

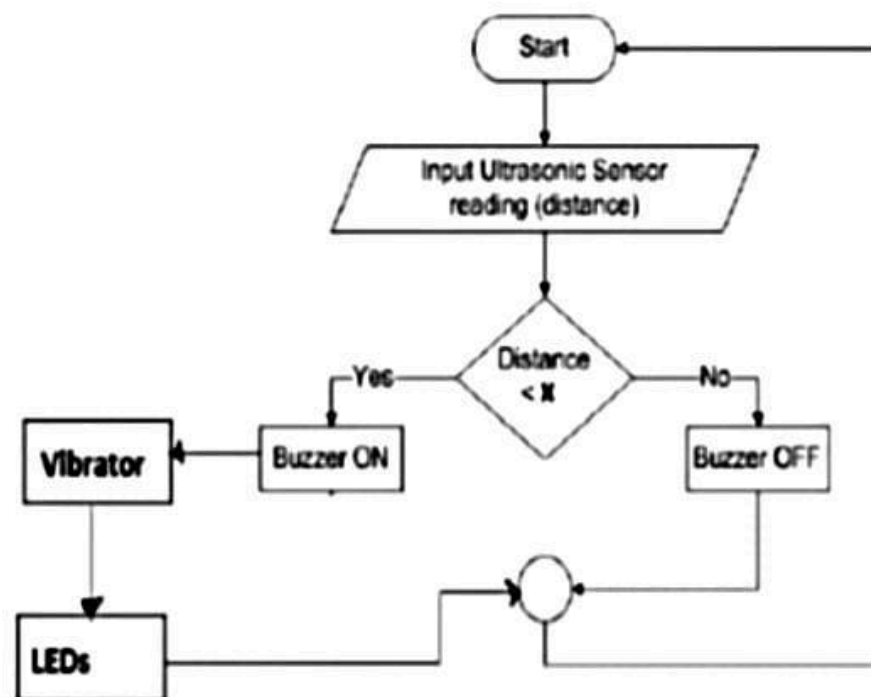
FLOOD MONITORING AND EARLY WARNING SYSTEM

- **NAME : P. UMA MAHESWARI**
- **COLLEGE CODE : 9530**
- **COLLEGE NAME : ST .MOTHER THERESA ENGINEERING COLLEGE**
- **GITHUB LINK : <https://github.com/Umamaheswari-74/lot2/tree/5a023f574d3e8b99dc4fb14ab01e55d31445a388>**

PROBLEM SOLUTION

Floods are the natural disasters that cause catastrophic destruction and devastation of natural life , agriculture ,property and infrastructure every year .The objective of this project is to monitor the flood situation and send alert in case of danger in the form of text message .The main objective of this project is to detect rising water level in a river at a reasonable distance from the rail track/roadways and intimate that to the respective authorities through SMS ,to take appropriate action .Our project solves problem by implementing an early flood detection mechanism .In this project we will connect water level sensors at different water levels.

METHODOLOGY USED WITH FLOWCHART



| S.N | Component | Quantity |
|-----|---------------------------------|----------|
| 1 | ESP8266 | 1 |
| 2 | GSM module (SIM800I) | 1 |
| 3 | Ultrasonic Sensor | 1 |
| 4 | Float Sensor | 1 |
| 5 | 16x2 LCD Display With I2C | 1 |
| 6 | Zero PCB | 1 |
| 7 | 5v Power Supply | 1 |

Specifications of Float Sensors

- Cable Length: 30.(cm)
- Maximum Load: 51 W
- Max Switching Voltage: 90V DC
- Minimum Voltage: 240V DC
- Maximum Switching Current: 0.6 A
- Max Load Current: 1.0 A

Ultrasonic Sensor

- Ultrasonic sensors find out the distance of the water level of the dam. And the Sensor mount on the top of the dam.
- Ultrasonic Sensor required a 5v power supply.

```
1 // Ultrasonic Sensor
2 #define TRIGGER_PIN 12
3 #define ECHO_PIN 11
4 #define MAX_DISTANCE 200
5 NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
6
7 #define FLOAT_SENSOR_PIN 10
8 #define FLOOD_THRESHOLD 50
```

GSM module (SIM8001)

GSM SIM800L is a popular module that enables communication over GSM (Global System for Mobile Communications) networks. It Sends a text message and calls to the particular Mobile Number. and is necessary to put a valid sim card in the gsm module.

16x2 LCD Display

- This is a basic 16-character by 2 lines Alphanumeric display. Black text on Green background. Utilises the extremely common HD44780 parallel interface chipset. Interface code is freely available.

ESP8266

ESP8266 is a Wi-Fi module developed by Espressif Microcontroller Systems. In the board is a microcontroller unit And a built-in Wi-Fi Chip, It is the low-cost solution for Wi-Fi connectivity to various projects.

```
1 void loop() {
2   // Read Ultrasonic Sensor
3   unsigned int distance = sonar.ping_cm();
4
5   // Read Float Sensor
6   int floatSensorValue = digitalRead(FLOAT_SENSOR_PIN);
7
8   // Calculate Flood Level
9   int floodLevel = distance;
10
11  // Update LCD Display
12  lcd.clear();
13  lcd.setCursor(0, 0);
14  lcd.print("Water Level: ");
15  lcd.print(floodLevel);
16  lcd.print("cm");
17
18  // Check Flood Threshold
19  if (floodLevel > FLOOD_THRESHOLD && floatSensorValue == HIGH) {
20    // Send Alert SMS
21    sendAlertSMS(floodLevel);
22  }
23
24  delay(500); // Delay for stability
25 }
```

```
19  if (floodLevel > FLOOD_THRESHOLD && floatSensorValue != 0)
20      // Send Alert SMS
21      sendAlertSMS(floodLevel);
22  }
23
24  delay(500); // Delay for stability
25 }
```

```
1 void setup() {
2     // Initialize LCD Display
3     lcd.begin(16, 2);
4     lcd.backlight();
5
6     // Initialize GSM Module
7     gsmSerial.begin(GSM_BAUDRATE);
8     delay(2000);           // Give GSM module time to
9     sendCommand("AT");     // Check communication
10    sendCommand("AT+CMGF=1"); // Set SMS text mode
11
12    // Display Initialization Message
13    lcd.clear();
14    lcd.setCursor(0, 0);
15    lcd.print("Flood Monitoring");
16    lcd.setCursor(0, 1);
17    lcd.print("System");
18
19    delay(3000); // Display initialization message for 3 s
20 }
```

```
1 void sendCommand(String command) {  
2     gsmSerial.println(command);  
3     delay(1000);  
4     while (gsmSerial.available()) {  
5         gsmSerial.read();  
6     }  
7 }
```

```
1 void sendAlertSMS(int floodLevel) {  
2     String message = "Flood Alert! Water level is ";  
3     message += floodLevel;  
4     message += "cm. Take necessary actions.";  
5  
6     for (int i = 0; i < sizeof(phoneNumbers) / sizeof(phone  
7         sendCommand("AT+CMGS=\"" + phoneNumbers[i] + "\"");  
8         delay(1000);  
9         sendCommand(message);  
10        delay(100);  
11        sendCommand((String) char(26));  
12        delay(1000);  
13    }
```

Circuit Diagram

