Design and Implementation of an IoT-Enabled Wireless Sensor Network for Real-Time Flood Detection and Alerting

Siddhi Patil[1], M. S. Patil[2], Shradha Wadkar[3], Swapnil Patil[4], Abhishek Potdar[5]

[1,3,4,5]Undergraduate students, Department of Electronics and Telecommunication Engineering, Kasegaon Education Society's Rajarambapu Institute of Technology, Sakharale, Maharashtra

[2]Head of Department,Department of Electronics and Telecommunication Engineering, Kasegaon Education Society's Rajarambapu Institute of Technology, Sakharale, Maharashtra

***Abstract*—Floods pose a serious and increasing threat. Immediate, real-time information is the most valuable tool for protecting communities. To address this need, our project outlines the creation of a real-time, IoT-based flood detection system. The proposed model employs a dual-sensor approach to gain a complete and accurate understanding of flood risk. We use an ultrasonic sensor (HC-SR04) to constantly measure the water level, tracking its rise and fall. At the same time, a water flow sensor (YF-S201) measures how fast the water moves a key detail that level-only sensors overlook. This data goes to an STM32 microcontroller, which acts as the system's "brain," processing the information. A dedicated Wi-Fi module (ESP8266) then transmits the data wirelessly to a cloud dashboard. This enables users and local authorities to see real-time water levels and flow rates displayed in a clear graphical format. By using a low-cost, scalable, and dependable two-chip design (STM32 for processing, ESP8266 for communication), our method provides an efficient and accessible flood monitoring solution, ideally suited for the rural and semi-urban areas that need it most.**

***Index Terms*—Sensor Networks, IOT, Dashboard, real time monitoring, Wi-Fi**

I. INTRODUCTION

The impact of floods leads to significant loss of life and severe economic damage worldwide. This ongoing crisis disrupts infrastructure and displaces communities. It requires a shift from slow, reactive responses [1,2]. Traditional flood management systems often rely on manual observations or isolated gauges, and they are frequently overwhelmed. These systems do not provide the timely, detailed data necessary for effective early warnings. This situation highlights the urgent need for real-time monitoring solutions.

The Internet of Things (IoT) has emerged as a promising and cost-effective technology to address this issue [5]. IoT allows the collection, processing, and transmission of live environmental data. This capability facilitates the creation of smart sensor networks. Researchers have explored various designs for these monitoring systems. Many studies show that low-cost microcontrollers, such as the NodeMCU (which uses the ESP8266 chip), can effectively connect sensors to the internet [1,6]. Other approaches utilize more powerful but often complex single-board computers or specialized systems to handle data processing at the node level [2,5]. While these existing systems demonstrate the feasibility of IoT, the main engineering challenge remains: creating a solution that is reliable, scalable, low-cost, and capable of efficiently transmitting high-volume hydrological data.

A crucial part of any effective warning system is its user interface and alert mechanism. Some systems send critical alerts through methods like GSM-based SMS messages [6]. While this method works for one-way alerts, it limits the user by not providing a continuous stream of informative data. A more robust approach is to offer users a live, data-rich dashboard, where cloud-based platforms like Blynk become essential. These platforms provide an easy way to visualize time-series sensor data, such as water levels and flow rates, directly on a mobile application [1]. They furnish users with immediate and actionable situational awareness.

Additionally, research now focuses on integrating complex Machine Learning (ML) or Deep Learning (DL) models for predictive forecasting [3]. However, these sophisticated algorithms rely heavily on the quality and consistency of the data they receive [5]. A solid and reliable data acquisition pipeline is essential for any smart system and is necessary for future predictive analytics. Without consistently collecting, processing, and transmitting clean data to the cloud, even the most advanced predictive models become ineffective.

This project aims to create a highly reliable, real-time flood monitoring system with a strong data pipeline at its core. It employs a dual-microcontroller setup to improve performance and stability. The powerful STM32 microcontroller handles sensor interfacing and data processing. Meanwhile, the efficient ESP8266 Wi-Fi module acts as a dedicated communication gateway. The primary goal is to stream sensor data, such as water levels and flow, directly to the Blynk application. This clear division of computational and communication tasks ensures system stability and provides an accessible, low-cost solution for immediate, high-quality, real-time flood-level awareness.

