# Flood Monitoring and Early Warning

Define Objectives and Scope:

Clearly define the goals of your flood monitoring system. Determine the geographic area you want to cover and the level of flood severity you aim to detect.

Sensor Selection:

Choose appropriate sensors to monitor relevant environmental parameters such as water level, rainfall, temperature, and humidity. Popular choices include ultrasonic water level sensors, rain gauges, and weather stations.

Data Collection and Processing:

Install the sensors in strategic locations within the target area.
Set up a data collection infrastructure using IoT devices and platforms. These devices will transmit sensor data to a central server or cloud platform for processing.
Implement data analytics and machine learning algorithms to process the incoming data in real-time. These algorithms can detect abnormal patterns, predict potential floods, and trigger alerts.

Communication Infrastructure:

Use wireless communication technologies like Wi-Fi, LoRaWAN, or cellular networks to transmit data from sensors to the central server or cloud platform.
Ensure redundancy and reliability in data transmission to prevent communication failures during critical situations.

Data Visualization:

Create a user-friendly dashboard or mobile application for stakeholders, including local authorities and residents, to access real-time data, flood forecasts, and warnings.
Include interactive maps, charts, and alerts to convey information effectively.

Early Warning System:

Set up automated alerting mechanisms based on predefined thresholds and flood prediction models. These alerts can be sent via SMS, email, mobile apps, or sirens to notify authorities and residents.
Develop a tiered warning system that differentiates between flood severity levels to facilitate appropriate responses.

Community Engagement:

Educate the local community about the system's capabilities, how to interpret alerts, and emergency response procedures.
Encourage residents to report local observations, which can complement sensor data.

## Design into innovation:

The Proposed system consists of Rain Sensor, Ultrasonic Sensor, Power Source, NodeMCU ESP8266, Buzzer and LEDs and finally Blank App. This Wireless Sensor Node is Kept in desired location Like dam, Bridge. etc. and Blynk App is downloaded by victims near the flooding area. The Schematic Diagram of the proposed flood Monitoring System.

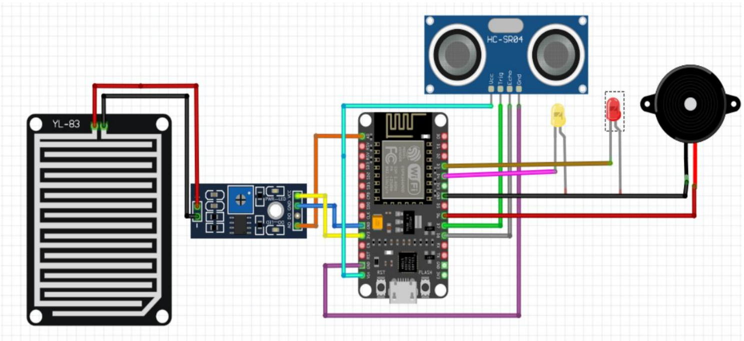
The proposed system will help in predicting the flood with the factors of water level change and rain fall intensity. The wireless sensor node consists of rain sensor which is used to measure the rain intensity. When there is no rain fall the intensity is “0” and if rain fall starts the intensity starts increasing and the speed the rain fall the intensity increases.

The intensity is sent to NodeMCU which checks the value which is defined in code. the checked value with threshold. if more than threshold then there will be a alert to victims as flood is predicted.

Even when the water level changes based on given threshold values it checks and send alerts. The warning is in three stages Safe level, Warning Level, Critical Level. The alert can of 2 types one way is send notification to victims and other one way is buzzer sound.

For Sending Notifications The victims should have Blynk App in Mobile phone. The LEDs are also displayed according to the water level, green, yellow, red. The connections of devices are done using jumper wires.

The code is Written is Arduino ide and dumped into NodeMCU. The sending of information from NodeMCU to Blynk app is with the help of Esp 8266. The ESP8266 is Wi-Fi module which is in built in NodeMCU.



Source code:

```
#include <ESP8266WiFi.h>
#include <Wire.h>

const char* ssid = "YourWiFiSSID";
const char* password = "YourWiFiPassword";

const int rainPin = D1;  // Pin for the rain sensor
const int ultrasonicTrigPin = D2;  // Trigger pin for ultrasonic sensor
const int ultrasonicEchoPin = D3;  // Echo pin for ultrasonic sensor
const int buzzerPin = D4;
const int ledPin = D5;

bool isRaining = false;

void setup() {
  Serial.begin(115200);
  pinMode(rainPin, INPUT);
  pinMode(ultrasonicTrigPin, OUTPUT);
  pinMode(ultrasonicEchoPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
}

void loop() {
  // Check the rain sensor
  int rainValue = digitalRead(rainPin);
  if (rainValue == HIGH) {
    if (!isRaining) {
      isRaining = true;
      // It's raining, do something
      digitalWrite(ledPin, HIGH);  // Turn on an LED
      tone(buzzerPin, 1000);  // Activate the buzzer
    }
  } else {
    isRaining = false;
    digitalWrite(ledPin, LOW);  // Turn off the LED
    noTone(buzzerPin);  // Turn off the buzzer
  }

  // Read the ultrasonic sensor
  // You can use this data to determine water level or other applications

  delay(1000);  // Wait for a second
}
```