThingSpeak for real-time analysis. The sensors measure water level, water flow, temperature, and humidity, and are processed by an ESP32 microcontroller. The data was sent using LoRa for communication and displayed on the ThinSpeak dashboard. The test was conducted after rainfall to simulate real-world conditions. Table 1 displays temperature results collected at specific timestamps, with each row displaying a timestamp and corresponding chart. ThingSpeak display visualizes temperature changes over time, allowing easy monitoring and analysis. Table 2 shows humidity results collected on the ESP32 microcontroller and displayed on ThingSpeak. Readings are taken every minute, showing frequent monitoring and the latest update time.

Table 1 *Temperature reading from ThingSpeak*

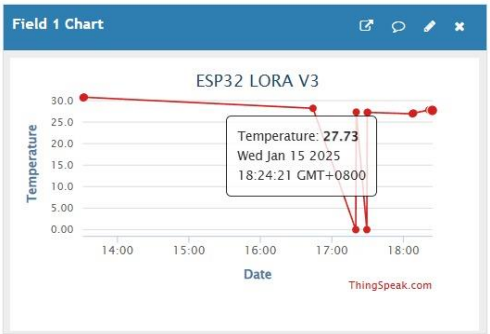
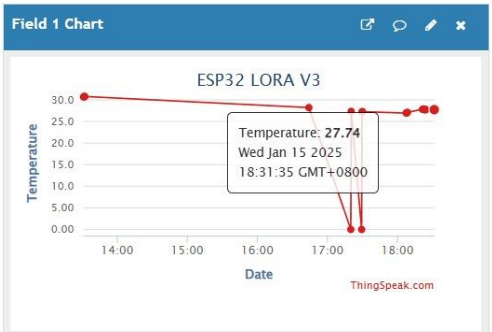
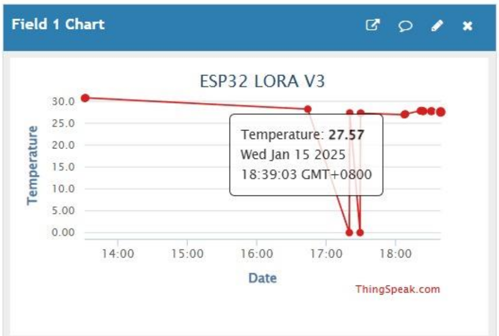
Timestamp	Dashboard
2025-01-15 18:24	
2025-01-15 18:31	
2025-01-15 18:39	

Table 2 *Humidity reading from ThingSpeak*

Timestamp	Dashboard
2025-01-15 18:24	

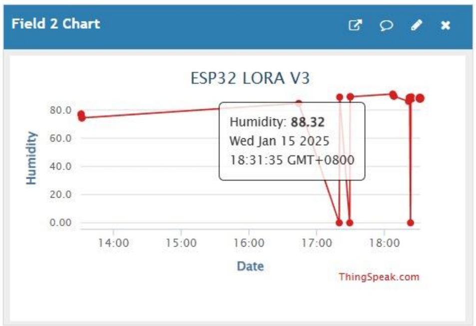
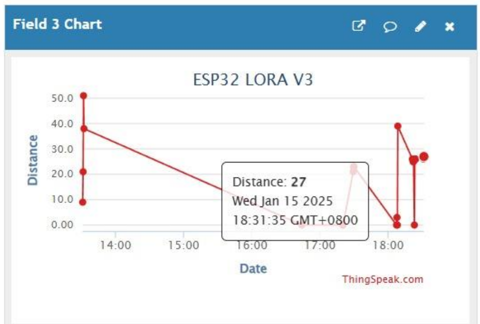
2025-01-15 18:31	
2025-01-15 18:39	

Table 3 shows a slight decrease in water level readings, indicating no operation at 0 cm. However, a rise in water level can be achieved by measuring the sensor's distance to the water's surface. Table 4 shows the flow rate readings remain constant at 0, indicating no movement or flow detected. Factors such as sensor placement and environmental conditions impact perception, highlighting the importance of considering these factors.

Table 3 *Water level reading from ThingSpeak*

Timestamp	Dashboard
2025-01-15 18:24	
2025-01-15 18:31	

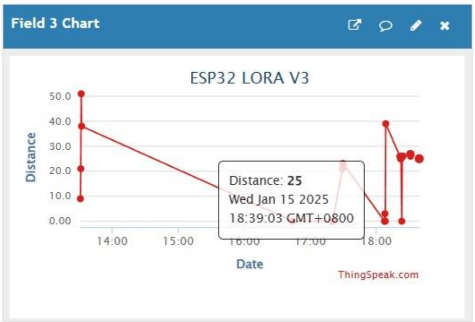
2025-01-15 18:39	

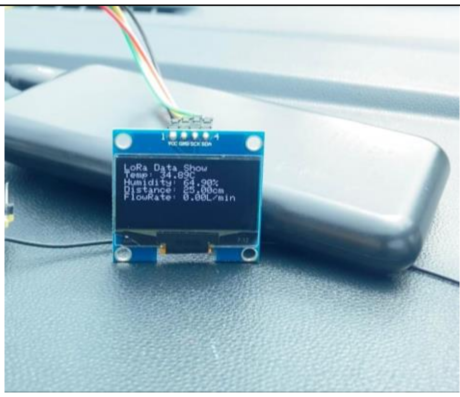
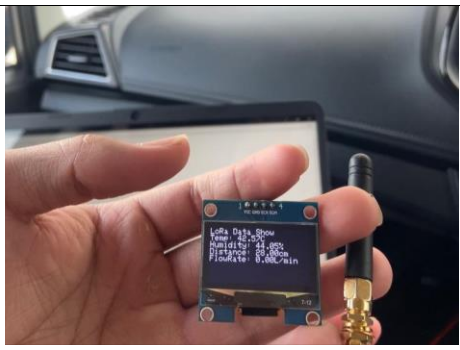
Table 4 *Water flow reading from ThingSpeak*

