

## 1.Introduction



The **Flood Monitoring and Alerting System** is designed to overcome existing challenges by providing a modern, automated approach to water level monitoring. The system ensures continuous observation of water bodies and enables timely alerts to residents, authorities, and emergency response teams, helping them prepare and respond effectively to potential flood situations.

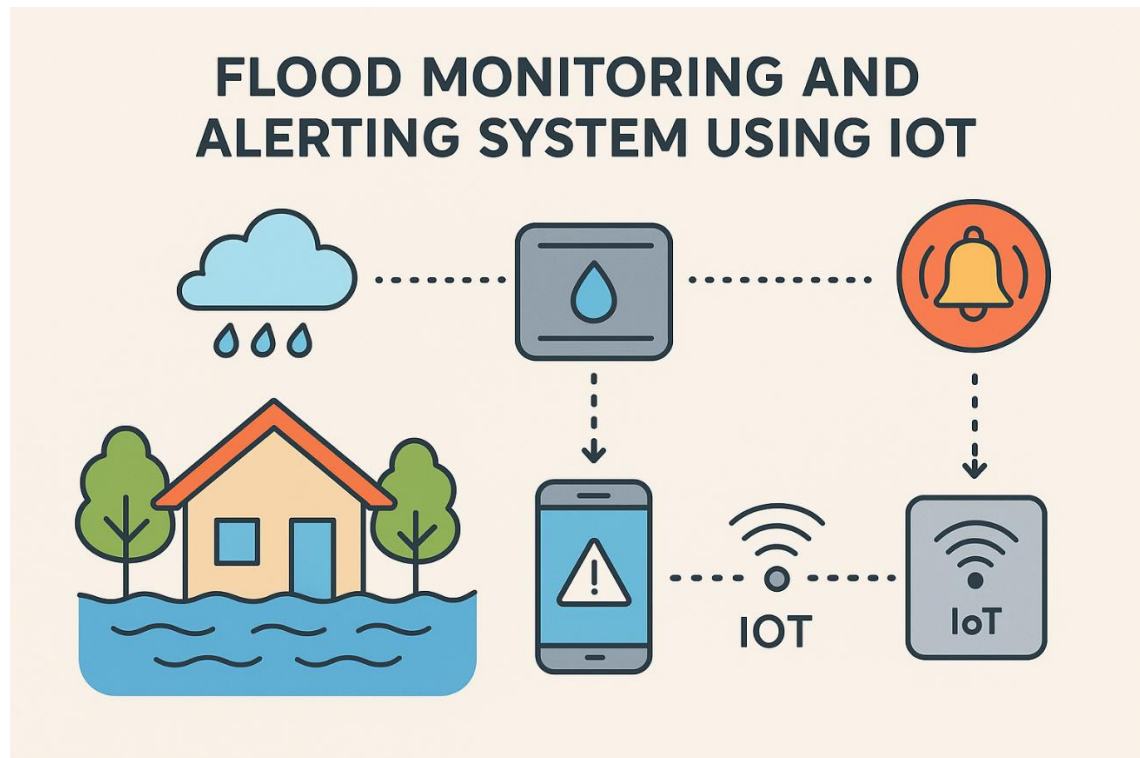
Flood is one of the major natural disasters, and when the water level suddenly rises in dams, river beds, and reservoirs, it causes large-scale destruction in the surrounding areas. This results in huge losses to both the environment and living beings. In such cases, it is very important to receive emergency alerts about the water level situation in different conditions.

Floods often lead to significant damage to human life, property, and the environment. Sudden rises in water levels can create dangerous situations for nearby communities. Early detection and timely warning are essential to minimize the impact of floods and to ensure the safety of people living in flood-prone areas.

Traditional flood monitoring relies heavily on manual observation, which is often slow, inefficient, and unsafe during extreme weather conditions. People may not become aware of rising water levels in time to take preventive action, which can lead to loss of life and property.

The purpose of this project is to sense the water level in river beds and check if it remains within normal conditions. If the level rises beyond the limit, the system alerts people through LED signals and buzzer sound. Additionally, it also provides SMS and Email alerts when the water level exceeds the threshold, ensuring timely warnings and improved safety.