II. METHODOLOGY

A. Sensor Layer (Input): The real-time flood monitoring system is built on the Sensor Layer. This layer is responsible for continuously and accurately collecting data from the physical monitoring site. It acts as the key link between the hydrological environment and the digital processing stages that follow. The system uses specific sensors designed for reliability and precision in outdoor conditions: the Ultrasonic Sensor and the Water Flow Sensor (Fig.1)

1. Water Level Monitoring (Ultrasonic Sensor): The system uses an Ultrasonic Distance Sensor (e.g., HC-SR04) to provide ongoing measurements of the water level. This non-contact method is essential for reliable performance in variable conditions. The sensor works by emitting a high-frequency ultrasound pulse and measuring the time it takes for the echo to return (Time-of-Flight principle). The distance to the water surface (D) is calculated using this formula, where t is the measured time and c is the speed of sound: $D = t \times c / 2$ This sensor is placed at a known height (H) above the riverbed or flow path. The actual water depth (W) is calculated by subtracting the measured distance from the total known height, yielding a precise, real-time value: $W= H - D$. The constant stream of water depth data is the main input for flood-level alarms.

2. Water Flow Velocity (Water Flow Sensor): To give a clear understanding of the hydrological conditions, a Water Flow Sensor (e.g., a Hall effect turbine meter) is added to the flow path. This sensor measures the water's velocity by tracking the rotation of an internal turbine or impeller. As water flows, the impeller spins, and a magnetic field activates a Hall effect sensor to generate a series of electrical pulses. The count of these pulses over a specific time period directly relates to the flow rate of the water (usually measured in Liters per minute, or L/min). This raw pulse-count data is sent to the processing unit, where it is converted into a measurable velocity metric. This flow data is important for short-term predictions and assessing the risk of flooding. The raw data stream, which consists of the digital level data from the Ultrasonic Sensor and the frequency data from the Water Flow Sensor, is transmitted directly to the processing unit (STM32 microcontroller) for analysis, aggregation, and threshold comparison. This triggers the next stages of the flood detection process.

B. Control & Processing Unit: The STM32 Microcontroller acts as the brain of the system. It handles all the essential edge computing tasks needed for real-time flood monitoring and local control. This setup allows the powerful STM32 to concentrate on computational and control tasks, which ensures reliability and quick responses (Fig.1)

1. Data Processing and Conversion: The STM32 starts by managing the incoming data stream from the Sensor Layer. It performs real-time data conversion and filtering to change raw electrical signals into usable environmental metrics:

- Water Level Calculation: The STM32 gets Time-of-Flight values from the Ultrasonic Sensor. It calculates the water level in centimeters by factoring in the speed of sound and the known mounting height. This includes using digital filters, like median or averaging filters, to reduce noise and ensure the reading is accurate.

- Flow Rate Calculation: The microcontroller continuously reads the pulse count from the Water Flow Sensor. It calculates the frequency of these pulses over set intervals and converts this into a Flow Rate in liters per minute using the sensor's calibration factor (K-factor).

2. Execution of Decision Logic: After calculating the environmental metrics, the STM32 runs the Decision Logic. This means comparing the Water Level and Flow Rate against established safety thresholds specific to the site. The system assigns one of three operational statuses:

- Normal: Water levels and flow rates are within typical ranges, requiring only routine data logging.
- Alert (Warning): Levels or flow rates have exceeded the initial threshold, which signals a possible, non-critical threat. This starts a notification sequence through the communication layer.
- Danger (Critical): Levels or flow rates have gone beyond the highest safety threshold, indicating a possible flood event. This state triggers immediate local control action.

3. Dedicated Communication Formatting: The STM32 also prepares data for cloud transmission. It formats the status along with the calculated Water Level and Flow Rate into a compact digital packet. This information is sent to the ESP8266 Wi-Fi Module through a wired Serial (UART) link. By keeping the data processing separate from the Wi-Fi connection, this two-microcontroller approach improves the system's stability and speed.