Timestamp	Dashboard
2025-01-15 18:24	
2025-01-15 18:31	
2025-01-15 18:39	

3.3 Testing 2: Monitoring and Receiving Data at Different Locations Using OLED

Table 5 shows the testing of an OLED display for real-time data monitoring and display at three locations: FPTV, FKEE, and F2. The setup includes an ESP32 microcontroller connected to the display. LoRa processed data from sensors and sent it to the ESP32 receiver. The display updated as data values changed, confirming the system's functionality. The OLED display displays water level, water flow, temperature, and humidity, providing a comprehensive view of monitored conditions. However, some data was not received by ThingSpeak at certain locations, indicating potential data transmission failure.

Table 5 *Monitor and display data on an OLED at different locations*

<table>
<tr><th>Location</th><th>Result from OLED</th></tr>
<tr><td>Front of FPTV</td><td>
</td></tr>
<tr><td>Front of FKEE</td><td>
</td></tr>
<tr><td>Front of F2</td><td>
</td></tr>
</table>

4. Conclusion

This project successfully developed the Real-Time Flood Monitoring Alert System with LoRa and IoT, achieving its primary objective to continuously collect and analyze data on water level, water flow, temperature, and humidity. This proactive strategy makes it easier to identify possible flooding early on, allowing for prompt warnings and risk-reduction precautions. The system's integration of real-time data analysis improves communities' and local authorities' ability to make well-informed decisions, which in turn increases safety and resilience against flooding. It also creates a foundation for future growth in disaster preparedness and environmental monitoring. The ESP32 microcontroller and various sensors efficiently monitor environmental variables and water levels in flood-prone areas. The interface with the ThingSpeak IoT platform enabled real-time visualization and precise data collecting. The UTHM pond's testing verified the system's ability to react to water flow and levels variations. Additionally, the results have shown that this system may improve flood management efforts and community safety by providing timely and accurate information for decision-making despite certain difficulties, such as sensor wiring problems and communication range constraints.

In future work, it is recommended that sensor calibration processes be enhanced to improve accuracy, the sensor network be expanded to include additional environmental parameters such as soil moisture and rainfall and alternative communication technologies be investigated to address range limitations. Creating a specialized mobile application or enhancing the ThingSpeak platform's user interface will increase accessibility for community members and local authorities, and machine learning techniques could aid in more accurate flooding event prediction. To validate system performance, extensive field testing in different flood-prone areas is necessary. Community training programs should be created to involve users effectively. Lastly, creating long-term sustainability plans, including maintenance plans and collaborations with neighbourhood organizations, will guarantee the Real-Time Flood Monitoring Alert System's continued effectiveness and influence.

Acknowledgement

The authors gratefully acknowledge the support of the Faculty of Electrical and Electronic Engineering at Universiti Tun Hussein Onn Malaysia.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the paper's publication.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Nurliyana, Mariyam Jamilah;* ***data collection:*** *Nurliyana;* ***analysis and interpretation of results:*** *Nurliyana, Mariyam Jamilah;* ***draft manuscript preparation:*** *Nurliyana, Mariyam Jamilah. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] S. B. Zahir *et al.*, “Smart IoT Flood Monitoring System,” *Journal of Physics Conference Series*, vol. 1339, no. 1, pp. 012043–012043, Dec. 2019, doi: https://doi.org/10.1088/1742-6596/1339/1/012043.
[2] Dr. C. K. Gomathy*, Ms. G. G. L. Priya, and Mr. H. Kumar K N, “A Study on IoT Based Flood Detection Management System,” *International Journal of Engineering and Advanced Technology*, vol. 10, no. 4, pp. 130–133, Apr. 2021, doi: https://doi.org/10.35940/ijeat.d2407.0410421.
[3] H. Hassan, M. I. Q. Mazlan, T. N. T. Ibrahim, and M. F. Kambas, “IOT System: Water Level Monitoring for Flood Management,” *IOP Conference Series: Materials Science and Engineering*, vol. 917, no. 1, p. 012037, Sep. 2020, doi: https://doi.org/10.1088/1757-899x/917/1/012037.
[4] K. S. Chari, Maturi Thirupathi, and Ch Hariveena, “RETRACTED: IoT-based Flood Monitoring and Alerting System using Raspberry Pi,” *IOP Conference Series Materials Science and Engineering*, vol. 981, no. 4, pp. 042078–042078, Dec. 2020, doi: https://doi.org/10.1088/1757-899x/981/4/042078.
[5] N. M. Zain, L. S. Elias, Zulfikri Paidi, and M. Othman, “Flood Warning and Monitoring System (FWMS) using GSM Technology,” *Journal of Computing Research and Innovation*, vol. 5, no. 1, pp. 7–18, Oct. 2020, doi: https://doi.org/10.24191/jcrinn.v5i1.158.
[6] Nandhini N, Lavanya N, and Kowshika V, “Flood Monitoring and Alerting System Using Arduino In IoT,” *SSRN Electronic Journal*, Jan. 2021, doi: https://doi.org/10.2139/ssrn.3852091.
[7] Nor and M. M. Kamal, “The development of smart flood monitoring system using ultrasonic sensor with blynk applications,” pp. 1–6, Nov. 2017, doi: https://doi.org/10.1109/icsima.2017.8312009.