# 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.

# Hardware Required:

1. Bolt-IoT wifi module
2. Arduino uno
3. Breadboard- 400 tie points
4. 5mm LED:(Green, Red, Orange) and Buzzer
5. 16×2 LCD Display
6. LM35 Temperature Sensor
7. HC-SR04 Ultrasonic Sensor
8. Some Jumper Wires
   1. Male to Female Jumper Wires- 15 pcs
   2. Male to Male Jumper Wires- 10 pcs
   3. Female to Female Jumper Wires- 5 pcs
9. 9v Battery and Snap Connector
10. USB Cable Type B

# Software Required:

1. [Arduino IDE](https://www.arduino.cc/en/software)
2. [Python 3.7 IDLE](https://www.python.org/downloads/)
3. [Bolt IoT Cloud](https://cloud.boltiot.com/)
4. [Bolt IoT Android App](https://play.google.com/store/apps/details?id=com.bolt.com.bolt)
5. [Twillo SMS Messaging API](https://www.twilio.com/)
6. [Mailgun EMAIL Messaging APISoftware components](https://www.mailgun.com/)

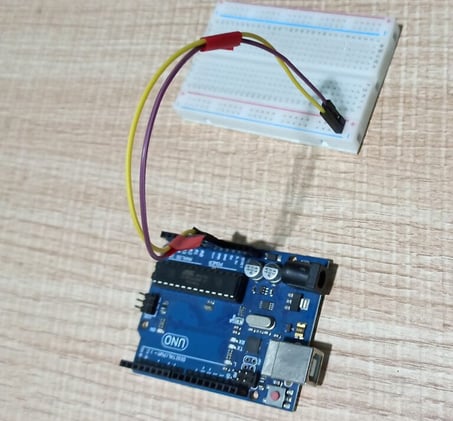
# Handtools and Fabrication Machines:

1. Electrical Tape
2. Green Cello Tape

# Hardware Setup:

For Building this project we first configure the hardware connections. Then later on moving to the software part.

**Step 1**: **Connecting 5v and GND of Arduino to the Breadboard for power connection to other components.**



**Step 2**: **Connecting LED’s**

**For Green LED:**

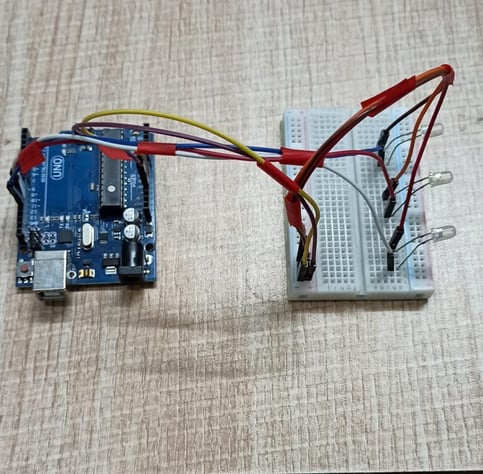
* VCC of  Green Colour LED to Digital Pin ‘10’ of the Arduino.
* GND of Green Colour LED to the GND of Arduino.

**For Orange LED:**

* VCC of  Orange Colour LED to Digital Pin ‘11’ of the Arduino.
* GND of Orange Colour LED to the GND of Arduino.

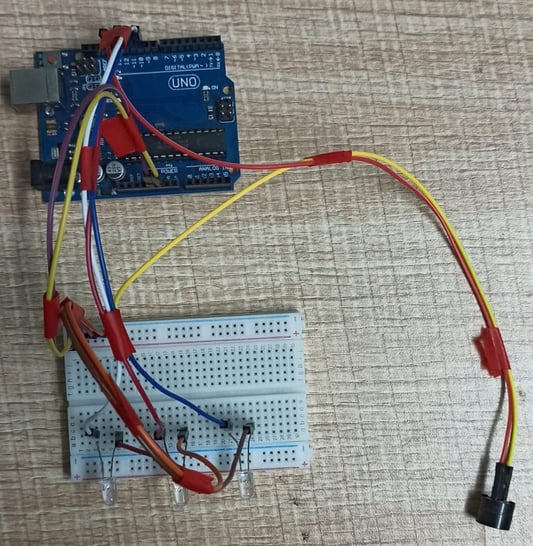
**For Red LED:**

* VCC of  Red Colour LED to Digital Pin ‘12’ of the Arduino.
* GND of Red Colour LED to the GND of Arduino.