4. Immediate Local Control: If the Decision Logic indicates Danger, the STM32 focuses on immediate local control to reduce potential damage. It sends a high signal to the Relay Module, which functions as an electronic switch. This relay quickly powers the Solenoid Valve, causing it to close and stop the main water flow. This direct hardware action avoids network delays, creating a vital failsafe response at the system's edge.

C. Alert & Monitoring Layer (Output): This final layer of the system focuses on establishing reliable connectivity, providing user feedback, and executing the control response based on the STM32's analysis. This layer includes the communication module and the output mechanisms (Fig.1)

1. Communication Gateway (ESP8266): The ESP8266 Wi-Fi Module acts solely as a communication gateway. It receives the formatted status data from the STM32 through the Serial (UART) link. The ESP8266's only job is to connect to the Internet and securely send the processed metrics to the Blynk Cloud Platform.

Decoupled Operation: This strategy keeps the STM32 free from the complexities and timing demands of the TCP/IP stack and Wi-Fi management. It maximizes the stability of the critical data processing functions.

Secure Transmission: The module sends data packets securely to the project dashboard on the cloud server. It serves as a bridge between the local monitoring node and the global network.

2. Cloud Platform and User Interface (Blynk): The Blynk Cloud Platform acts as the main hub for data management and user interaction, offering two important outputs:

Real-time Dashboard: The cloud server quickly updates the Real-time Dashboard, accessible through a mobile app or web browser. This allows users to monitor the Water Level and Flow Rate remotely. Users can also see the color-coded status (Normal, Alert, Danger) based on the STM32's logic. This live visualization is key for improving situational awareness.

Automated Early Warning System: The Cloud manages the proactive early warning system. If the metrics exceed an 'Alert' or 'Danger' threshold, the system automatically triggers and sends Mobile/Web Notifications (such as push alerts and emails) to registered users and authorities. This key automated function ensures warnings reach a wide audience quickly, meeting the main goal of an early flood detection system.

3. Automatic Control Response (Solenoid Valve): The final output element is the Automatic Control Response, which provides immediate physical action at the monitoring site.

Local Command Execution: The Solenoid Valve, connected to and controlled by the STM32 through the

Relay Module, represents the system's automated physical response.

Hazard Mitigation: When the system receives a 'Danger' command from the processing unit, the Solenoid Valve quickly shuts, helping to stop the hazard by cutting off the water flow. This mechanism ensures the system offers not just warnings, but also a direct, local control measure to prevent further escalation of a critical event.

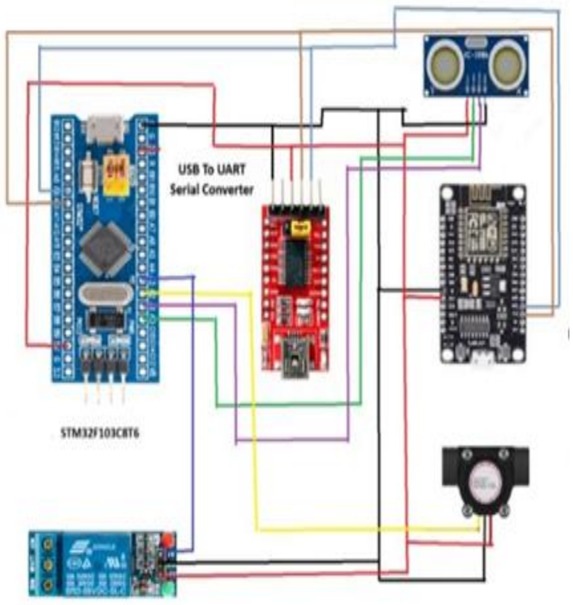


Figure 1: Circuit diagram of the system.

