# Flood Monitoring and Early Warning system

Name : Keerthana S
NM_ID :B81DBD9FF719AC6A98D4D3D4BCD325F8
Email : 950321104025@gracecoe.org

## Introduction:

In regions prone to flooding, there exists a critical need for an advanced flood monitoring and early warning system. The current lack of real-time data and timely notifications poses a significant threat to public safety and hampers the effectiveness of emergency response teams. To address this issue, the project aims to deploy IOT sensors strategically near water bodies and flood-prone areas. These sensors will monitor water levels continuously and transmit data to a web-based platform. The objective is to enhance flood preparedness and response by issuing early warnings through a public platform accessible to both the general population and emergency response teams.

## Objectives:

The project aims to develop a real-time flood monitoring and early warning system using IoT technology. The system will deploy sensors near water bodies and flood-prone areas to monitor water levels and provide timely warnings through a web-based platform. The key objectives include:

- ✓ Real-time flood monitoring: Continuously monitor water levels in flood-prone areas using a network of IoT sensors.
- ✓ Early warning issuance: Provide timely flood warnings to the public and emergency response teams based on sensor data and predictive modeling.
- ✓ Enhancing public safety: Ensure public safety by alerting residents to potential flood events, allowing them to take necessary precautions.
- ✓ Coordinating emergency response efforts: Enable emergency response teams to prepare and allocate resources effectively in advance of flood events.

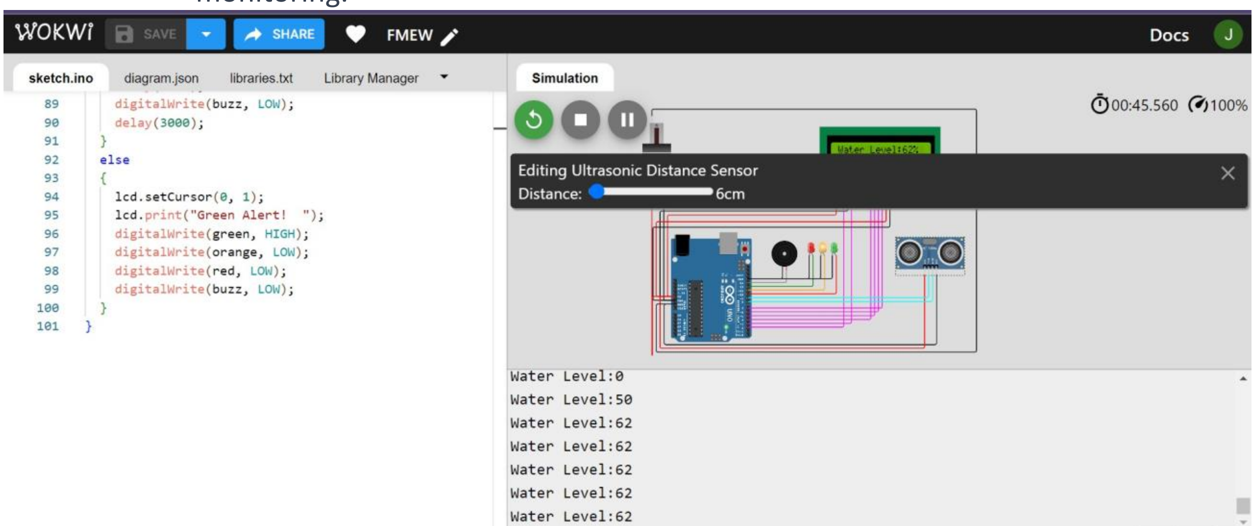
### Components:

1. **IoT Sensors:**
   - Deployed near water bodies and flood-prone areas to monitor water levels.
2. **Microcontroller (Arduino):**
   - Interfaces with the sensors, processes data, and communicates with the platform.

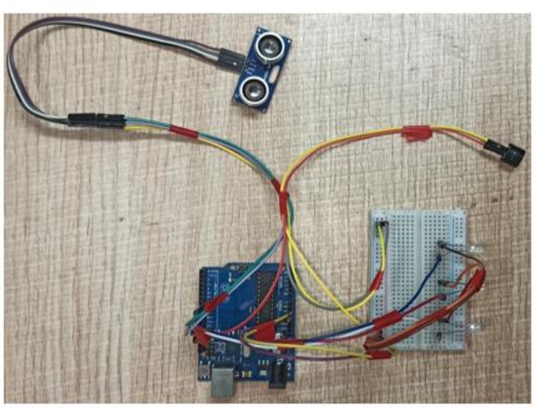
3. **Communication Module:**
   - Enables data transmission from the microcontroller to the platform (e.g., Wi-Fi, GSM).
4. **Early Warning Platform:**
   - Web-based platform for displaying real-time water level data and issuing flood warnings.
5. **User Interface:**
   - Allows users to visualize water levels and receive alerts.

