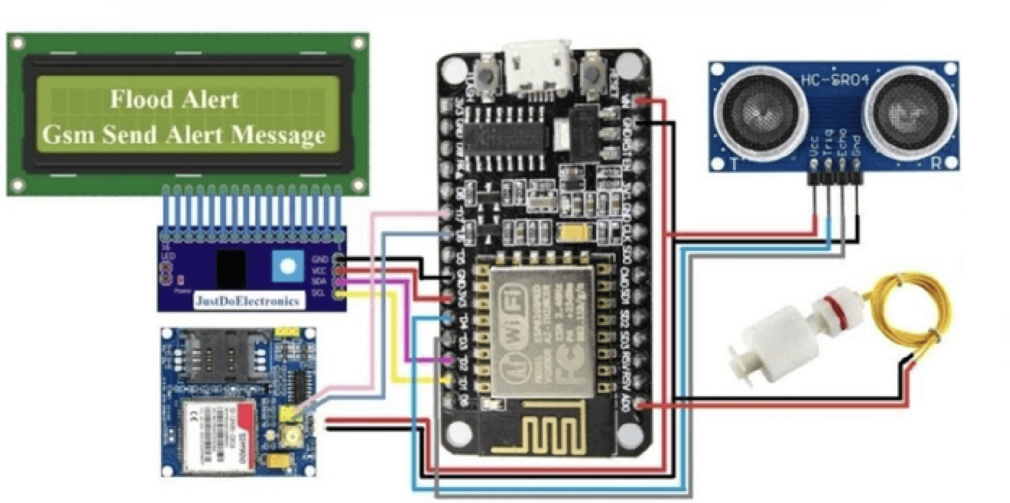
FLOOD MONITORING AND EARLY WARNING

Objectives:

Develop an advanced early flood detection system integrating data sources, assessing risk, and issuing timely warnings to communities. Use technology, community engagement, and response planning to minimize flood impact and save lives while ensuring sustainability and continuous improvement.

Sensor Setup:

In this project, we will explore the design and implementation of a Flood Monitoring System using various components such as an Arduino UNO, 16×2 LCD display with I2C, ultrasonic sensor, float sensor, and SIM800L module. This system aims to provide real-time flood level monitoring and alert notifications to receive Text Messages.



Mobile App Develop:

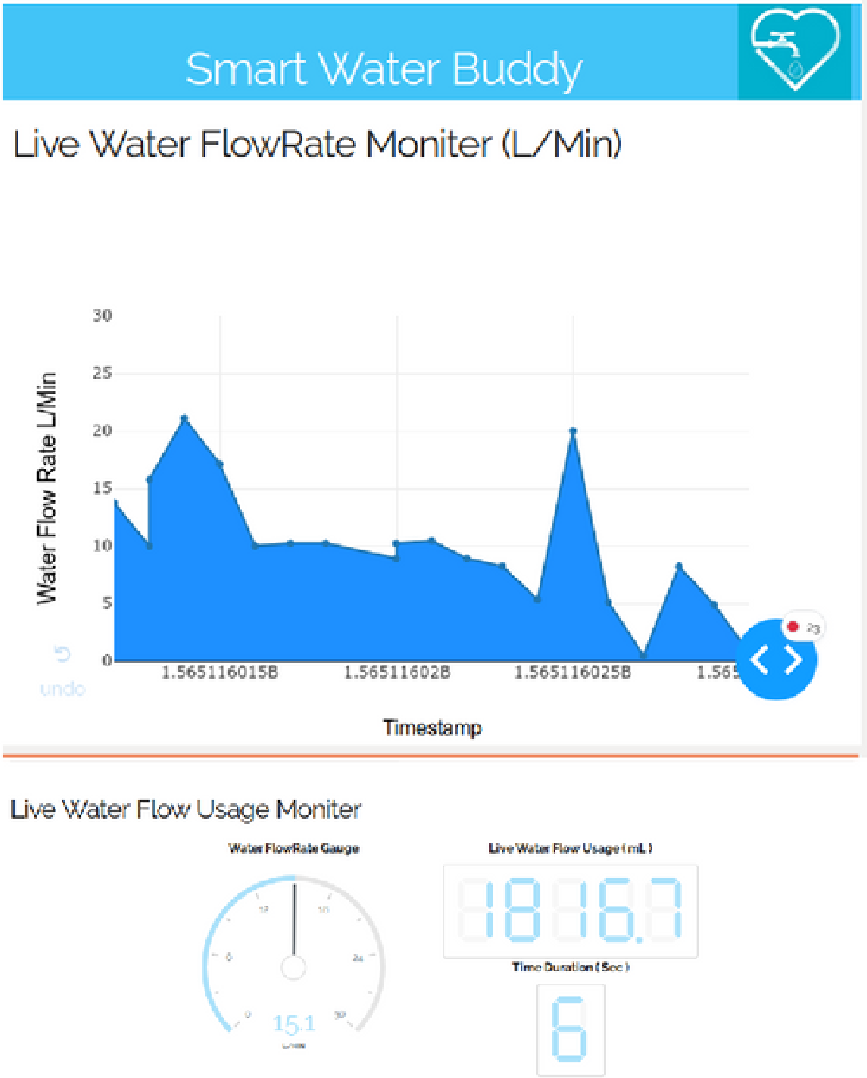
The mobile app is the user interface for the system. It allows users to:

View real-time water consumption data.
Set daily, weekly, or monthly water usage targets.
Receive notifications and alerts when approaching or exceeding these targets.
Access historical water consumption data in the form of graphs and reports.
The app can be developed for iOS and Android platforms using appropriate frameworks (e.g., React Native, Flutter).

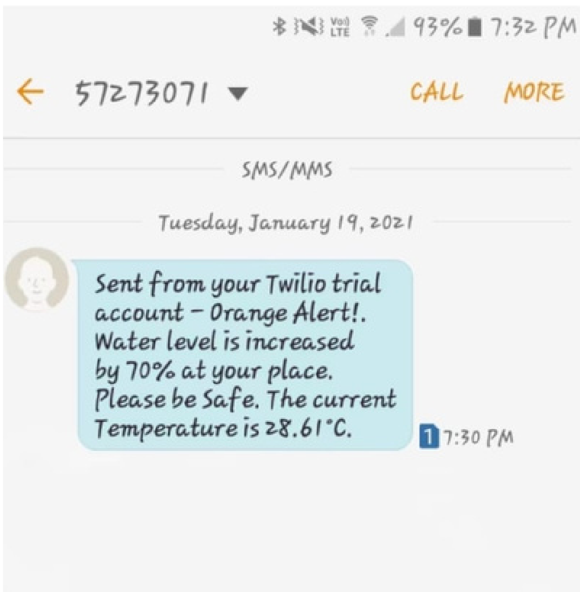
Code To Connect:

```
#define BLYNK_TEMPLATE_ID "TMPL3tobBFjj-"
#define BLYNK_TEMPLATE_NAME "IOT FLOOD
MONITORING"
#define BLYNK_AUTH_TOKEN "gy2bzR-i-
RbPW3oWOpAiDgr6sSVzlHVZ"

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <ESP32Servo.h>
Servo gate;
const int trigPin=2;//d2
const int echoPin=4;//d4
const int servoPin = 18;//d18
long duration;
int distance;
```

```
void setup() {
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
   pinMode(trigPin, OUTPUT);
   pinMode(echoPin, INPUT);
   gate.attach(servoPin, 500, 2400);
 }

 void loop()
 {
  digitalWrite(trigPin, LOW);
  delay(2);
  digitalWrite(trigPin,HIGH);
  delay(10);
  digitalWrite(trigPin, LOW);


  duration=pulseIn(echoPin,HIGH);
  distance=duration*0.034/2;
  Serial.println(distance);
  Blynk.virtualWrite(V0,distance);
if(distance<250)
{
  gate.write(90);
 Blynk.virtualWrite(V1,"FLOOD DETECTED GATES
OPENED");
}
else
{
  gate.write(0);
Blynk.virtualWrite(V1,"SAFE CONDITIONS GATES
CLOSED");
}
}
```


Smart Water Buddy
Live Water FlowRate Moniter (L/Min)
Water Flow Rate L/Min
Timestamp
Live Water Flow Usage Moniter
Water FlowRate Gauge
Live Water Flow Usage (mL)
1816.7
Time Duration (Sec)
6
15.1


89% 9:44 PM
57575791
CALL
MORE
SMS/MMS
Tuesday, January 19, 2021
Sent from your Twilio trial account - Red Alert!. Water level is increased by 90% at your place. Please Don't move out of the house. The Current Temperature is 26.56°C
9:27 PM


93% 7:32 PM
57273071
CALL
MORE
SMS/MMS
Tuesday, January 19, 2021
Sent from your Twilio trial account - Orange Alert!. Water level is increased by 70% at your place. Please be Safe. The current Temperature is 28.61°C.
7:30 PM

Explanation:

This code initializes the ESP32,16×2 LCD display with I2C, ultrasonic sensor, float sensor, SIM800L module and the Blynk communication

ESP32:

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.