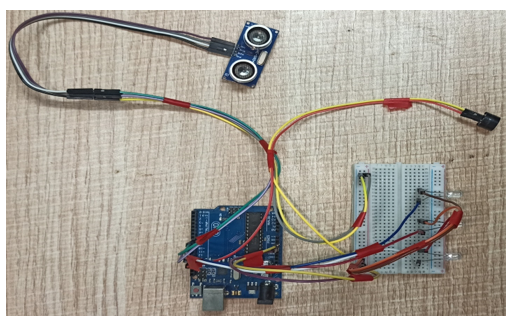
**Step 3**: **Connecting Buzzer**

* VCC of  Buzzer to Digital Pin ‘13’ of the Arduino.
* GND of Buzzer to the GND of Arduino.



**Step 4**: **Connecting HC-SR04 Ultrasonic Sensor**

* VCC of Ultrasonic Sensor to 5v of Arduino.
* GND of Ultrasonic Sensor to GND of Arduino.
* Echo of Ultrasonic Sensor to Digital Pin ‘8’ of Arduino.
* Trig of Ultrasonic Sensor to Digital Pin ‘9’ of Arduino.

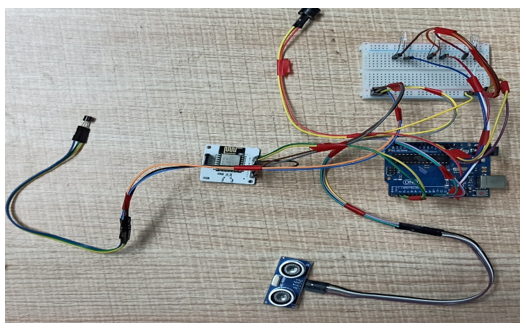


**Step 5: Connecting Bolt WiFi Module**

* 5v of Bolt WiFi Module to 5v of Arduino.
* GND of Bolt WiFi Module to GND of Arduino.
* TX of Bolt WiFi Module to RX of Arduino.
* RX of Bolt WiFi Module to TX of Arduino.

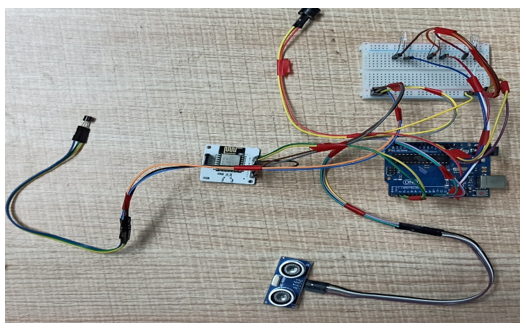
**Step 6: Connecting LM35 Temperature Sensor**

* VCC of LM35 to 5v of Bolt WiFi Module.
* Output Pin of LM35 to Pin ‘A0’ of  Bolt WiFi Module.
* GND of LM35 to GND of Bolt WiFi Module.

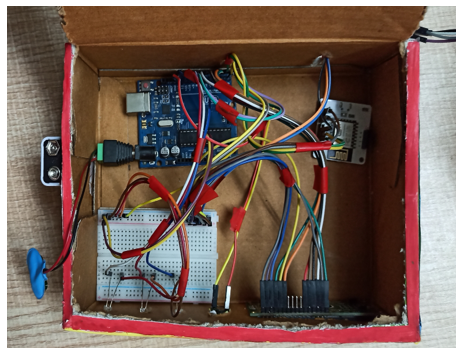


**Step 7:Connecting 16×2 LCD  Display**

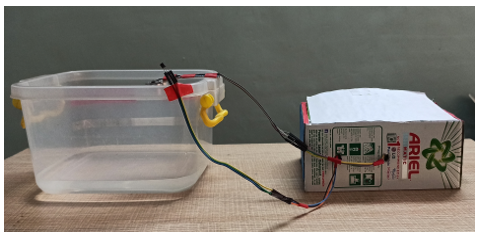
* Pin 1,3,5,16 of 16×2 LCD to GND of Arduino.
* Pin 2,15 of 16×2 LCD to 5v of Arduino.
* Pin 4 of 16×2 LCD to Digital Pin ‘2’ of Arduino.
* Pin 6 of 16×2 LCD to Digital Pin ‘3’ of Arduino.
* Pin 11 of 16×2 LCD to Digital Pin ‘4’ of Arduino.
* Pin 12 of 16×2 LCD to Digital Pin ‘5’ of Arduino.
* Pin 13 of 16×2 LCD to Digital Pin ‘6’ of Arduino.
* Pin 14 of 16×2 LCD to Digital Pin ‘7’ of Arduino.