This system not only reduces human effort but also enhances the reliability of flood monitoring. By integrating sensors with automated alert mechanisms, it provides a practical and efficient solution for disaster management. This project has the potential to be implemented in real-time flood-prone areas, ultimately saving lives and minimizing damage.



## **2.Literature Survey / Existing Systems**

### **1. Smart Flood Monitoring with GSM Communication**

A study by Kaphungkui et al. presents an IoT-based flood monitoring system utilizing Arduino Uno microcontrollers, ultrasonic sensors, and GSM modules. This system measures water levels and sends SMS alerts to local populations when thresholds are exceeded. The modular design ensures scalability and energy efficiency, making it suitable for remote areas.

### **2. Integrated Flood and Weather Monitoring**

Research by Sushma et al. introduces a system that combines water level, rainfall, temperature, and humidity sensors with an Arduino Uno microcontroller. Data is transmitted to the ThingSpeak cloud platform via an ESP8266 Wi-Fi module, enabling real-time monitoring. SMS alerts are sent through a GSM module when predefined thresholds are breached.

### **3. Flood Monitoring with Flow Sensors and Blynk Integration**

Patil et al. propose a system using Arduino Mega boards, water level and flow sensors, and the Blynk application for remote monitoring. The system calculates water flow rates and predicts flood arrival times, providing timely alerts to authorities and residents.

### **4. IoT-Based Flood and Weather Monitoring System**

A project by Sushma et al. focuses on detecting environmental parameters such as water levels, rainfall, temperature, and humidity using an Arduino Uno microcontroller. The system triggers SMS alerts via a GSM module and displays real-time data on a web-based dashboard for remote monitoring.

### **5. Wireless Sensor Networks in Disaster Management**

Adeel et al. survey the role of wireless sensor networks (WSNs) and IoT in disaster management. They discuss how these technologies enable real-time data collection and monitoring, enhancing disaster prediction and management capabilities

### **3.Problem Statement**

#### **1. Lack of Real-Time Flood Data Collection**

Many existing flood monitoring systems rely on manual data collection or delayed reports, leading to untimely flood alerts. There is a need for automated, real-time monitoring systems to provide instant updates on water levels and weather conditions.

#### **2. Inadequate Early Warning and Alert Systems**

Flood warning systems in many regions are either absent or inefficient, resulting in delayed evacuation and increased loss of life and property. There is a need for an accurate and reliable alerting mechanism that can notify authorities and residents in advance.

#### **3. Limited Coverage in Remote and Flood-Prone Areas**

Many flood-prone areas, especially in rural or remote regions, lack proper monitoring infrastructure due to connectivity and power challenges. IoT-based systems need to be scalable, low-power, and capable of operating in such environments.

#### **4. Integration of Multiple Environmental Parameters**

Flood events are often influenced by various factors like rainfall, water flow, humidity, and temperature. Existing systems often monitor only water levels. A comprehensive flood monitoring system should integrate multiple sensors for holistic environmental monitoring.

#### **5. High Cost and Maintenance of Flood Monitoring Infrastructure**

Conventional flood monitoring infrastructure can be expensive and require regular maintenance. There is a need for cost-effective, low-maintenance IoT devices that can operate reliably over long periods.

#### **6. Lack of Cloud-Based Data Analytics and Visualization**

Many systems lack a centralized platform to store, analyze, and visualize flood data for decision-making. Incorporating cloud platforms and real-time dashboards can improve accessibility and response efficiency.

## **4.Objectives**

### **1. Real-Time Water Level Monitoring**

- Continuously tracks water levels using sensors like ultrasonic to detect sudden rises immediately.
- Enables quick awareness for residents and authorities, reducing reliance on slow, manual observation methods.

### **2. Automated Alert System**

- Automatically triggers visual and audible alerts (LEDs, buzzers) when water crosses a set threshold.
- Provides instant warnings to nearby people, helping prevent accidents and property damage.

