International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Real-Time IoT-Based Flood Detection Using ESP32 a Scalable and Low-Power Solution for Risk Mitigation

Justin Phinni Jose[1], Jogimol Joseph[2]

[1]Department of Computer Applications, Musaliar College of Engineering & Technology, Pathanamthitta, Kerala, India
Email: *juztinjoz2002[at]gmail.com*

[2]Professor, Department of Computer Applications, Musaliar College of Engineering & Technology, Pathanamthitta, Kerala, India

Abstract: ***The ESP32 microcontroller is used in this study's Internet of Things-based flood detection system to allow for real-time environmental state monitoring. The system combines flow, temperature, humidity, ultrasonic, raindrop, and water level sensors. Data is sent over Wi-Fi to the ThingSpeak cloud platform for analysis and visualization. Authorities and people receive early warnings from automated alerts that are based on sensor thresholds. High sensor precision, dependable data transmission, and low power consumption are confirmed by experimental findings. Scalability and deployment in various flood-prone areas are supported by the modular architecture. Future developments include real-world field testing, alternative communication protocols like LoRaWAN, and AI-driven predictive analytics, providing a reliable and affordable flood risk reduction solution.***

Keywords: Flood Monitoring, IoT, Water Level Sensor, ESP32, Real-Time Data

1. Introduction

The Floods are some of the most destructive natural events we face, often leaving behind widespread damage to homes, infrastructure, farmland, and, most critically, putting lives at risk. With climate change and rapid urban growth making floods more frequent and intense, it's more important than ever to have reliable systems in place that can monitor and manage these risks in real time.

Traditional flood monitoring methods can fall short-they're often slow, labor-intensive, and lack the ability to provide real-time updates, especially in remote or high-risk areas. That's where modern technology steps in. The Internet of Things (IoT) offers a smarter way to keep track of changing environmental conditions through connected, automated systems.

In this project, we developed an IoT-based flood monitoring system that's both cost-effective and scalable. It uses a combination of sensors-ultrasonic, water level, and flow-to continuously measure and report on water conditions. These sensors feed data to an ESP32 microcontroller, which sends the information over Wi-Fi to a cloud platform where it can be monitored from anywhere. When water levels or flow rates rise beyond safe limits, the system can instantly send alerts via SMS, email, or mobile apps-giving communities and authorities the precious time they need to act.

By combining real-time data with accessible technology, this system aims to make flood response faster and more effective-ultimately helping to protect people, property, and the places we live.

2. Literature Survey

Flood monitoring has become a critical area of research, especially with the growing threat posed by climate change and the increasing need to prepare for disasters. Traditional methods like manual water level checks or large-scale hydrological models are often slow and lack the real-time data needed for quick and effective responses. With this in mind, researchers have turned to IoT technologies, which offer a potential solution to improve the speed and accuracy of flood detection.

IoT in Flood Monitoring: A number of studies have explored how IoT can help with real-time environmental monitoring. For example, Sharma et al. (2019) developed a low-cost water level detection system using an Arduino board and ultrasonic sensors to track river levels. This system worked well for smaller environments but didn't include flow rate measurements or an alert system, limiting its usefulness in more complex scenarios. Similarly, Ali et al. (2020) built an IoT-based flood warning system that sent water level data to the cloud using GSM modules. While it enabled remote monitoring, it didn't use multiple sensors to provide a comprehensive picture of the flood situation. [15]

Sensor-Based Systems: Ultrasonic sensors, which measure the distance to the water's surface, are often used in flood monitoring systems because they are reliable in various environmental conditions. Other types of water level sensors, like capacitive and float-based sensors, provide more direct depth measurements. Flow sensors, which track the speed and volume of water, are less commonly used in basic flood systems but are crucial for detecting overflow or unexpected water movements. Many existing systems focus on only one or two types of sensors, which limits their ability to capture the full scope of a flood event.