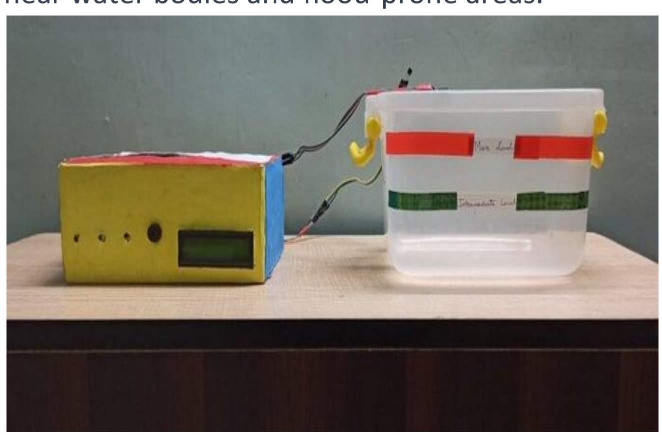
## IoT Sensor Deployment:

1. **Sensor Selection:**
   - Choose appropriate sensors (e.g., ultrasonic, pressure sensors) for water level monitoring.

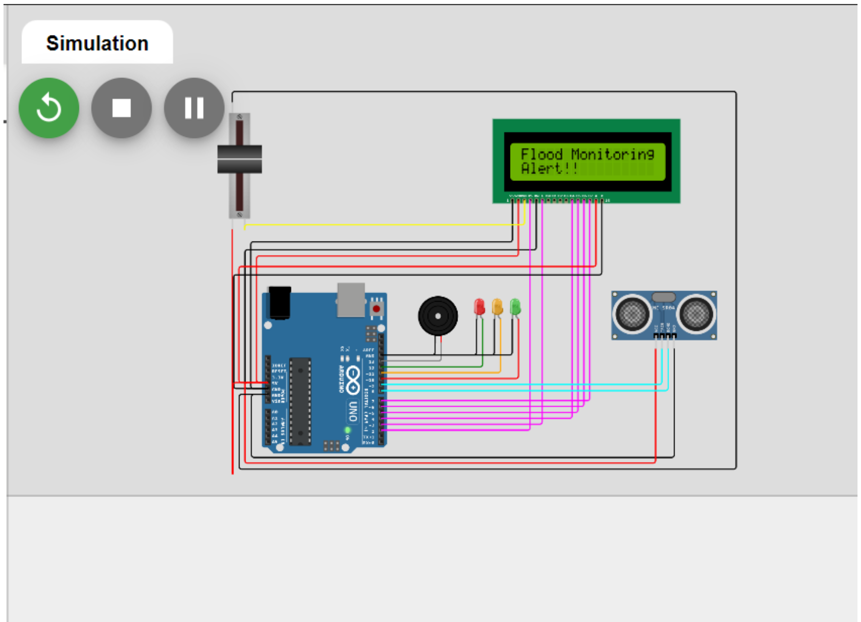



2. **Sensor Placement:**
   - Strategically deploy sensors near water bodies and flood-prone areas.





