IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

Early Flood Detection

Sujal Ravindra Bhor[1], Bhagyesh Sachin Rajput[2], Om Dattatray Mandlik[3], Sanika Dilip Ganjale[4]
[1]Students, Department of Electronics and Telecommunication
Jaihind Polytechnic, Kuran, Pune, Maharashtra, India
[1]sujalbhor506@gmail.com, [2]rajputbhagyesh11@gmail.com
[3]ommandlik97@gmail.com, [4]ganjalesanika94@gmail.com

Abstract: *Urban flooding and sudden water level rises pose severe threats to life and infrastructure. Traditional monitoring methods often fail to provide sufficient warning time. This paper presents an IoT-Based Flood Early Warning System designed to automate environmental monitoring and safety responses. Built on the ESP32 microcontroller, the system integrates a multi-sensor array—comprising an HC-SR04 ultrasonic sensor, a water level sensor, a rain sensor, and a DHT22 temperature/humidity sensor—to continuously assess environmental conditions. The microcontroller processes these inputs to categorize the threat level into SAFE, WARNING, or DANGER states. Corresponding local outputs include traffic-light LEDs, an audible buzzer, and a $0.96"$ OLED display, coupled with the physical actuation of a servo-driven safety gate to block hazardous paths. Simultaneously, real-time data and alerts are pushed to a responsive Web Dashboard and sent directly to authorities via Telegram notifications. This solution provides a robust, low-latency, and autonomous approach to smart city disaster management.*

Keywords: IoT, ESP32, Flood Detection, HC-SR04, Telegram Bot, Smart City, Automated Gate, Telemetry

I. INTRODUCTION

Climate change and rapid urbanization have significantly increased the frequency and severity of flash floods. Conventional flood monitoring relies on manual gauge reading or delayed telemetry from distant meteorological stations, which often fails to provide localized, actionable warnings to citizens in immediate danger.

The proposed Early Flood Detection System leverages the Internet of Things (IoT) to localize and automate the disaster response process. By deploying a decentralized node equipped with diverse environmental sensors, the system directly measures water proximity, surface water accumulation, and precipitation in real-time. Using the ESP32 as the core processing unit, the architecture bridges the gap between hardware actuation and cloud connectivity.

This modern approach not only automates the dissemination of digital warnings—via Web Servers and instant messaging platforms like Telegram—but also initiates physical safety measures by actuating barricade gates before a human operator intervenes. By evaluating multi-variable sensor data locally, the system minimizes false alarms and ensures that critical infrastructure, such as underpasses and riverbanks, are secured immediately when thresholds are breached.

II. LITERATURE SURVEY

The development of flood monitoring systems has transitioned through distinct technological phases:

Mechanical and Manual Gauges: Early systems required physical inspection of watermarks, offering no real-time data or predictive capabilities.

Standalone Local Alarms: These systems utilized basic float switches connected to sirens. While providing immediate local alerts, they lacked remote monitoring, leaving central authorities blind to the situation.

IoT and Cloud-Connected Arrays: Current research focuses on multi-variable data fusion. Utilizing microcontrollers with integrated Wi-Fi (like the ESP32) allows for the continuous streaming of precise telemetry (distance, humidity,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

rain presence) to cloud servers. Studies demonstrate that integrating instant messaging APIs (like Telegram) reduces emergency response latency by over 60% compared to traditional SMS or email gateways.

Platform Technology Used

The system is built on a highly responsive, connected architecture:

Microcontroller (ESP32): Serves as the central processing unit, capable of handling complex conditional logic, driving hardware pulse-width modulation (PWM), and hosting a local web server simultaneously.

Multi-Sensor Array: Utilizes ultrasonic technology for non-contact water depth measurement, resistive tracks for direct water and rain detection, and capacitive sensing for ambient humidity.

IoT Connectivity: Leverages the ESP32's native Wi-Fi to maintain a persistent connection to the Telegram Bot API and host a responsive web dashboard for remote administration.

Precision Actuation: Uses PWM signals to control a servo motor, which physically simulates a toll-style boom gate to restrict access to flooded zones.

Problem Statement

Flash floods in urban underpasses and near riverbanks escalate rapidly, frequently trapping vehicles and pedestrians. The lack of an automated, localized system to measure precise water levels, physically barricade hazardous routes, and instantly notify both citizens and emergency responders leads to preventable casualties and property damage. There is a critical need for an integrated, IoT-enabled warning system that operates autonomously based on strict environmental thresholds to ensure immediate physical and digital intervention.