Cloud Integration and Alerts: Cloud platforms such as ThingSpeak, Firebase, and Blynk have made it easier to collect, store, and visualize data in real-time. Kumar et al. (2021) developed a system that used these platforms to provide live water level data and send SMS alerts. While these systems were effective, they often required expensive hardware and weren't designed for large-scale deployments, especially in rural or flood-prone areas where affordability and accessibility are key concerns [6].

Challenges in Existing Systems: Despite the advances, existing flood monitoring systems still face several challenges. These include limited sensor coverage, the absence of predictive analytics, reliance on mobile networks that may not be reliable in remote locations, and high setup costs. Many systems also lack user-friendly interfaces for easy access to real-time data and customizable alerts, making it difficult for non-experts to use the systems effectively in emergency situations.

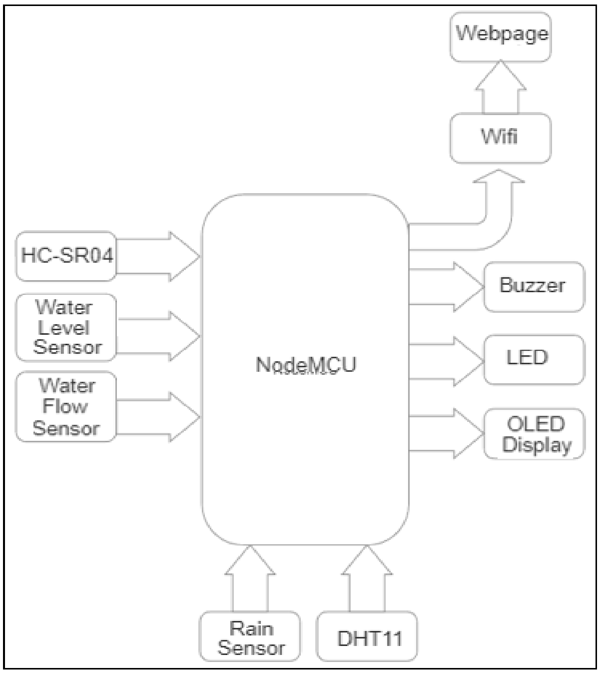
3. Methodology

This section describes the approach taken to design, build, and assess the IoT-based flood detection system proposed in this project. It covers key aspects such as system architecture, choice of sensors, integration of hardware and software components, data communication, and the testing methods used to ensure effective real-time flood monitoring and alert generation. The methodology is divided into the following key stages:

a) System Architecture Design

This stage involves planning the overall structure of the flood monitoring system. It includes defining how different components (like sensors, the microcontroller, and cloud services) will interact. The ESP32 microcontroller acts as the central hub, collecting and processing sensor data before sending it to the cloud in real-time.

b) Sensor Selection & Integration



After The system incorporates a variety of sensors to gather detailed environmental data relevant to flood conditions. The water level sensor monitors the depth of water in locations such as rivers, reservoirs, and drainage systems. The ultrasonic sensor provides a non-contact method for measuring the distance to the water's surface, helping to detect rising water levels. A raindrop sensor is used to identify the presence and intensity of rainfall, while the DHT11 sensor captures temperature and humidity levels, offering insights into weather patterns that could contribute to flooding. Additionally, the flow sensor measures the rate and volume of water movement, which is essential for detecting potential overflow scenarios. All sensors are properly calibrated and integrated with the ESP32 microcontroller, ensuring accurate data collection and reliable system performance.

c) Microcontroller Programming

During this stage, the ESP32 microcontroller is programmed with custom code to manage the system's operations. The code is responsible for collecting data from all connected sensors, processing or filtering the readings as necessary, and establishing a Wi-Fi connection for internet access. Once processed, the data is transmitted to the ThingSpeak cloud platform for real-time monitoring and visualization. This programming ensures that the ESP32 continuously gathers and sends environmental data without interruption

d) Cloud Platform Setup

In this stage, the ThingSpeak cloud platform is set up to collect and store data transmitted by the ESP32. It offers a real-time dashboard that visually presents sensor readings using graphs, charts, and gauges. This enables both users and authorities to access and monitor flood-related information remotely from any internet-connected device, ensuring timely and informed decision-making.

