CREATING A FLOOD MONITORING AND EARLY WARNING SYSTEM.

HARDWARE REQUIRED:

- NODE MCU
- 16*2 LCD DISPLAY (12 CM MODULE)
- CSM 900A MODULE
- ULTRASONIC SENSOR
- FLOAT SENSOR
- ZERO PCB
- 5V 2A DC POWER SUPPLY

STEP 01:

1. Visit the Wok Wi website (https://wokwi.com/).
2. Create an account if you don't have one already.
3. Once logged in, click on "Create a new simulation”.
4. Select "ZERO PCB" as your board.

STEP 02:

(Connecting the Hardware in Wok Wi)

1. In the Wok Wi simulator, you can add components like the ZERO PCB, ULTRASONIC SENSOR and FLOAT SENSOR by dragging them from the components panel onto the virtual breadboard.
2. Connect the components using virtual jumper wires. Connect the power and ground pins appropriately.
3. Connect the Ultra Sonic Sensor to an analog input pin on the ZERO PCB.
4. Connect the 5V 2A DC POWER SUPPLY to a digital output pin on the ZERO PCB

STEP 03:

(Writing the Arduino Code)

Here's a simple Arduino code example to get you started with flood monitoring and early warning. This code reads the water level from the sensor and triggers the alarm when the water level exceeds a certain threshold.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16 column and 2 rows

const int trigPin = D1; // Ultrasonic sensor trigger pin
const int echoPin = D2; // Ultrasonic sensor echo pin
const int floatSensorPin = D3; // Float sensor pin
const int alarmPin = D4; // Alarm pin
const int waterThreshold = 30; // Water level threshold in centimeters

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(floatSensorPin, INPUT);
  pinMode(alarmPin, OUTPUT);

  lcd.begin(16, 2);
  lcd.print("Flood Monitoring");
}

void loop() {
```

```
// Read water level from ultrasonic sensor
long duration;
float distance;

digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2;

// Read float sensor status
int floatSensorValue = digitalRead(floatSensorPin);

// Display water level on LCD
lcd.setCursor(0, 1);
lcd.print("Water Level: ");
lcd.print(distance);
lcd.print(" cm");

// Check water level and trigger alarm if it exceeds threshold
if (distance <= waterThreshold || floatSensorValue == HIGH) {
  digitalWrite(alarmPin, HIGH); // Turn on alarm
  lcd.setCursor(0, 0);
```

```
    lcd.print("ALERT: Flooding!");
  } else {
    digitalWrite(alarmPin, LOW); // Turn off alarm
    lcd.setCursor(0, 0);
    lcd.print("Flood Monitoring");
  }

  delay(1000); // Delay for stability
}
```

In this code:

- The ultrasonic sensor measures the water level and calculates the distance in centimetres.
- The float sensor checks if the float is lifted (indicating water level) and triggers an alert.
- If the water level exceeds the specified threshold or the float sensor is triggered, an alarm is triggered, and the LCD displays an alert message.
- The LCD displays the current water level continuously.

Upload this code to your CSM 900A MODULE in the Wok Wi simulator.

Step 04:

(Simulate the Flood Monitoring)

1. In the Wok Wi simulator, click on the "Start Simulation" button to run your project.

2. Simulate the water level sensor by clicking on it and changing its value to simulate rising water levels.

3. Observe how the 5V 2A DC POWER SUPPLY activates when the water level exceeds the threshold

STEP 05:

(Features Of Flood Monitoring and Early Warning)

1. Timely Alerts and Early Warning:

- One of the most critical aspects of flood monitoring systems is their ability to provide timely alerts and early warnings to communities at risk. By detecting rising water levels and extreme weather patterns in real-time, these systems can issue warnings promptly, allowing residents to evacuate and take necessary precautions before the situation escalates.

2. Data-Driven Decision Making:

- Flood monitoring systems rely on accurate data collected from various sensors. Analysing this data helps authorities make informed decisions. By understanding historical patterns and current conditions, officials can deploy resources effectively, plan evacuations, and allocate emergency funds where they are most needed.

3. Community Engagement and Education:

- Engaging communities in the monitoring process is crucial. Educating the public about flood risks, safety protocols, and evacuation routes enhances their preparedness. Informed communities are better equipped to respond to warnings, follow evacuation plans, and assist others during emergencies, reducing the overall impact of floods.

4. Integration of Technology:

- Modern flood monitoring systems integrate advanced technologies such as Geographic Information Systems (GIS), remote sensing, and data analytics. These technologies provide comprehensive insights into flood-prone areas, enabling precise mapping of risks. Integrating weather forecasts into the system enhances predictive capabilities, allowing for proactive measures before a flood occurs.

5. Continuous Improvement and Adaptability:

- Flood monitoring systems must undergo continuous improvement based on feedback, post-event analysis, and advancements in technology. Regular assessments help identify weaknesses and areas for enhancement. Additionally, these systems need to be adaptable to changing climate patterns, ensuring they remain effective in the face of evolving environmental challenges.

STEP 06:

(Description Of Flood Monitoring and Early Warning)

Flood monitoring system uses NODE MCU, 16x2 LCD, CSM 900A module, ultrasonic sensor, float sensor, and ZERO PCB. It detects water levels via sensors and displays warnings on the LCD if levels rise, ensuring timely alerts for potential floods. Powered by a 5V 2A DC supply, it aids in early flood detection.