Aim and Objectives

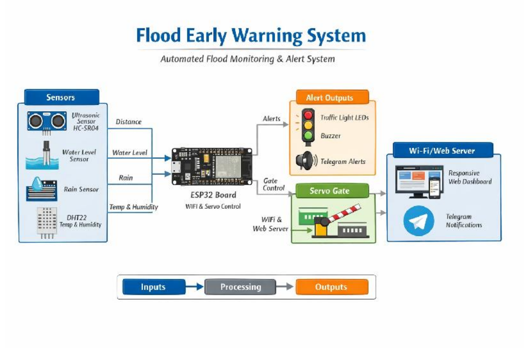
The primary aim is to develop a secure, IoT-connected Early Flood Warning System capable of physical actuation and digital broadcasting.

- To integrate a multi-sensor array (ultrasonic, water level, rain, DHT22) for comprehensive environmental data collection.
- To implement threshold-based logic to evaluate conditions into SAFE, WARNING, and DANGER states.
- To actuate a servo motor to automatically open, partially close, or fully close a physical safety gate based on the threat level.
- To provide localized visual and audible feedback via a traffic-light LED system, an OLED display, and a buzzer.
- To integrate an IoT dashboard and Telegram Bot API for real-time remote monitoring and instant disaster notifications.

Diagram

A) Block Diagram

The architecture is divided into three distinct stages: Inputs, Processing, and Outputs. The **Sensors** block gathers environmental data: Distance (Ultrasonic HC-SR04), Water Level, Rain presence, and Temp & Humidity (DHT22). These inputs are fed into the **ESP32 Board**, which handles both WiFi and Servo Control. The processing unit distributes commands to three parallel output streams: **Alert Outputs** (Traffic Light LEDs and Buzzer for local warning), the **Servo Gate** (for physical path control), and the **Wi-Fi/Web Server** block, which pushes data to a Responsive Web Dashboard and sends Telegram Notifications to mobile devices.



Flow Chart

The software logic follows a continuous monitoring loop. Upon powering ON, the system initializes all components and displays a boot animation on the OLED before connecting to Wi-Fi. If connected, it starts the Web Server.

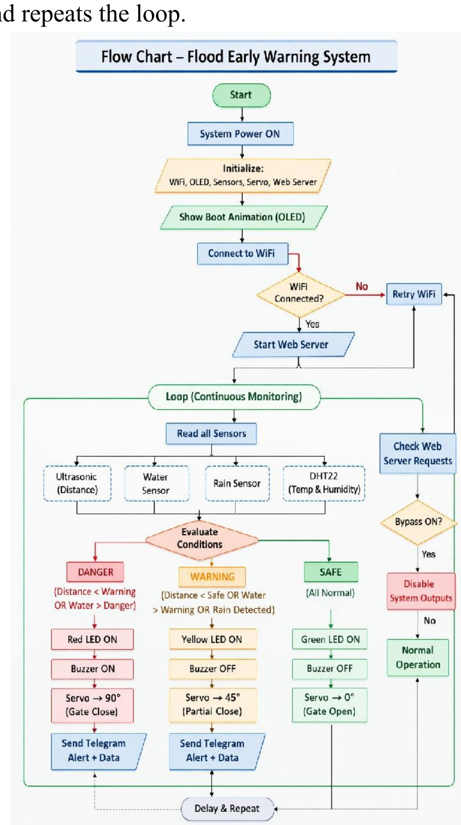
The main loop reads all sensors and checks for any "Bypass" requests from the Web Server (which can disable system outputs for manual override). If not bypassed, it evaluates the conditions:

DANGER: If (Distance < Warning OR Water > Danger). Action: Red LED ON, Buzzer ON, Servo 90°(Gate Close), Send Telegram Alert.

WARNING: If (Distance < Safe OR Water > Warning OR Rain Detected). Action: Yellow LED ON, Buzzer OFF, Servo 90° (Partial Close), Send Telegram Alert.

SAFE: All conditions normal. Action: Green LED ON, Buzzer OFF, Servo 0° (Gate Open).

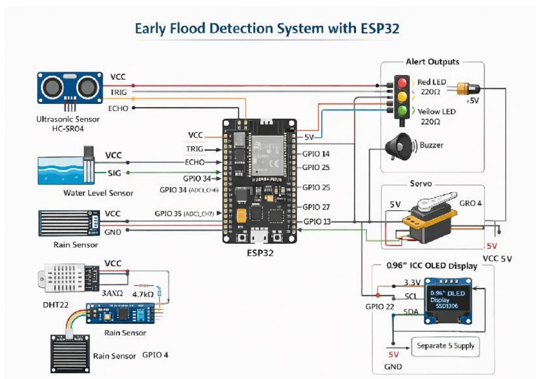
The system then executes a delay and repeats the loop.



IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

c) Circuit Diagram

The hardware connections utilize the versatile GPIOs of the ESP32. The **HC-SR04 Ultrasonic Sensor** uses dedicated digital pins for TRIG and ECHO to measure distance. The **Water Level Sensor** and **Rain Sensor** output analog signals routed to GPIO 34 and GPIO 35 (ADC pins) respectively. The **DHT22** uses a single digital pin with a $4.7\text{k}\Omega$ pull-up resistor. For outputs, the Traffic LEDs (Red, Yellow, Green) are connected via 220Ω current-limiting resistors, alongside a standard Buzzer. The **Servo Motor** receives PWM signals to control gate angles. A **$0.96"$ I2C OLED Display** is connected via SDA/SCL pins to provide localized data readout. A separate 5V supply line is utilized to ensure the servo and sensors receive adequate power without browning out the microcontroller.



Components / Materials

The following components form the backbone of the flood warning system:

ESP32 Development Board: A dual-core MCU with built-in Wi-Fi, managing sensor polling, web hosting, and Telegram API requests.

Ultrasonic Sensor (HC-SR04): Mounted above the water surface, it emits high-frequency sound waves to accurately calculate the distance to the rising water level.

Water Level & Rain Sensors: Resistive sensors that detect the presence and accumulation of water through variable conductivity.

DHT22 Sensor: Measures ambient temperature and humidity, providing contextual weather data.

Servo Motor: A rotary actuator that physically raises or lowers the warning gate based on the calculated threat level.

$0.96"$ OLED Display (SSD1306): An I2C screen that displays real-time sensor metrics and current system status at the physical location.

LEDs & Buzzer: A local visual (Red/Yellow/Green) and auditory alarm system to warn pedestrians and drivers nearby.

Working

The system operates autonomously in a high-frequency polling cycle:

Data Acquisition: The ESP32 triggers the ultrasonic sensor to measure the gap between the bridge/mount and the water surface. Simultaneously, it reads analog values from the rain and water level sensors to detect surface flooding.

State Evaluation: The software applies the threshold logic. For example, if heavy rain is detected and water levels begin to rise, the system shifts from SAFE to WARNING.

Local Actuation: In a WARNING state, the Yellow LED illuminates, and the servo motor rotates to 45°, acting as a cautionary barrier. If the water breaches the critical threshold (DANGER), the Red LED activates, the buzzer sounds continuously, and the servo rotates to 90°, fully closing the gate.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

Telemetry & Notifications: Concurrent with physical actuation, the ESP32 updates the local Web Dashboard with the live sensor metrics. It also formats an HTTP POST request to the Telegram API, instantly messaging the configured chat group with the alert level and live data.

Manual Override: An authorized user can access the Web Dashboard to toggle a "Bypass" state, allowing emergency vehicles to pass by overriding the sensor logic and manually opening the gate.

Results

The developed prototype demonstrated high reliability during testing:

Intent Recognition Accuracy: The dual-threshold logic successfully differentiated between unintentional artifacts, intentional quick blinks, and sustained muscle/brain holds with a high success rate.

Actuation Reliability: The digital control of the relay via pin D6 provided instantaneous toggling of the connected lightbulb upon meeting the 1.5-second hold criteria.

User Safety: The implementation of an isolated amplifier (BioAmp) and an opto-isolated relay successfully prevented any electrical feedback to the user, ensuring clinical-grade safety levels.

Connectivity Stability: The ESP32-S3 maintained a consistent link to the local Wi-Fi, allowing the PC/Smartphone dashboard to plot the bio-signals with minimal latency.

Advantages & Applications

ADVANTAGES

Hands-Free Control: Entirely eliminates the need for physical interaction with switches or buttons.

Highly Accessible: Empowers individuals with severe mobility impairments (e.g., ALS, quadriplegia) to control their environment.

Safe and Isolated: Medical-grade isolation ensures the user is entirely protected from mains voltages.

Dual-Function Input: Extracts multiple complex commands (pattern switching and power toggling) from a single analog input stream.

APPLICATIONS

Assistive Smart Homes: Controlling lights, fans, and alarms for bedridden patients.

Healthcare Facilities: Allowing patients to call for a nurse or adjust room settings independently.

Industrial Environments: Hands-free control of local equipment for operators whose hands are occupied or in sterile environments.