e) Threshold Configuration & Alerts

Upon Threshold values are predefined for each sensor to identify potential flood risks. When any sensor detects readings that exceed these safe limits-such as a sudden rise in water level-the system is programmed to instantly send alert notifications through SMS, email, or mobile applications. This automated alert mechanism helps ensure quick and effective responses to emerging flood threats.

f) System Testing & Evaluation

Once The system is thoroughly tested under simulated flood conditions to evaluate its performance and reliability. This includes checking the accuracy of sensor readings, assessing the responsiveness of real-time data updates, and ensuring that alerts are properly triggered when sensor values exceed their predefined thresholds. These evaluations are crucial for fine-tuning the system before it is deployed in actual flood-prone areas.

g) Power and Network Optimization

To ensure the system operates effectively in remote or rural areas, the ESP32 and sensors are optimized for minimal power usage, potentially utilizing battery or solar power. The Wi-Fi connection is also tested for reliability, and in regions with weak signals, alternative communication methods such as LoRaWAN-a long-range, low-power wireless protocol-are explored to guarantee continuous data transmission.

3.1 System Design

The device is composed of a NodeMCU, HC-SR04 ultrasonic sensor, water flow sensor, rain sensor, water level sensor, DHT11 temperature and humidity sensor, a 0.91-inch OLED display, a buzzer, and an LED. These sensors measure various environmental conditions such as temperature, humidity, rainfall, water flow, and water level. The data from these sensors is transmitted to the NodeMCU, which processes the information to assess the potential risk of flooding. The OLED display shows the sensor readings, while the buzzer and LED provide alerts when a flood risk is detected. Additionally, all the sensor data can be monitored through a web interface.

a) The ESP32-WROOM microcontroller

The ESP32-WROOM microcontroller acts as the core of the flood detection system, handling sensor data collection, data processing, and alert transmission when needed. Renowned for its suitability in IoT projects, the ESP32 offers low power consumption, strong processing capabilities, and built-in Wi-Fi and Bluetooth support. Powered by a dual-core Xtensa LX6 processor, it efficiently manages multiple tasks at once, making it well-suited for real-time monitoring. It also supports various communication interfaces like SPI, I2C, and UART, ensuring smooth integration with a wide range of sensors. A standout feature of the ESP32 is its deep sleep mode, which significantly reduces power usage when continuous sensor readings aren't necessary-an essential benefit for systems operating in remote or energy-constrained locations.

b) DHT11 Sensor

During The DHT11 sensor plays a crucial role in tracking environmental temperature and humidity levels. These factors are significant indicators, as sudden changes-especially increased humidity-can signal the likelihood of heavy rainfall, which may lead to flooding. This sensor outputs data in digital form, making it straightforward to interface with the ESP32 microcontroller. It operates within a voltage range of 3.3V to 5V, ensuring smooth integration with the system's power supply. With a temperature accuracy of ±2°C and a humidity accuracy of ±5%, the DHT11 offers sufficient precision for monitoring weather conditions linked to flood risks.

By consistently recording humidity and temperature trends, the sensor helps detect atmospheric changes that typically occur before storms, enabling the system to issue early flood warnings.

c) HC-SR04 Ultrasonic Sensor

The HC-SR04 ultrasonic sensor is essential for tracking water levels in the system. It operates using the time-of-flight method, where ultrasonic pulses are emitted and reflected off the water surface. By measuring the time, it takes for the echo to return, the system accurately calculates the distance from the sensor to the water surface. This makes the sensor ideal for monitoring water levels in areas like rivers and reservoirs, with a reliable measurement range of up to 4 meters. It runs on a 5V supply, making it directly compatible with the ESP32 microcontroller. A major benefit of the HC-SR04 is its ability to measure water levels without making physical contact, which reduces the risk of sensor damage and ensures long-term performance even in challenging environmental conditions.

d) Raindrop Sensor

The raindrop sensor is used to monitor the presence and intensity of rainfall. It features a conductive surface that alters its resistance when water droplets come into contact with it. Depending on the required sensitivity, the sensor can output either analog or digital signals.