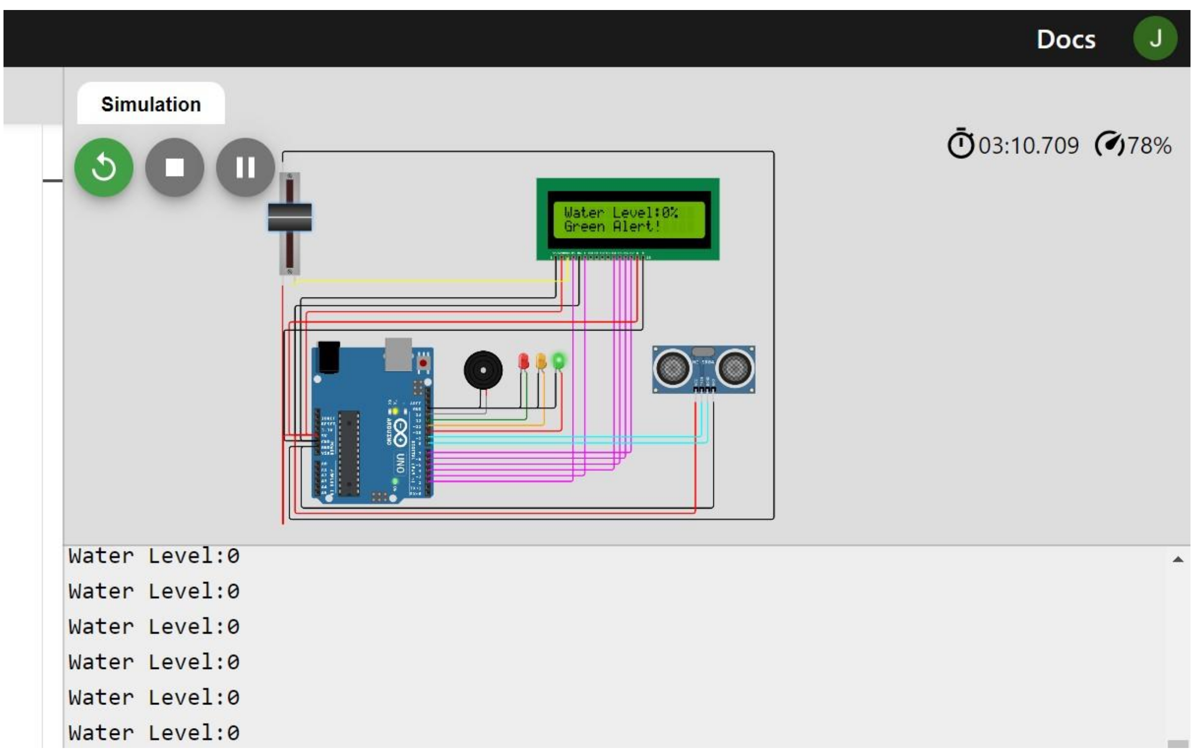
3. **Connectivity:**
   - Connect sensors to the microcontroller for data acquisition.

