# Flood monitoring and early warning system

Name: Sri Jaya Jothi M

Reg no: 952021106006

Team: Bhanusri G Brindha R A

Divya R

Objective:

* Flood Monitoring and Early warning to the people about Flood occurring.
* This can be done using ULTRASONIC

Sensors.

* There are simple Audrino code which

reads the water level from the sensors

and triggers the alarm.

* Use the ESP32 DevKit in Wokwi web-

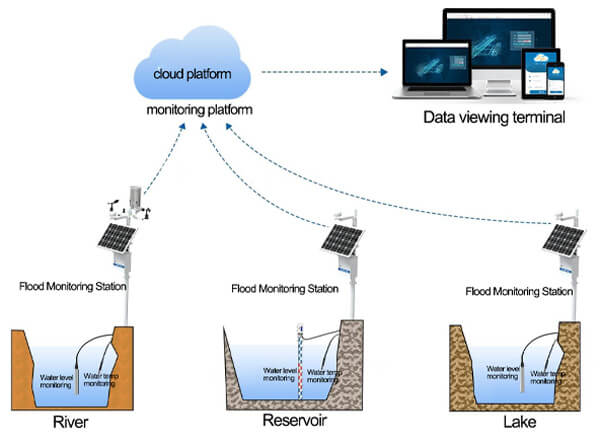
site to simulate the flood monitoring

code.

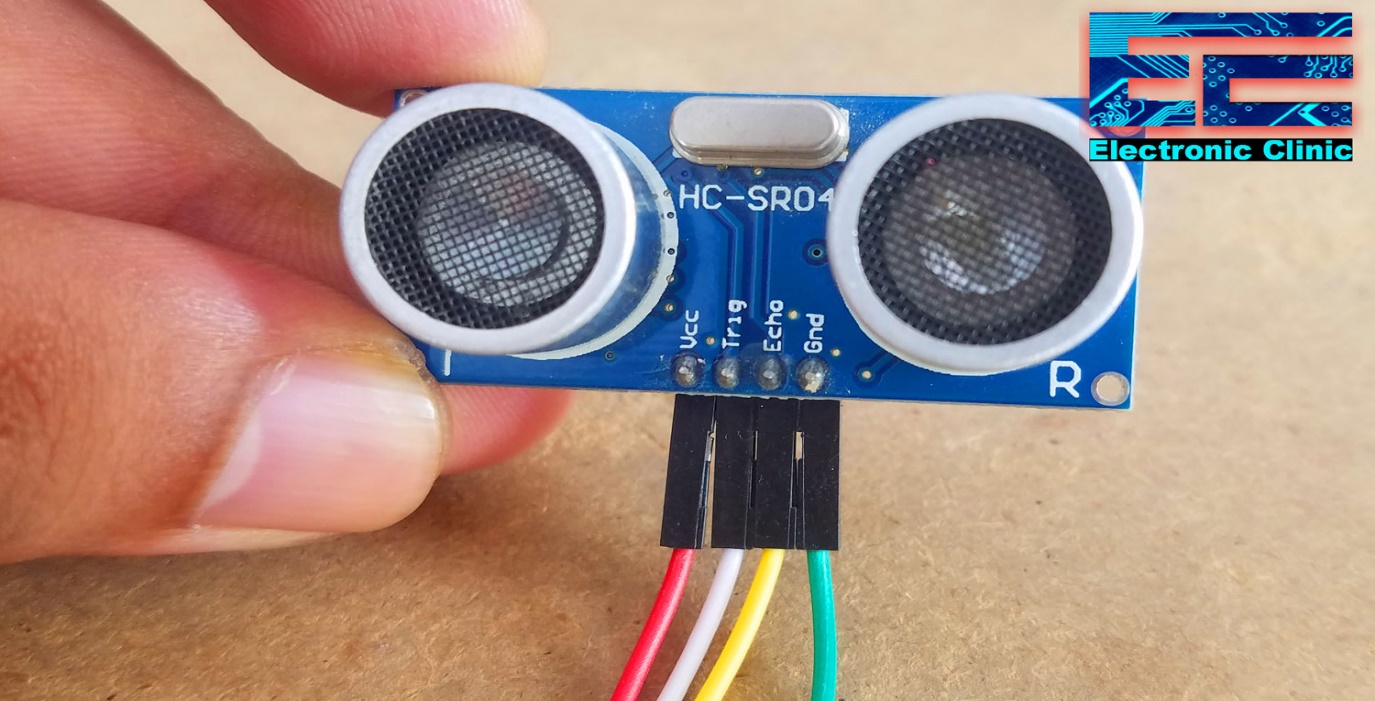
* This may give a correct warning so that we can avoid the death of the

fisherman by alerting them.

Flood Monitoring:



Ultrasonic sensor:



My Idea for solving this problem…

* To monitor the flood

we are choosing the ultrasonic sensors and

float sensors in the system.

* For early warning the ESP32 Dev Kit is used for the alarm setting process.
* Simulating the ESP32 kit using adurino coding

that I have already created for the alarm process.