### **3. Simple and Cost-Effective Design**

- Uses affordable and easily available components to keep the overall system low-cost.
- Designed for easy maintenance and operation without requiring specialized technical skills.

### **4. Enhance Local Safety Awareness**

- Raises community awareness by signaling flood risks promptly through alerts.
- Encourages quick safety actions, reducing injury and property loss at the local level.

### **6. Scalability and Integration**

- Design the system to be easily scalable, allowing multiple sensor nodes to cover large or multiple flood-prone areas.
- Integrate with existing early warning systems and weather forecast APIs to improve flood prediction accuracy and response.

## 5. Software requirement specification

### 1. Functional Requirements

Functional requirements describe what the system should do:

- Collect real-time data from water level sensor and DHT11 (temperature & humidity) sensor connected to the ESP32.
- Display real-time water level and environmental parameters on the OLED SSD1306 display.
- Trigger visual alerts (Green, Orange, Red LEDs) based on water level thresholds:
  - Green → Safe
  - Orange → Warning
  - Red → Danger
- Trigger an audible alert using a buzzer when the water level crosses the danger threshold.

### 2. Non-Functional Requirements

Non-functional requirements describe how the system performs:

- Reliability: Must continuously monitor water levels and environmental conditions without frequent errors.
- Usability: Provide simple, clear visual (OLED & LEDs) and audio (buzzer) alerts.
- Scalability: Support adding more sensors (e.g., rainfall sensor, soil moisture sensor) with minimal modifications.
- Maintainability: Code should be modular, with separate functions for sensor reading, display, and alert handling.
- Portability: System should run smoothly on standard ESP32 hardware without extra configuration.

### 3. User Requirements

- Users should be able to view water level, temperature, and humidity values on the OLED display.
- Users should understand flood risk status through LED indications and buzzer alerts.
- Users should be able to configure threshold values in the code if required.

### 4. Hardware Requirements

- ESP32 – Microcontroller for sensor interfacing and processing.
- Water Level Sensor / Ultrasonic Sensor (HC-SR04) – To detect water level.
- DHT11 Sensor – To measure temperature and humidity.
- OLED Display (SSD1306) – For real-time display of sensor data.
- LEDs (Green, Orange, Red) – For visual alerts.
- Buzzer – For audible alerts.
- Power Supply (5V/3.3V) – To power ESP32 and sensors.
- Breadboard and Jumper Wires – For prototyping.

### 5. Software Requirements

- Thonny IDE – To write and upload MicroPython code to the ESP32.
- MicroPython firmware – To enable programming of the ESP32.
- Libraries:
  - SSD1306 OLED display library
  - DHT11 sensor library
  - GPIO control libraries for LEDs and buzzer

### 6. Performance Requirements

- LEDs and buzzer must trigger immediately when threshold levels are crossed.
- OLED display should update readings in real time (every few seconds).





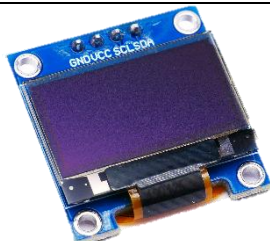
- The system should run continuously for at least 24 hours without failure.
- Alerts should be noticeable from a reasonable distance (buzzer loudness + LED brightness).

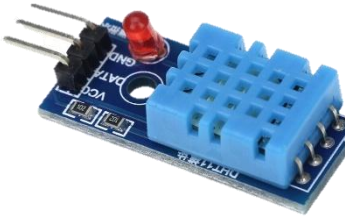
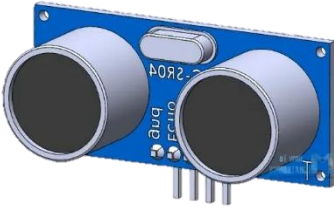


## **7. Other Requirements**

- The system should handle sensor errors (e.g., invalid readings) gracefully.
- The project should be documented for easy installation, wiring, and code understanding.
- The design should allow expansion for additional features (e.g., IoT cloud monitoring in the future).
- The hardware setup should be compact and energy-efficient.