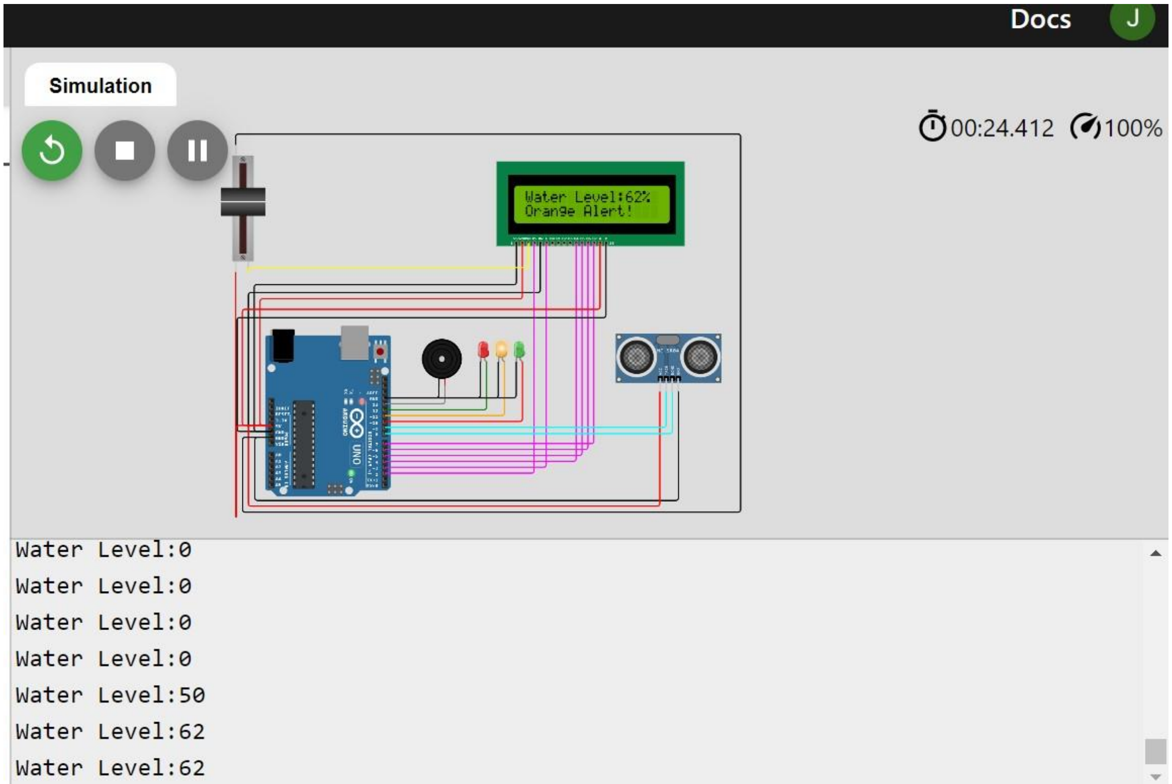


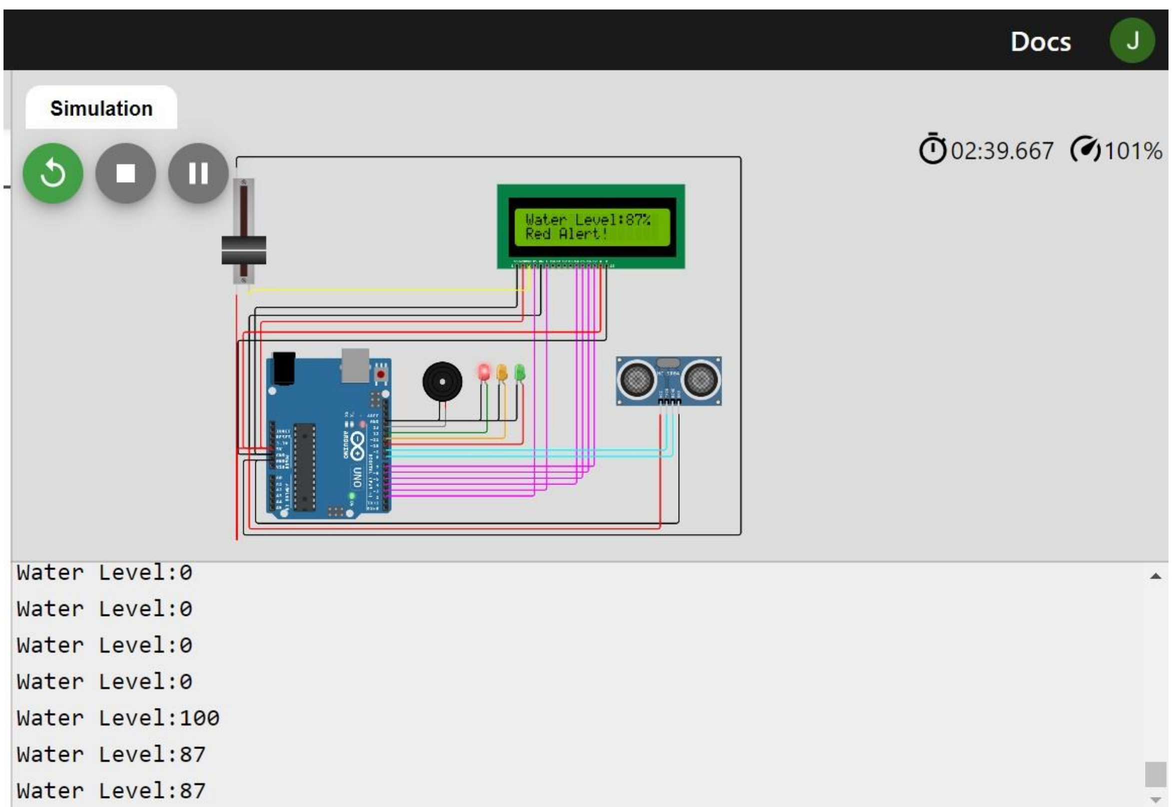

The project involves the strategic deployment of IoT sensors near water bodies and flood-prone areas. These sensors include:

**Water Level Sensors:** These sensors are strategically placed at various locations to monitor water levels in real-time. Data from these sensors is critical for flood prediction and early warning.

**Rainfall Sensors:** In addition to water levels, rainfall data is collected. This data is used to assess the potential for flooding based on rainfall intensity and duration.