Adurino Coding for Simulation…

#include <LiquidCrystal.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);  // Create an instance of the LiquidCrystal library

const int in = 8;                         // This is the ECHO pin of The Ultrasonic sensor HC-SR04

const int out = 9;                        // This is the TRIG pin of the ultrasonic Sensor HC-SR04

// Define pin numbers for various components

const int green = 10;

const int orange = 11;

const int red = 12;

const int buzz = 13;

void setup()

{

  // Start serial communication with a baud rate of 9600

**Serial**.begin(9600);

  // Initialize the LCD with 16 columns and 2 rows

  lcd.begin(16, 2);

  // Set pin modes for various components

  pinMode(in, INPUT);

  pinMode(out, OUTPUT);

  pinMode(green, OUTPUT);

  pinMode(orange, OUTPUT);

  pinMode(red, OUTPUT);

  pinMode(buzz, OUTPUT);

  // Display a startup message on the LCD

  lcd.setCursor(0, 0);

  lcd.print("Flood Monitoring");

  lcd.setCursor(0, 1);

  lcd.print("Alerting System");

  // Wait for 5 seconds and then clear the LCD

  delay(5000);

  lcd.clear();

}

void loop()

{

  // Read distance from the ultrasonic sensor (HC-SR04)

  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;

  // Map the distance value to a percentage value

  per = map(dist, 10.5, 2, 0, 100);

  // Ensure that the percentage value is within bounds

  if (per < 0)

  {

    per = 0;

  }

  if (per > 100)

  {

    per = 100;

  }

  // Print water level data to serial

**Serial**.print("Water Level:");

**Serial**.println(String(per));

  lcd.setCursor(0, 0);

  lcd.print("Water Level:");

  lcd.print(String(per));

  lcd.print("%  ");

  // Check water level and set alert levels

  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);

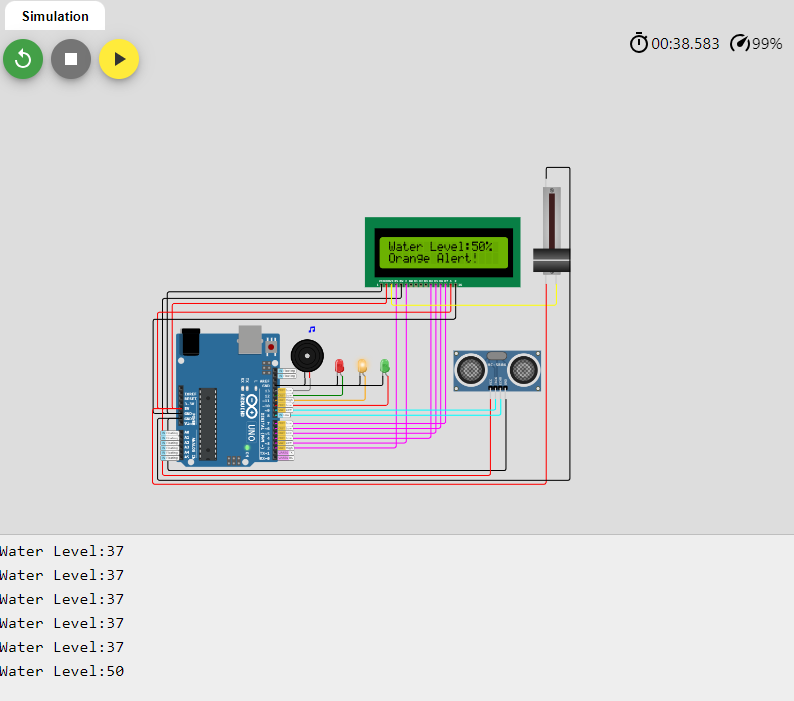
    digitalWrite(buzz, LOW);

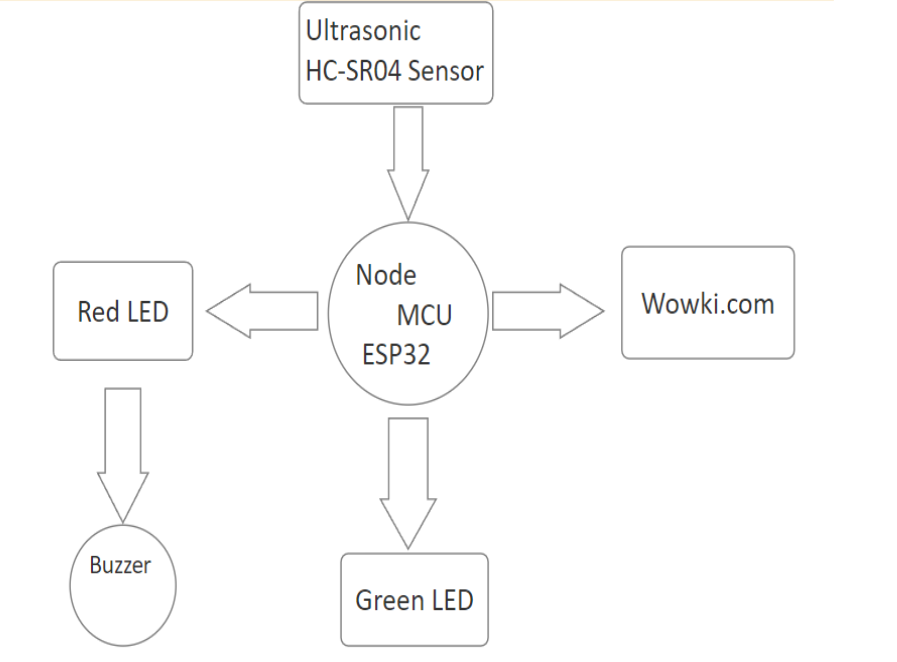
  }

}

This is the program for the simulation process and the output is given below…

Simulation process…



Block Diagram…

Block diagram is the sample of my project work.

These are the components that I am going use in my project.