## 6. Hardware requirement specification

Component	Specification / Details	Purpose in Project
ESP32 Development Board		Acts as the main microcontroller, responsible for processing data and controlling outputs.
Breadboard (Generic)		For prototyping and connecting all components without soldering.
5 mm LED (Green, Orange, Red)		Visual indicators for safe (green), warning (orange), and danger/flood (red) water levels.
Buzzer		Audible alert to indicate water level crossing thresholds.
OLED Display (SSD1306)		Display real-time water level readings and status messages.

Component	Specification / Details	Purpose in Project
<b>DHT11 Sensor</b>		(Optional) Could monitor temperature to correlate environmental conditions with water levels.
<b>Ultrasonic Sensor - HC-SR04</b>		Measures water level distance accurately without contact.
<b>Jumper Wires (Male/Female, Male/Male, Female/Female)</b>		Used for connecting sensors, LEDs, buzzer, LCD, and Arduino on breadboard.
<b>5V Battery</b>		Powers the Arduino and connected components when USB is not used.

## 7. Methodology

### 1. Project Overview

The Flood Monitoring and Alerting System is an IoT-based solution that continuously measures water levels using an ultrasonic sensor and environmental parameters using a DHT11 sensor. The system provides real-time monitoring on an OLED display and issues early warnings through LEDs and a buzzer when critical thresholds are crossed. This ensures timely alerts to prevent flood damage, making the system reliable, automated, and efficient for flood-prone areas.

### 2. Data Set Required

- Water level distance measurements (from Ultrasonic Sensor HC-SR04).
- Calculated water level percentage.
- Temperature and humidity data (from DHT11).
- Alert status logs (Green, Orange, Red zones).
- Buzzer activation status (timestamps of danger alerts).

### 3. Method of Data Collection

- Ultrasonic sensor continuously measures the distance to the water surface.
- ESP32 processes the data and converts it into water level percentage.
- DHT11 sensor records temperature and humidity for additional context.
- Processed data is displayed in real-time on the OLED SSD1306.
- Alerts are triggered locally using LEDs and buzzer whenever thresholds are exceeded.

### 4. Algorithm / Working Steps

#### Step 1: Initialization

- Initialize ESP32, ultrasonic sensor, DHT11 sensor, OLED display, LEDs, and buzzer.
- Set predefined threshold values for water levels (Safe, Warning, Danger).

**Step 2: Data Collection**

- Continuously measure water level using ultrasonic sensor.
- Convert sensor data into percentage values of water level.
- Collect temperature and humidity data from DHT11.

**Step 3: Decision Making and Alerts**

- Categorize water levels into three zones:
  - Green → Safe
  - Orange → Warning
  - Red → Danger
- Activate the corresponding LED based on the zone.
- Trigger **buzzer** when water level crosses the danger threshold.
- Display real-time water level, temperature, humidity, and alert status on the OLED screen.