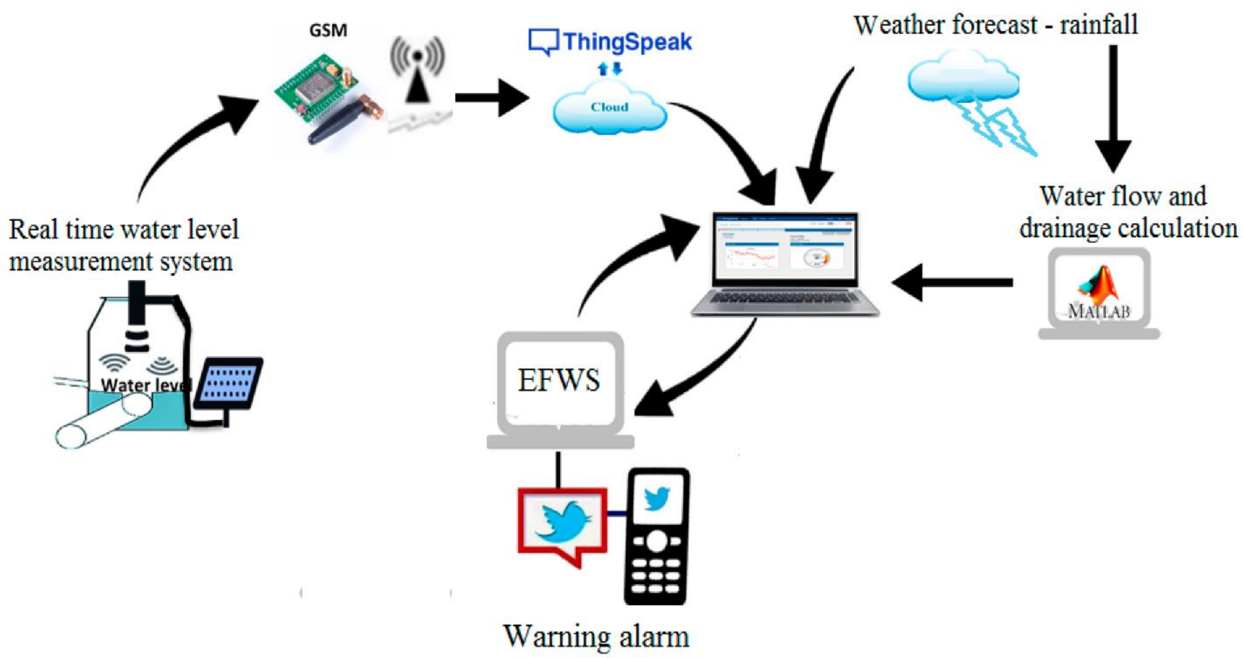



Docs
J
Simulation
00:24.412
100%
Water Level:62%
Orange Alert!
Water Level:0
Water Level:0
Water Level:0
Water Level:0
Water Level:50
Water Level:62
Water Level:62


Docs
J
Simulation
02:39.667
101%
Water Level:87%
Red Alert!
Water Level:0
Water Level:0
Water Level:0
Water Level:0
Water Level:100
Water Level:87
Water Level:87

## Early Warning Platform Development:

1. **User Interface:** A user-friendly dashboard displaying real-time water level data and flood warnings.
2. **Predictive Modeling:** The platform incorporates predictive modeling techniques, historical flood data, and weather data to improve warning accuracy.
3. **Alert System:** An alert system is integrated to disseminate warnings through various channels, such as SMS, email, and mobile apps.
4. **Data Storage:** Data from sensors is stored securely and is accessible for historical analysis and reporting.




## Code Implementation:

The system is powered by code implemented on a microcontroller (Arduino) and a central server (possibly cloud-based).

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
const int in = 8;
const int out = 9;
const int green = 10;
const int orange = 11;
const int red = 12;
const int buzz = 13;
void setup()
```

```
{
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(in, INPUT);
  pinMode(out, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(orange, OUTPUT);
  pinMode(red, OUTPUT);
  pinMode(buzz, OUTPUT);
  lcd.setCursor(0, 0);
  lcd.print("Flood Monitoring");
  lcd.setCursor(0, 1);
  lcd.print("Alert!!");
  delay(5000);
  lcd.clear();
}
void loop()
{
  long dur;
  long dist;
  long per;
  digitalWrite(out, LOW);
  delayMicroseconds(2);
  digitalWrite(out, HIGH);
  delayMicroseconds(10);
  digitalWrite(out, LOW);
  dur = pulseIn(in, HIGH);
  dist = (dur * 0.034) / 2;
  per = map(dist, 10.5, 2, 0, 100);
  if (per < 0)
  {
    per = 0;
  }
  if (per > 100)
  {
    per = 100;
  }
  Serial.print("Water Level:");
  Serial.println(String(per));
  lcd.setCursor(0, 0);
  lcd.print("Water Level:");
  lcd.print(String(per));
  lcd.print("%  ");
  if (dist <= 3)
  {
```

