Phase 3 Development part 1

FLOOD MONITORING SYSTEM

# DESCRIPTION:

This document is describing the process of developing the project with the mentioned components in phase 2 innovation segment. For our project we are going to use ultrasonic sensor and optical rain sensor and gps sensor

# COMPONENTS REQUIRED:

# Ultrasonic sensor.

# Optical rain sensor.

# GPS sensor.

# Arduino IDE.

# Power Supply.

# Basic components like Bread board, Wires.

# WORKING DESCRIPTION:

# A flood monitoring system using ultrasonic sensors, GPS sensors, and optical rain sensors is designed to detect and monitor potential flooding events in real-time. Here's a working description of how this system functions:

# The Arduino Board is chosen for the simulation available in the wowki platform. Then Ultrasonic Sensor,Optical Rain Sensor and GPS sensor is connected to the microcontroller with the corresponding pins and checked for installation. The LED is placed in a circuit for the indication of changes received from the sensor.Then the required wiring can be done and it will given to the power supply as the source of the elements. Then the circuit is checked for operation.

# The Arduino continuously reads data from the ultrasonic sensor, rain sensor, and GPS module in the loop() function.Ultrasonic Sensor will measures the distance to an water surface. If the measured distance is less than a predefined threshold , it indicates a flood.

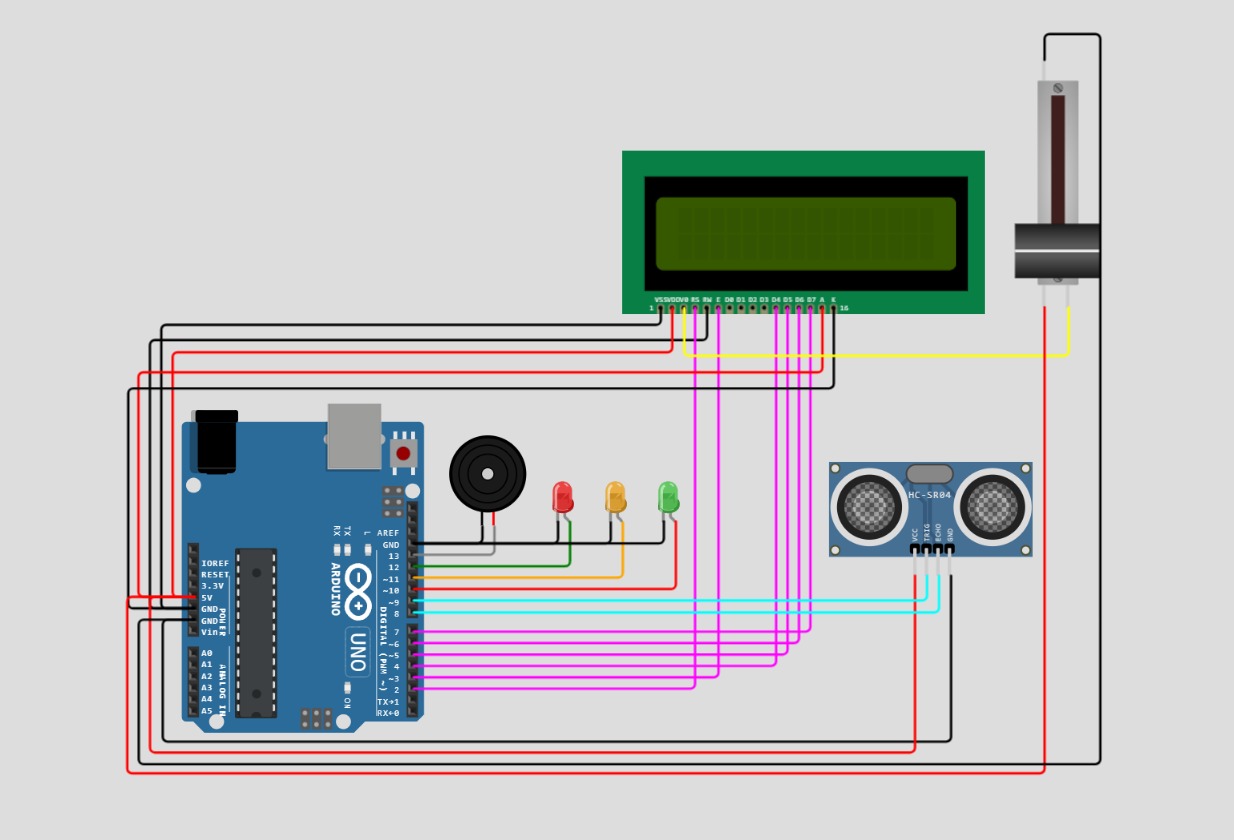
# Optical Rain Sensor reads the analog output to detect rain. If the analog value is higher than threshold rain value, it indicates rain.GPS Module reads latitude and longitude information. When a valid GPS fix is obtained, it means the module has accurate location data.

# The program code was written for the operation of the sensor in the Arduino Board for reading the values.The code will checks if rain or a flood is detected based on sensor readings.

# 

# If a flood is detected then it will be sent the data to the cloud using communication protocols and it is made available for the public using our concept of creating an API with the access to the stored data in the cloud

# CIRCUIT DIAGRAM:



**Program:**

#include <SoftwareSerial.h>

#include <TinyGPS++.h>

SoftwareSerial gpsSerial(4, 3); // RX, TX for GPS

TinyGPSPlus gps;

const int trigPin = 7; // Ultrasonic sensor trigger pin

const int echoPin = 6; // Ultrasonic sensor echo pin

const int rainSensorPin = A0; // Optical rain sensor analog pin

void setup() {

Serial.begin(9600);

gpsSerial.begin(9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

}

void loop() {

// Read GPS data

while (gpsSerial.available() > 0) {

if (gps.encode(gpsSerial.read())) {

// Retrieve GPS information

float latitude = gps.location.lat();

float longitude = gps.location.lng();

Serial.print("Latitude: ");

Serial.print(latitude, 6);

Serial.print(" Longitude: ");

Serial.println(longitude, 6);

}

}

// Read ultrasonic sensor data

long duration;

float distance;

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration / 2) / 29.1; // Speed of sound in air is 343 m/s

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

// Read rain sensor data

int rainValue = analogRead(rainSensorPin);

Serial.print("Rain Sensor Value: ");

Serial.println(rainValue);

// Add your flood monitoring logic here

// You can compare the distance and rain sensor values to set flood thresholds

delay(1000); // Adjust the delay as needed

}