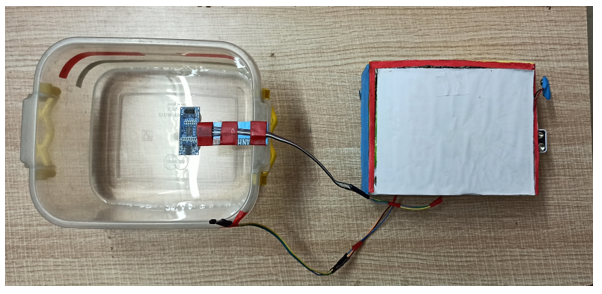
After doing the hardware connection put all the hardware components in one box.



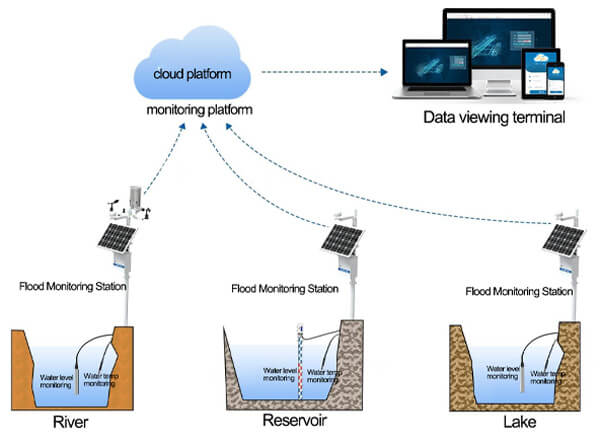
Also attach LM35 Temperature Sensor on the side of the container.



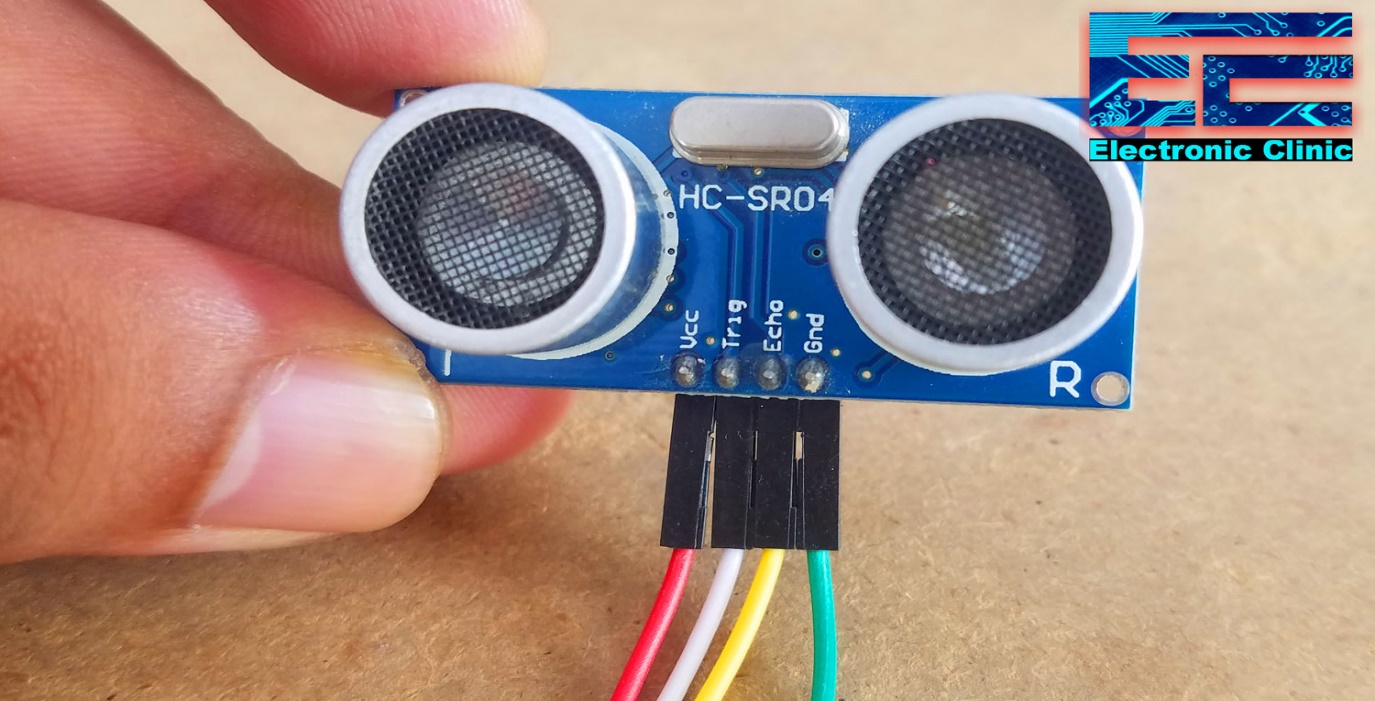
Also attach Ultrasonic sensor on the top of the container.