This component is particularly effective in detecting sudden and heavy rainfall, which is crucial for anticipating flash flood situations. Operating within a voltage range of 3.3V to 5V, it delivers real-time rainfall data, enabling the system to evaluate weather conditions independently of external weather services.

Because the sensor is directly exposed to the environment, careful placement is important. It should be mounted in an open yet protected location to minimize false readings caused by factors like condensation or debris accumulation.

e) Water Level Sensor

The water level sensor offers a direct analog output to measure rising water levels, providing a practical alternative to the distance-based measurements of the HC-SR04 ultrasonic sensor. It classifies flood risk into three stages-low (normal level), medium (rising water, early warning), and high (critical level, immediate action) -allowing for timely preventive responses. Due to its compact size, affordability, and ease of installation, it is especially suitable for use in areas vulnerable to flooding

f) YFS201 Flow Sensor

The YFS201 flow sensor is designed to measure the rate of water flow in rivers, pipelines, or drainage systems. It functions by detecting the pulse signals produced by a small internal turbine; the frequency of these pulses corresponds directly to the flow rate of the water.
Monitoring real-time flow data enables the system to identify abrupt surges in water movement, which could signal events such as dam breaches, intense rainfall, or blocked drainage systems. Its energy-efficient design makes the YFS201 well-suited for ongoing, long-term monitoring in flood detection systems.

g) OLED Display

The OLED display acts as an on-site interface, providing immediate visualization of real-time data from the connected sensors. It allows users to quickly view key environmental parameters such as water levels, temperature, humidity, and rainfall intensity directly from the device. With a resolution of 128x64 pixels, the display ensures clear and readable output, even under bright outdoor lighting. Using the I2C communication protocol, it requires minimal wiring and operates with low power consumption, making it a practical and energy-efficient component for IoT-based flood monitoring solutions.

3.2 Implementation and Hardware Setup

The implementation of the proposed system involves the integration of machine learning-based face recognition with IoT-enabled hardware to create an automated, intelligent car parking solution. The system architecture consists of three main components: the facial recognition module, the

Paper ID: SR25417203249 DOI: https://dx.doi.org/10.21275/SR25417203249

hardware control unit, and the web-based registration and monitoring interface. Each component operates in coordination to ensure seamless functionality.

The facial recognition module is developed using Python and OpenCV. It runs on a computer or edge device such as a Raspberry Pi. The module captures images from a connected camera (e. g., USB webcam or ESP32-CAM), processes them to detect facial features, and performs real-time recognition using the LBPH model trained on registered user images. Upon successful recognition, a command is generated to control the hardware.

The hardware setup comprises an ESP32 microcontroller, a servo motor, and optional ultrasonic sensors for parking slot detection. The ESP32 is programmed using the Arduino IDE and is responsible for controlling the servo motor, which physically opens and closes the parking gate. The servo receives input from the face recognition system via serial communication (USB/COM port) or via Wi-Fi using HTTP requests, depending on the chosen architecture.

For slot availability management, ultrasonic sensors can be placed in each parking slot to detect the presence or absence of a vehicle. The sensor data is read by the ESP32 and can be transmitted to the database or web dashboard in real-time, enabling dynamic monitoring of available slots.

The web application is built using HTML, CSS, JavaScript, PHP, and MySQL. It provides a user-friendly interface for new users to register their details and upload facial images. The admin panel allows authorized personnel to monitor parking logs, view slot availability, and manage user access. The web app and the face recognition module share access to the image dataset folder or a database, ensuring a unified data pipeline for training and recognition.

Power supply to the system components is provided on the servo and sensor requirements. All wiring connections between ESP32, servo motor, and sensors are handled through a breadboard or custom PCB, with GPIO pins configured for input/output as required.