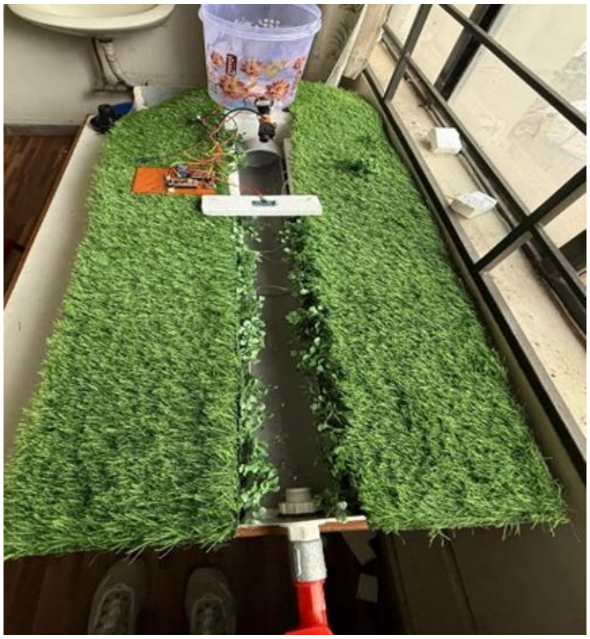


Figure 2: Prototype of the system.

III. RESULTS AND DISCUSSION

The physical prototype created to test the real-time flood monitoring system in a controlled setting. (Fig.2). The prototype represents a simulated water channel, featuring a central trough with artificial turf. Here, water levels and flow rates can be controlled precisely.

Key Components & Validation Focus:

1.Water Source: A reservoir (plastic container) at the top acts like rainfall or a rising river source.

2.Sensor Layer: The Ultrasonic Sensor continuously measures the water depth. The Water Flow Sensor is placed in the flow path to measure velocity.

3.Control Unit: The main electronic parts (STM32/ESP8266) process the sensor data to check the Decision Logic, which includes Normal, Alert, and Danger statuses.

4.Validation: The main goal is to confirm:

- The accuracy of the sensor readings.
- The timing and correctness of the automated 'Alert' and 'Danger' triggers.
- The functionality of the Communication Layer for sending real-time data to the cloud.

For cloud integration here we are using Blynk which is a low-code IoT (Internet of Things) platform that connects hardware (the STM32-ESP8266 node) to the end-user. It allows for secure data exchange and configuration management between the physical hardware and a user interface available through a mobile application on iOS or Android devices. The main goal of the platform is to speed up project development by allowing users to create a professional-looking graphical user interface (GUI) with little effort. Instead of needing complex and time-consuming app code, users can quickly build the application using a visual interface. They can drag and drop pre-built components onto a digital canvas. This method enables a fast deployment of the monitoring interface.

Key Operational Components: The platform works through a central Blynk Server (Cloud), which acts as a secure middleman, managing the data flow and user authentication between the hardware and the client application. In the context of the flood monitoring system, the Blynk App (Mobile) provides the main visualization and interaction interface using these specialized widgets:

Value Display Widget: This widget shows the real-time numeric output of primary metrics, like the water level in centimeters (e.g., "6.97 cm") from the ultrasonic sensor.

Gauge Widget: This widget offers an analog view of the water flow metric (e.g., "26.67 L/min"). The gauge format allows for a quick and easy assessment of the current reading in relation to maximum safe limits.

Button Widget: This widget enables a remote human-machine interface (HMI) for manual control. It sends a command back through the cloud to the hardware, allowing users to remotely turn the control mechanism on or off, like the Relay linked to the Solenoid Valve.

Notification Widget: This widget receives and displays automated system alerts right away. It includes important messages, such as "Alert! Water level is Increased!" ensuring users receive timely warnings on their mobile devices when the Decision Logic exceeds set safety thresholds.

Applications in This Project:
The Blynk platform in this "STM32 Controller" project is essential to the Communication and Output Layer, performing three distinct and vital functions:

1.Remote Monitoring: It acts as the user interface, displaying real-time sensor data (water height and water flow) received wirelessly from the STM32-ESP8266 node directly on a user's mobile phone via the Blynk App.