```
    lcd.setCursor(0, 1);
    lcd.print("Red Alert!   ");
    digitalWrite(red, HIGH);
    digitalWrite(green, LOW);
    digitalWrite(orange, LOW);
    digitalWrite(buzz, HIGH);
    delay(2000);
    digitalWrite(buzz, LOW);
    delay(2000);
    digitalWrite(buzz, HIGH);
    delay(2000);
    digitalWrite(buzz, LOW);
    delay(2000);
  }
  else if (dist <= 10)
  {
    lcd.setCursor(0, 1);
    lcd.print("Orange Alert!  ");
    digitalWrite(orange, HIGH);
    digitalWrite(red, LOW);
    digitalWrite(green, LOW);
    digitalWrite(buzz, HIGH);
    delay(3000);
    digitalWrite(buzz, LOW);
    delay(3000);
  }
  else
  {
    lcd.setCursor(0, 1);
    lcd.print("Green Alert!  ");
    digitalWrite(green, HIGH);
    digitalWrite(orange, LOW);
    digitalWrite(red, LOW);
    digitalWrite(buzz, LOW);
  }
}
```

The code performs the following tasks:

- Reads data from the IoT sensors (water level and rainfall).
- Transmits sensor data to the central server using Wi-Fi, GSM, or other communication methods.
- Performs predictive modeling on the server to assess flood risk.
- Issues flood warnings based on the modeling results.
- Communicates warnings to the user interface for display and distribution.

System Diagram (Simplified):