This modular and scalable implementation ensures that the system can be deployed in various environments - from residential buildings to office campuses - with minimal modification. Its ability to combine machine learning with real-world physical control via microcontrollers forms the backbone of the proposed intelligent parking system

4. Results and Evaluation

This section provides an evaluation and performance analysis of the IoT-based flood detection system, which was tested in controlled settings to assess its functionality, sensor accuracy, communication efficiency, and alert system effectiveness. The results confirm the system's capability to perform reliably in flood-prone areas.

The integrated system, which includes sensors, a microcontroller, alert mechanisms, and cloud services, functioned smoothly without any issues. Real-time monitoring and data collection were successfully executed, with the ESP32 microcontroller efficiently processing data from the sensors and transmitting it to the ThingSpeak platform for storage, analysis, and visualization.

Each sensor was tested individually for accuracy, responsiveness, and stability. The DHT11 sensor effectively measured temperature and humidity, providing reliable readings in varying environmental conditions, making it useful for predicting potential flood risks. The HC-SR04 ultrasonic sensor accurately gauged the water surface distance, which is essential for water level monitoring. Its non-contact nature ensured it remained durable in wet environments. The raindrop sensor reliably detected rainfall onset and intensity, triggering timely alerts for heavy rain. The water level sensor provided direct analog readings of water depth, categorizing flood risks into normal, warning, and critical stages. It performed consistently well in detecting changes in water levels. The YFS201 flow sensor accurately monitored water flow, aiding in the detection of surges or overflow conditions, crucial for flood prediction.

The ESP32 microcontroller's Wi-Fi feature enabled seamless transmission of data to the ThingSpeak platform, with no significant delays or interruptions. The cloud platform displayed real-time updates, and historical data was logged for further analysis, ensuring efficient communication with no data loss.

The alert system, including the buzzer, LED, and cloud-based notifications, functioned as expected. When water levels exceeded preset thresholds, alerts were triggered and displayed in real-time on the web dashboard. The system accurately differentiated between normal, warning, and critical conditions, triggering the appropriate alerts. No alerts were issued for normal levels, a yellow alert was triggered for warning levels, and a red alert, along with buzzer and LED activation, was issued for critical levels. These alerts allowed timely actions based on flood risk levels.

The system was designed for low power consumption, with the ESP32's deep sleep mode effectively conserving energy. This feature made it suitable for remote installations powered by solar panels or batteries.

Throughout the testing process, the system demonstrated stable performance, with sensors, communication, and alert systems working seamlessly together. The combination of local feedback (OLED display, buzzer, LED) and cloud-based remote monitoring ensured a comprehensive flood detection and alerting solution.

5. Conclusion

This paper presented the flood detection system developed in this study provides a dependable, affordable, and real-time approach for monitoring water levels and identifying potential flood risks in at-risk regions. By combining a range of IoT sensors-including water level sensors, ultrasonic sensors, and raindrop sensors-with the powerful ESP32 microcontroller, the system is capable of gathering critical environmental data and transmitting it seamlessly to the ThingSpeak cloud platform. This integration facilitates the real-time analysis and visualization of data, allowing for

Paper ID: SR25417203249 DOI: https://dx.doi.org/10.21275/SR25417203249

prompt detection of rising water levels and extreme weather conditions that could lead to floods.

The use of IoT technology in this system enhances its ability to provide continuous, remote monitoring, offering an efficient way to track environmental changes in real-time. The system's cloud-based architecture ensures that data is stored securely, and visual dashboards allow for easy access and analysis, making it a practical solution for authorities and local governments to monitor flood-prone areas effectively.

Additionally, the system's alert mechanism, which includes notifications and real-time updates, helps to trigger timely responses when water levels reach predefined thresholds, enabling rapid intervention. The inclusion of low-power components ensures that the system can operate in remote or off-grid locations where access to electricity may be limited, making it an ideal solution for diverse geographic settings.