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);

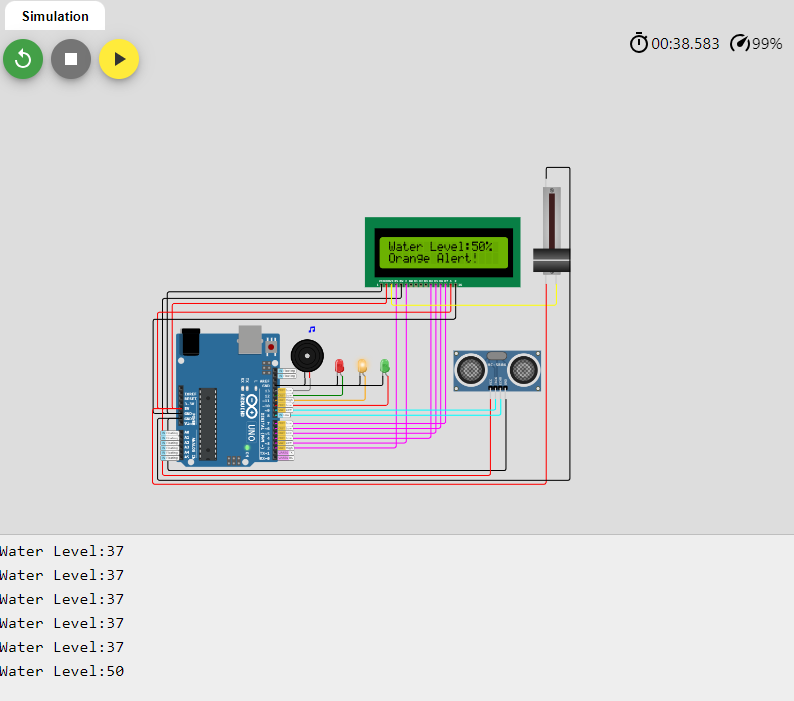
    digitalWrite(buzz, LOW);

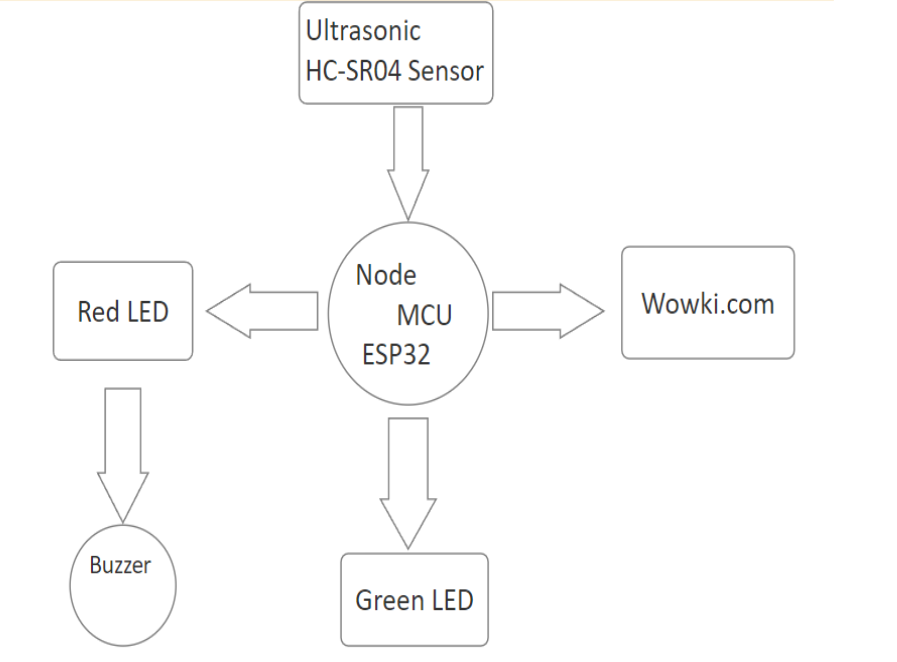
  }

}

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.

# Conclusion:

Nowadays the Internet Of things (IoT) is broadly used in worldwide, this system will display the data of the water level measured on lcd display. This project can be very helpful to the Meteorological Department to continuously monitor the dams and river beds water level. With this project it can save many people lives by giving alerts when the water level crosses beyond the limit. This project is very cost-effective, flexible and productive in areas where flood conditions happens everytime…