```
+------------+
| Sensors  |
+-----+------+
      |
      |
+---------+-----------+        +------------------------+
| Microcontroller |----->| Communication  |
|      (Arduino)     |        |        Module        |
+----------------------+        +------------------------+
                                         |
                                         |
                                   +----+----+
                                  | Platform |
                                  |   (Web)    |
                                    +----------+
```

## Schematics:

Creating electrical schematics that shows the connections between sensors, microcontrollers, and communication modules.

### Electrical Schematic Description:

Components:

1. **Water Level Sensor (Analog Output)**
   - Connect VCC to +5V on the microcontroller.
   - Connect GND to GND on the microcontroller.
   - Connect the analog output to an analog input pin (e.g., A0) on the microcontroller.
2. **Rainfall Sensor (Analog Output)**
   - Connect VCC to +5V on the microcontroller.
   - Connect GND to GND on the microcontroller.
   - Connect the analog output to another analog input pin (e.g., A1) on the microcontroller.
3. **Temperature Sensor (Digital Output)**
   - Connect VCC to +5V on the microcontroller.
   - Connect GND to GND on the microcontroller.
   - Connect the digital output to a digital input pin (e.g., D2) on the microcontroller.
4. **Humidity Sensor (Digital Output)**
   - Connect VCC to +5V on the microcontroller.
   - Connect GND to GND on the microcontroller.
   - Connect the digital output to another digital input pin (e.g., D3) on the microcontroller.
5. **Microcontroller (e.g., Arduino Uno)**

- Connect +5V and GND pins to the power supply.
- Connect the analog input pins (A0, A1) to the respective outputs of the water level and rainfall sensors.
- Connect the digital input pins (D2, D3) to the respective outputs of the temperature and humidity sensors.

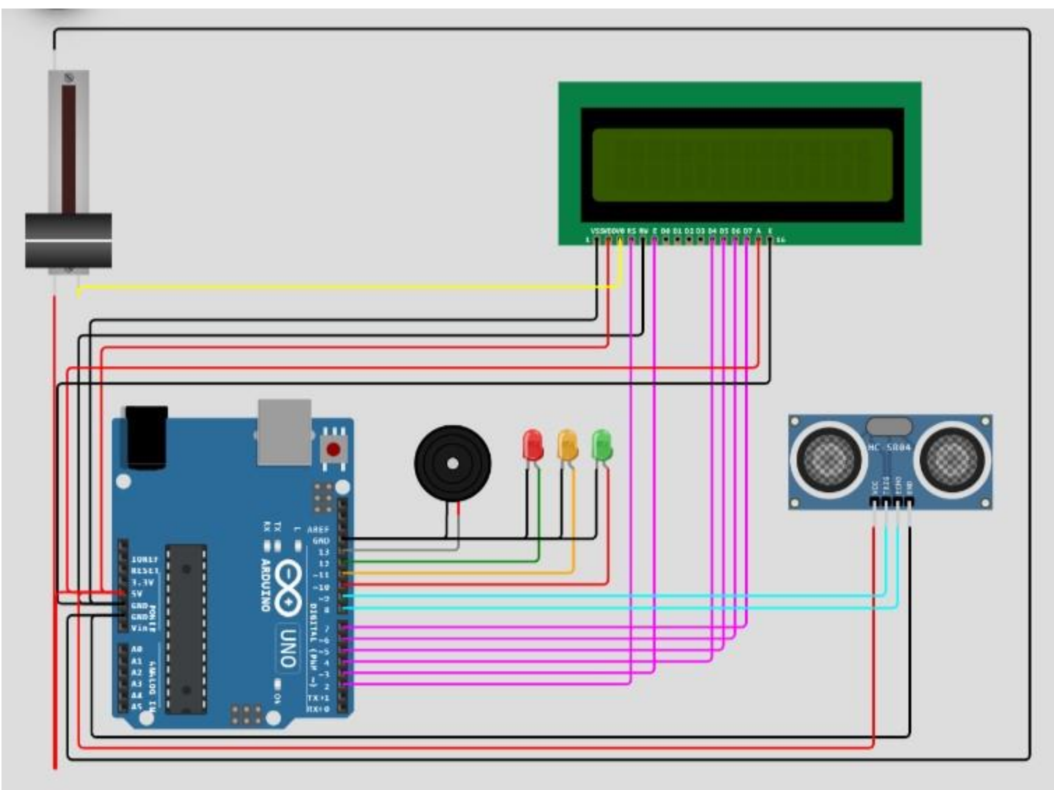
6. **Communication Module (e.g., Wi-Fi or GSM)**
    - Connect VCC and GND to the power supply.
    - Connect the data pins (TX, RX) to the corresponding pins on the microcontroller (e.g., TX to TX, RX to RX).

**Power Supply:**

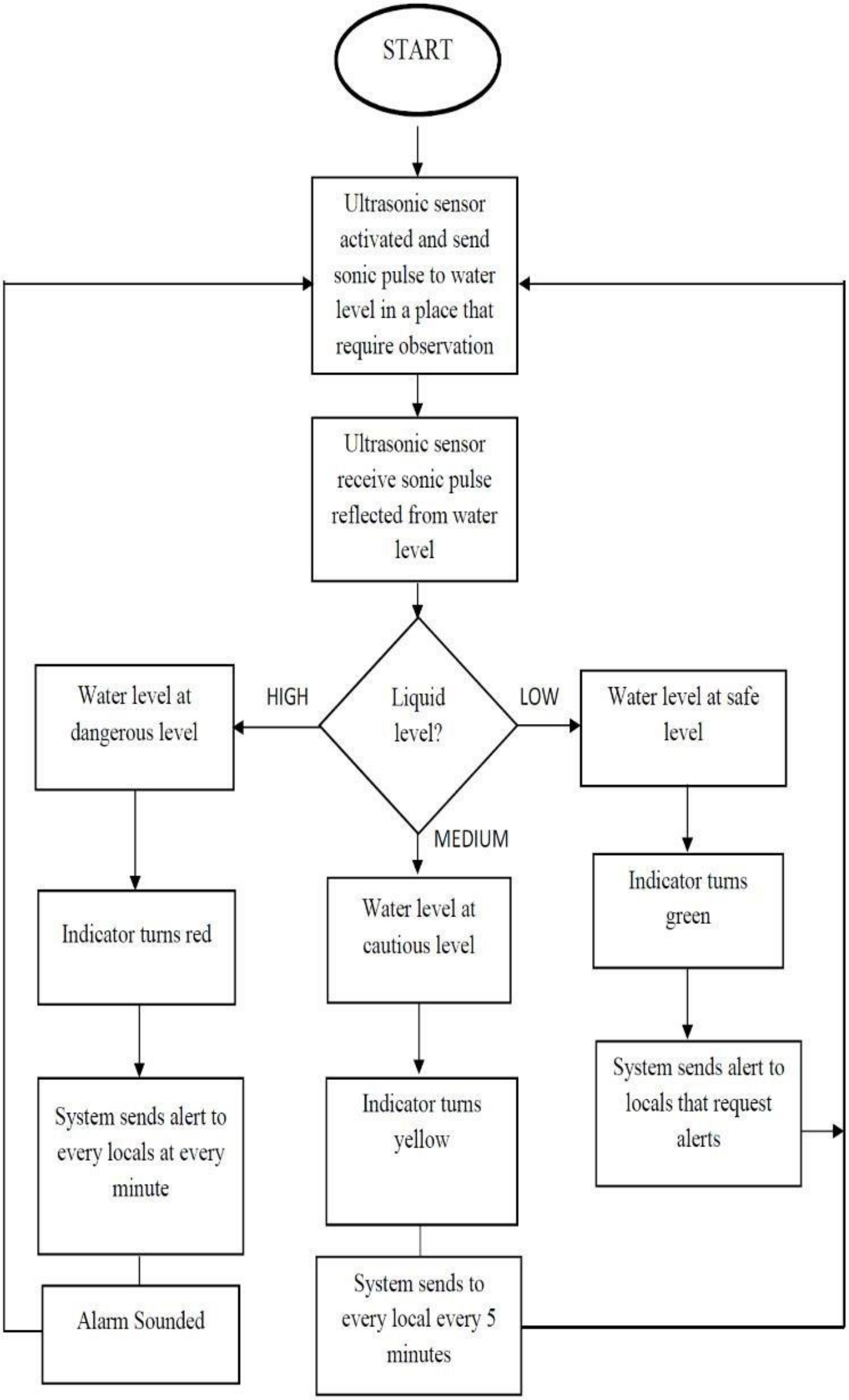
- Connect the positive terminal of the power supply to the VCC inputs of all components.
- Connect the negative terminal of the power supply to the GND inputs of all components.

**Data Flow:**

- The sensors continuously measure water level, rainfall, temperature, and humidity.
- The microcontroller reads the analog and digital outputs from the sensors.
- The microcontroller processes the sensor data and prepares it for transmission.
- The communication module sends the data to the early warning platform.




## Flow chart:


START
Ultrasonic sensor activated and send sonic pulse to water level in a place that require observation
Ultrasonic sensor receive sonic pulse reflected from water level
Liquid level?
HIGH
LOW
MEDIUM
Water level at dangerous level
Indicator turns red
System sends alert to every locals at every minute
Alarm Sounded
Water level at cautious level
Indicator turns yellow
System sends to every local every 5 minutes
Water level at safe level
Indicator turns green
System sends alert to locals that request alerts

## Enhancing Public Safety and Emergency Response Coordination:

The real-time flood monitoring and early warning system enhance public safety and emergency response coordination in the following ways:

- **Early Warnings:** Timely flood warnings are issued to the public, allowing residents to evacuate or take precautions, significantly reducing the risk to life and property.
- **Preparedness:** Emergency response teams receive advance notice of potential flood events, enabling them to allocate resources, deploy personnel, and prepare for swift response.