Overall, this flood detection system demonstrates great potential for contributing to flood management efforts, providing a reliable and scalable solution to minimize the risks associated with flooding and improve response times in vulnerable communities. With future improvements in sensor accuracy and communication technology, the system could be expanded for even more widespread applications in flood monitoring and early warning systems globally.

References

[1] Khan, M. S., & Shah, S. R. (2018). *IoT-based Flood Detection and Monitoring System*. International Journal of Computer Applications, 178 (11), 1–6. https: //doi. org/10.5120/ijca2018917401
[2] Patel, H., Patel, D., & Kumar, A. (2019). *Design and Implementation of an IoT-based Flood Monitoring System using Water Level Sensors and GSM*. Proceedings of the 2nd International Conference on Emerging Trends in Computer Science and Engineering, 50–55. https: //doi. org/10.1109/ICETCS.2019.8762251
[3] Zhang, Y., Chen, X., & Wang, X. (2020). *AI-based Flood Prediction System using IoT Sensors and Machine Learning Algorithms*. Journal of Environmental Management, 260, 110123. https: //doi. org/10.1016/j. jenvman.2019.110123
[4] Singh, P., Gupta, R., & Verma, S. (2021). *Neural Network-based Flood Risk Prediction System using IoT and Historical Data*. International Journal of Advanced Computer Science and Applications, 12 (7), 50–58. https: //doi. org/10.14569/IJACSA.2021.0120731
[5] Ghosh, S., & Bhattacharya, A. (2020). *Role of ESP32 in IoT-based Flood Detection Systems: A Case Study*. IEEE Access, 8, 145678–145686. https: //doi. org/10.1109/ACCESS.2020.3013447
[6] Kumar, M., & Kumar, V. (2018). *Long-range Communication using LoRaWAN for IoT-based Flood Detection*. International Journal of Communication Systems, 31 (9), e3693. https: //doi. org/10.1002/dac.3693
[7] Liu, H., Zhang, X., & Li, Y. (2021). *Integrated Flood Prediction System with IoT and Data Visualization: A Case Study*. Journal of Hydrology, 601, 126546. https: //doi. org/10.1016/j. jhydrol.2021.126546
[8] Li, J., Zhang, Q., & Feng, Z. (2019). *Challenges and Solutions in IoT-based Flood Detection and Early Warning Systems*. Environmental Monitoring and Assessment, 191 (6), 400. https: //doi. org/10.1007/s10661-019-7380-5
[9] Smith, J. (2017). *Traditional Methods of Flood Detection and Their Limitations*. Natural Hazards Review, 18 (3), 03017002. https: //doi. org/10.1061/ (ASCE) NH.1527-6996.0000267
[10] ThingSpeak. (n. d.). *ThingSpeak: The Open IoT Platform*. Retrieved from https: //thingspeak. com/
[11] Espressif Systems. (2020). *ESP32 Technical Reference Manual*. Retrieved from https: //www.espressif. com/en/products/socs/esp32
[12] Agarwal, S., & Gupta, P. (2019). *Development of an IoT-based Flood Monitoring and Warning System*. International Journal of Electronics and Communication Engineering, 13 (2), 73–80.
[13] Rao, A. S., & Reddy, V. R. (2020). *IoT-based Smart Flood Monitoring System using Cloud and Wireless Sensor Networks*. International Journal of Engineering Research and Technology (IJERT), 9 (8), 450–454. https: //doi. org/10.17577/IJERTV9IS080261
[14] Islam, M. N., Hossain, S., & Karim, R. (2020). *Design of an IoT-based Real-Time Flood Monitoring and Alert System*. International Journal of Scientific & Engineering Research (IJSER), 11 (4), 1230–1235.
[15] Sharma, R., & Kaur, K. (2021). *IoT and Cloud-Based Flood Monitoring System Using ESP32*. International Journal of Computer Trends and Technology (IJCTT), 69 (6), 39–44. https: //doi. org/10.14445/22312803/IJCTT-V69I6P107