2.Remote Control: It provides a two-way communication channel. The user can execute remote actions, such as turning the physical "Relay" on or off using a designated "ON" button widget within the application.

3.Instant Alerting: It manages the critical early warning system by immediately sending a push notification to the user's phone as soon as the water level (measured by the ultrasonic sensor) is processed by the control unit and goes over the 10 cm safety limit.

Test Result:
Test Case 1: Normal Operation (Fig.3)
This snapshot shows the "STM32 Controller" app during normal monitoring
Water Height: 6.97 cm (below the 10 cm limit)
Water Flow: 26.67 (active flow detected)
Relay: ON.
Status: The system is monitoring normal water levels and flow with the relay engaged.

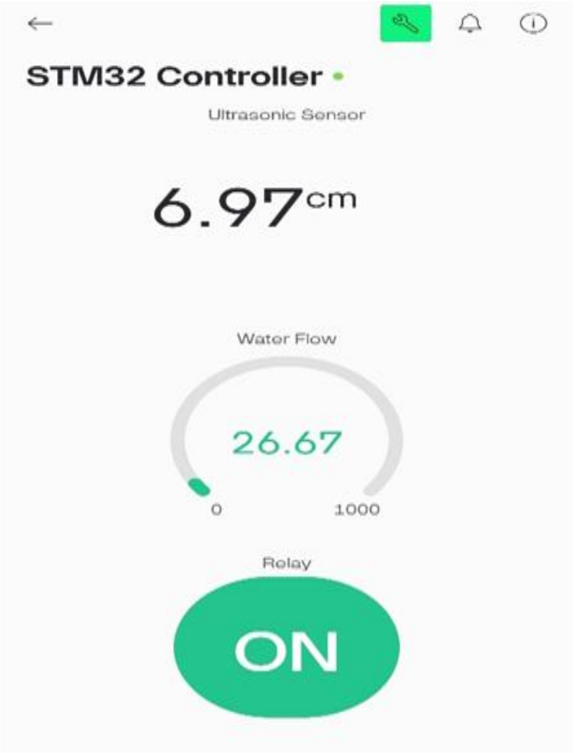
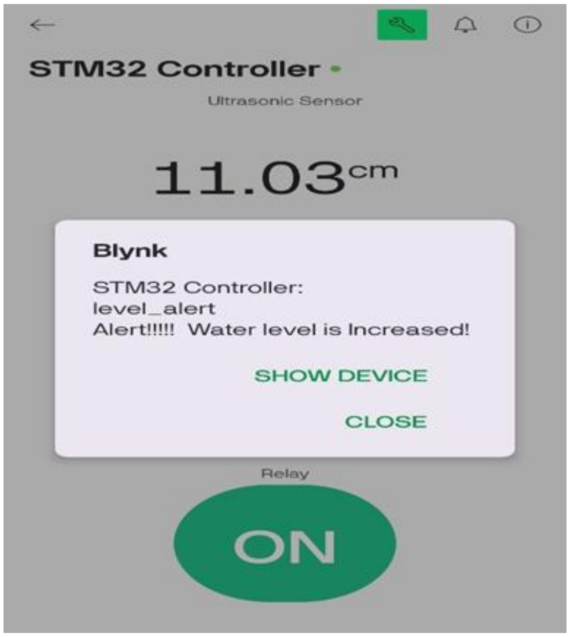


Figure 3: Result 1 on Blynk App

Test Case 2: Alert Condition (Fig.4) This snapshot shows how the system responds to a critical water level.
Water Height: 11.03 cm (Exceeds 10 cm safety limit).
Blynk Alert: Triggered. The notification "Alert!!!!! Water level is Increased!" was sent successfully.
Relay: ON (this is shown at the moment of the alert but is set to turn OFF to deactivate the solenoid valve).
Status: This test confirms that the safety feature works. It shows that high water levels trigger the alert and the relay response as intended.



Blynk

STM32 Controller:
level_alert
Alert!!!!! Water level is Increased!