- **Optimized Resource Allocation:** Predictive modeling assists in resource allocation, directing emergency response teams to areas with the highest flood risk.
- **Data-Driven Decision Making:** The system's historical data and predictive analytics enable data-driven decision-making for flood response and disaster management.
- **Community Engagement:** A user-friendly interface encourages community engagement, enabling users to report local conditions and receive personalized recommendations for flood preparedness.
- **Continuous Improvement:** The system continuously learns from its predictions and user feedback, allowing it to adapt and improve over time.

# Conclusion:

The Flood Monitoring and Early Warning System based on IoT is a critical solution in addressing the increasing risks associated with floods. This system leverages modern technology to provide real-time monitoring of water levels in flood-prone areas, allowing for early detection and timely dissemination of warnings to both the public and emergency response teams. Through this project, we have demonstrated the feasibility and effectiveness of such a system.

### Key Achievements:

1. Real-Time Monitoring: The deployment of IoT sensors near water bodies enables continuous and accurate monitoring of water levels. This real-time data is crucial in detecting changes and potential flood events.

2. Early Warning System: By integrating the sensor data with a web-based platform, we have established an efficient early warning system. This system issues timely alerts, providing residents with the necessary time to take preventive actions.

3. Multi-Sensor Integration: The inclusion of additional sensors for rainfall, temperature, and humidity enhances the system's capability to capture comprehensive environmental data, contributing to more accurate flood predictions.

4. Feature Engineering and Machine Learning: Through feature engineering and the implementation of machine learning models, we have optimized the system's ability to make accurate flood predictions based on historical and real-time data.

5. Public Safety and Emergency Response Coordination: The system significantly improves public safety by empowering individuals with information to make informed decisions during flood events. Additionally, it enhances the coordination of emergency response teams, ensuring a more efficient allocation of resources.

### Future Considerations:

1. Scale and Coverage: This project can be further expanded to cover larger geographical areas and incorporate a broader network of sensors. This would provide even more comprehensive flood monitoring capabilities.

2. Integration with Weather Forecasting: Incorporating real-time weather forecasting data can enhance the system's predictive accuracy, allowing for more advanced warnings and better preparation.

3. Mobile Application: Developing a mobile application would provide residents with a more accessible and user-friendly platform for receiving flood warnings and accessing real-time data.

4. Community Engagement and Education: Raising awareness about flood risks and educating the community on how to respond to warnings is crucial in maximizing the effectiveness of the system.

5. Resilience and Sustainability: Implementing backup power sources and ensuring the durability of sensors in harsh environmental conditions will contribute to the long-term sustainability of the system.

In conclusion, the flood monitoring and early warning system based on IoT represents a significant advancement in flood preparedness and response. By leveraging technology and data-driven approaches, we are better equipped to mitigate the impact of floods on communities and improve overall public safety. This project serves as a foundation for further innovation in disaster management and environmental monitoring.