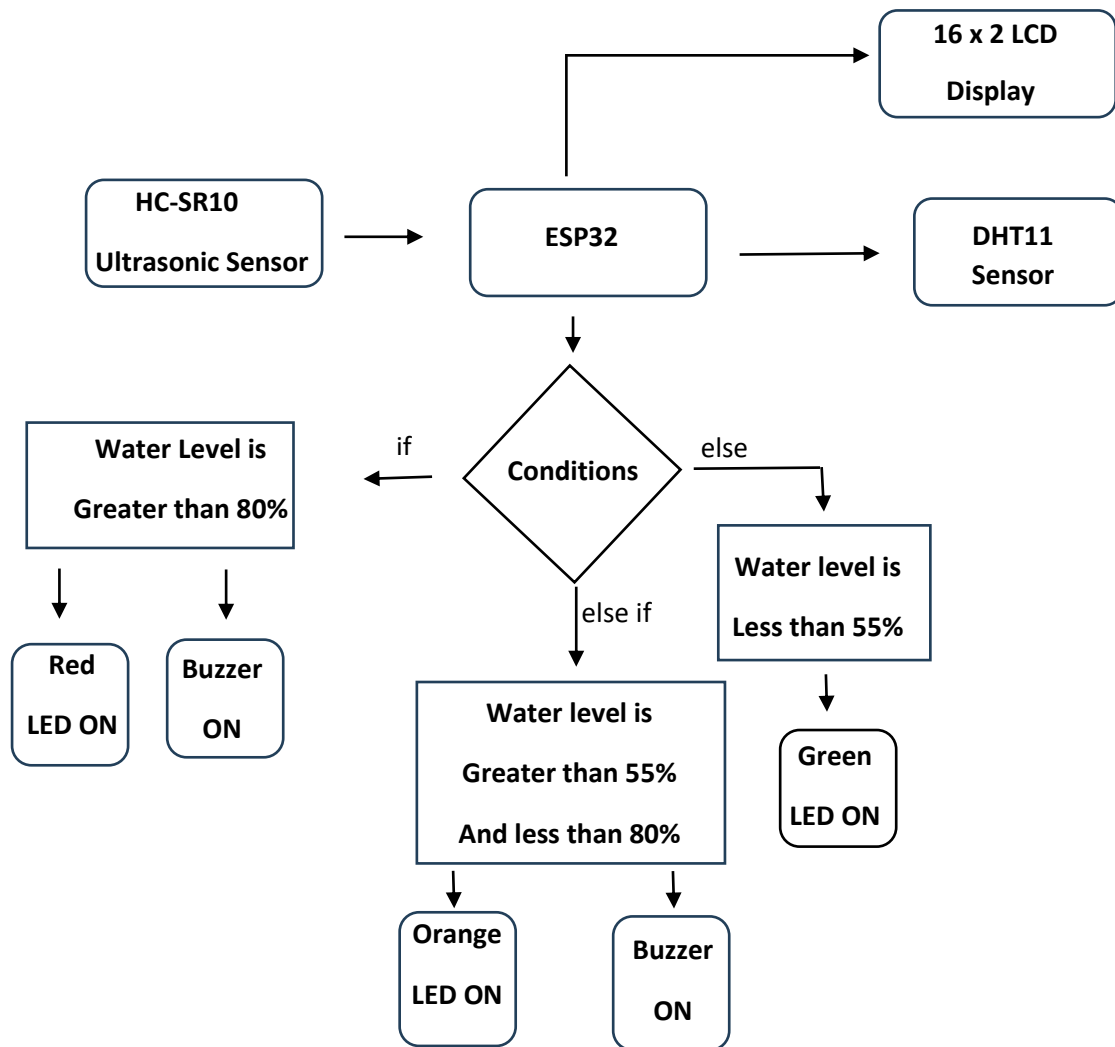
**Step 4: Local Monitoring & Visualization**

- Users can observe real-time readings directly on the OLED display.
- LED and buzzer provide clear indications of flood risk levels.

**Step 5: Continuous Monitoring**

- The system repeats the measurement and decision-making process every few seconds.
- Alerts are generated instantly whenever thresholds are exceeded, ensuring timely flood warnings.

## 5. Block diagram



## 8.References

### Online references:

<https://youtu.be/NEZtcNrtXKA?si=8kgfcA8xMj4w93Qj>

### Offline references:

#### 1. "Design Analysis of an IoT-based Early Flood Detection and Alerting System"

*Authors:* Okeke Remigius Obinna, Ehikhamenle Mathew

*Published in:* International Journal of Advanced Engineering Research and Science, 2024

#### 2."IoT-based Flood Monitoring and Alerting System using Raspberry Pi"

*Authors:* K. Subramanya Chari, Maturi Thirupathi, Ch Hariveena

*Published in:* IOP Conference Series: Materials Science and Engineering, 2020

#### 3."Flood Detection and Alerting System Using IoT"

*Authors:* B. Priyanka, G. Pavithra, M. Nivetha, P. Shreeswetha, S. Saravanan

*Published in:* International Journal of Science and Research in Technology, 2021

**Project Guide**

**Mr. N. V. Sarnaik**

**Project Co-ordinator**

**Mr. R. S. Todkar**

**HOD (AIML)**

**Mr. R. S. Kamble**