Figure 4: Result 2 on Blynk App

This test confirms the system's main functionality. It shows that it can automatically detect anomalies based on set thresholds and notify users right away through the Blynk platform. The successful completion of this test indicates that the control logic programmed on the STM32 works well and responds quickly to changes in real-time data.

Additionally, it confirms the smooth integration and dependable data transfer between the sensor layer, the STM32 processor, and the ESP8266 communication module. This setup ensures that any critical status detected at the node is quickly displayed on the user's dashboard.

Together, these two test cases show that the STM32-based controller fulfills its main design goals: reliable real-time environmental monitoring and immediate automated alerts when critical user-defined thresholds are met.

IV.CONCLUSION

The rising global number of flash floods, worsened by unpredictable climate, shows the urgent need for real-time, proactive flood management systems that go beyond simple water-level checks. This work presents the design, integration, and thorough testing of a new IoT-based Flood Detection and Mitigation Prototype.

The main focus of this research is the development of a Data Fusion engine that runs on the powerful STM32 microcontroller. By taking inputs from the Ultrasonic sensor (water level) and the Water Flow sensor (velocity) at the same time, the system overcomes the limits of monitoring just one parameter. This combined data stream goes to a specific classification model that accurately calculates a Flood Risk Index (FRI). This improves the system from only reporting an event to predicting how severe its impact will be. Importantly, the prototype can take automated, decisive physical action. When the FRI surpasses a critical level, the ESP8266 module sends urgent alerts through the Blynk cloud, while the STM32 simultaneously triggers the Relay to shut the Solenoid Valve. This offers a way for immediate, localized flow control, which is crucial for protecting downstream infrastructure and resources. The successful testing of the internal software and component integration confirms the reliability and quick response of the design.

In conclusion, the prototype provides a cost-effective, strong, and smart solution for early flood warnings and automated local responses. It marks an important advance in embedded systems engineering for disaster management.

REFERENCE

[1] (Journal) C.K. Gomathy, G.G.Lasya Priya, Hemanth Kumar K.N.A Study on IoT-Based Flood Detection Management System April 2021International Journal of Engineering and Advanced Technology 10(4):130-133Available at: https://www.researchgate.net/publication/351295316_A_Study_on_Iot_Based_Flood_Detection_Management_System

[2] (Journal) Emil Daniel Maer, Adrian Augustin Pop. Design of IoT based flood monitoring and alerting system. 2022 IEEE 20th International Power Electronics and Motion Control Conference (PEMC), Available at: https://ieeexplore.ieee.org/abstract/document/9962925

[3] (Journal) Chandra Prakash, Anurag Barthwal, Debopam Acharya. FLOODWALL: A Real-Time Flash Flood Monitoring and Forecasting System Using IoT. In IEEE Sensors Journal (Volume: 23, Issue: 1, 01 January 2023) Available at: https://ieeexplore.ieee.org/abstract/document/9966518

[4] (Journal) David C. Mason; Ian J. Davenport; Jeffrey C. Neal; Guy J.-P. Schumann; Paul D. Bates 2012 Near Real-Time Flood Detection in Urban and Rural Areas Using High-Resolution Synthetic Aperture Radar Images. In IEEE Transactions on Geoscience and Remote Sensing (Volume: 50, Issue: 8, August 2012) Available at: https://ieeexplore.ieee.org/abstract/document/6155739

[5] (Journal) Azimah Abdul Ghapar. Internet of Things (IoT) Architecture for Flood Data Management. In International Journal of Future Generation Communication and Networking January 2018 ,11(1):55-62, DOI:10.14257/ijfgcn.2018.11.1.06, Available at: https://www.researchgate.net/publication/323124296_Internet_of_Things_IoT_Architecture_for_Flood_Data_Management

[6] (Journal) L. Saravanan, W. Nancy; K. Prabhu Chandran,D. Vijayanandh, J.R. Arunkumar,R. Thandaiah Prabhu,A Novel Approach for a Smart Early Flood Detection and Awareness System using IoT. In 2022 8th International Conference on Smart Structures and Systems (ICSSS), Available at: https://ieeexplore.ieee.org/abstract/document/9782286