Future Scope

Machine Learning Integration: Moving beyond simple amplitude thresholds to implement TinyML on the UNO R4, allowing the system to recognize complex, specific thought patterns (e.g., thinking "left" vs. "right").

IoT Ecosystem Expansion: Connecting the ESP32-S3 directly to cloud platforms (like AWS IoT or Blynk) to control devices globally, rather than just locally.

Proportional Control: Using the intensity of the bio-signal to drive PWM outputs, allowing for proportional control like dimming a lightbulb or controlling the speed of a motorized wheelchair.

III. CONCLUSION

The Brain-Computer Interface system successfully addresses the limitations of physical switch-based environments by providing a secure, hands-free, and highly efficient control solution. By integrating the Arduino UNO R4 WiFi with the BioAmp sensor and opto-isolated relays, the project demonstrates a robust architecture for real-time biological signal processing and appliance actuation. The threshold-based logic ensures high intent-recognition reliability, while the Wi-

Fi capabilities align with modern smart home infrastructures. This system serves as a scalable, low-cost foundation for future advancements in assistive technologies and accessible living.

REFERENCES

[1] Arduino, "Arduino UNO R4 WiFi Datasheet: 32-bit RA4M1 and ESP32-S3 Architecture," *Arduino Hardware Documentation*, 2023.

[2] Upside Down Labs, "BioAmp EXG Pill / V1.5: Biopotential Amplifier Technical Specifications and Usage Guide," *Open Source Hardware Documentation*, 2023.

[3] Espressif Systems, "ESP32-S3 Series Datasheet: 2.4 GHz Wi-Fi and Bluetooth LE 5.0 SoC," Version 1.6, 2023.

[4] Renesas Electronics, "RA4M1 Group Microcontroller Hardware Manual," *Renesas Advanced (RA) Family Datasheets*, 2023.

[5] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767-791, 2002.

[6] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain Computer Interfaces, a Review," *Sensors*, vol. 12, no. 2, pp. 1211-1279, 2012.

[7] M. Teplan, "Fundamentals of EEG measurement," *Measurement Science Review*, vol. 2, no. 2, pp. 1-11, 2002.

[8] A. Al-Fuqaha, M. Guizani, and M. Mohammadi, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015.

[9] N. Thakor, "Biopotential Amplifiers," in *Medical Devices and Systems*, 3rd ed., CRC Press, 2006, pp. 14-1–14-16.

[10] S. Sanei and J. A. Chambers, *EEG Signal Processing*. West Sussex, U.K.: John Wiley & Sons, 2007.

[11] P. K. Singh et al., "Artifact Removal from EEG Signals: A Comprehensive Review," *IEEE Sensors Journal*, vol. 20, no. 15, pp. 8190-8205, 2020.

[12] K. M. A. Hasan, M. A. Hasan, and M. Z. Islam, "Design and implementation of an eye-blink controlled smart home system for paralyzed patients," *Proceedings of the IEEE International Conference on Robotics, Electrical and Signal Processing Techniques*, pp. 45-50, 2019.

[13] R. Barea, L. Boquete, M. Mazo, and E. López, "System for assisted mobility using eye movements based on electrooculography," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 4, pp. 209-218, 2002.

[14] A. K. Jain and P. Sharma, "Smart Home Automation System using IoT and BCI Integration," *International Journal of Advanced Research in Computer Science*, vol. 9, no. 2, pp. 150-155, 2018.

[15] T. C. Handy, *Event-Related Potentials: A Methods Handbook*. Cambridge, MA, USA: MIT Press, 2005.

[16] Vishay Intertechnology, "Optocouplers in Solid-State Relays: Safety and Isolation Standards," *Application Note 56*, 2021.

[17] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: A review," *Computer Methods and Programs in Biomedicine*, vol. 161, pp. 1-13, 2018.

[18] S. S. Sivaraman, "Real-Time Telemetry and Data Logging in ESP32-based IoT Systems," *Proceedings of the IEEE International Conference on Computation System (CSITSS)*, pp. 1-5, 2022.

[19] C. Guger, S. Daban, E. Sellers, C. Holzner, G. Krausz, R. Carabalona, F. Gramatica, and G. Edlinger, "How many people are able to control a P300-based brain-computer interface (BCI)?," *Neuroscience Letters*, vol. 462, no. 1, pp. 94-98, 2009.

[20] H. M. Yasin, "A Comprehensive Review on IoT-Based Assistive Systems with Real-Time Medical Monitoring," *International Journal of Advanced Research in Science (IJARSCT)*, vol. 4, no. 2, pp. 1823-1830, 2024.