16×2 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.

GSM module (SIM800l):

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.

Ultrasonic Sensor:

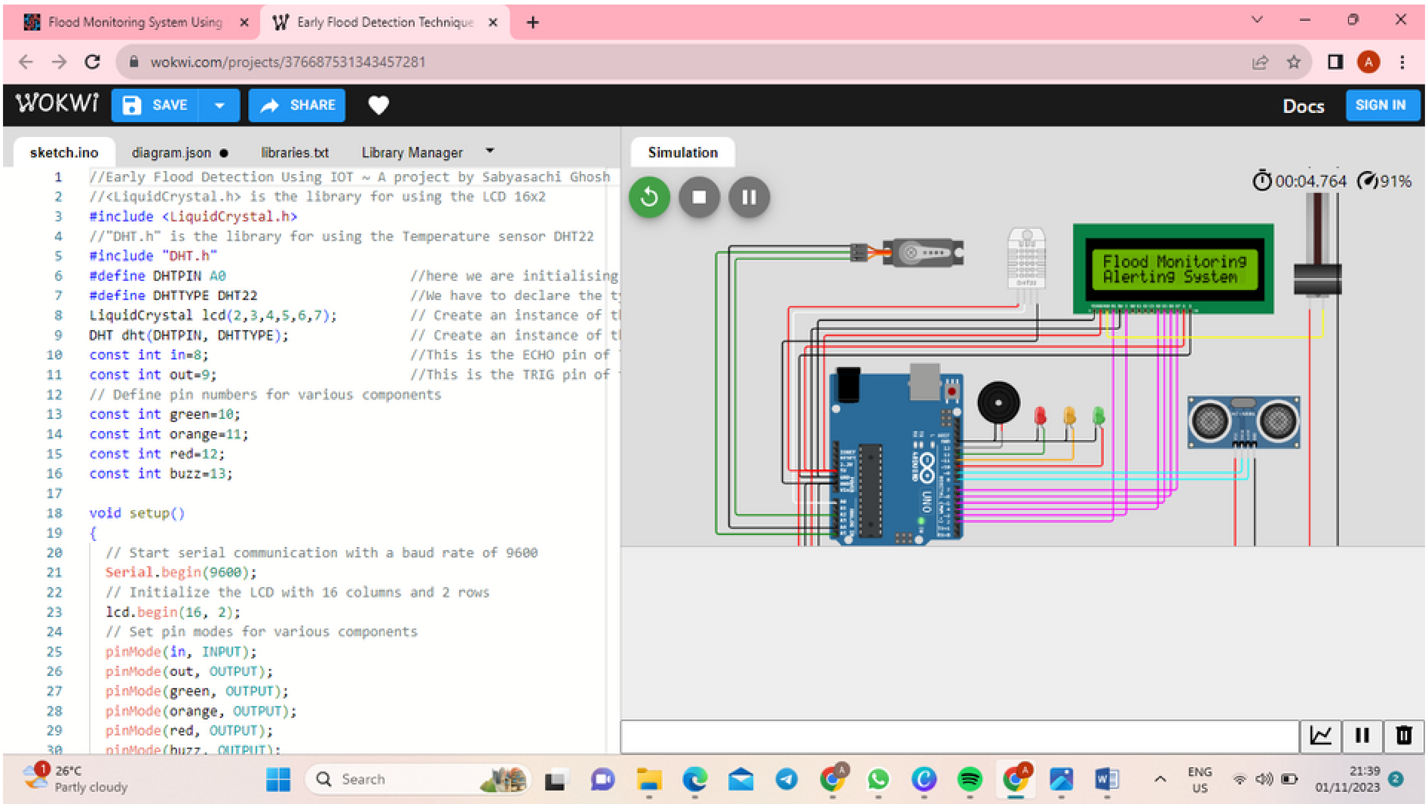
Ultrasonic sensors find out the distance of the water level of the dam. And the Sensor mount on the top of the dam. Ultrasonic Sensor required a 5v power supply.

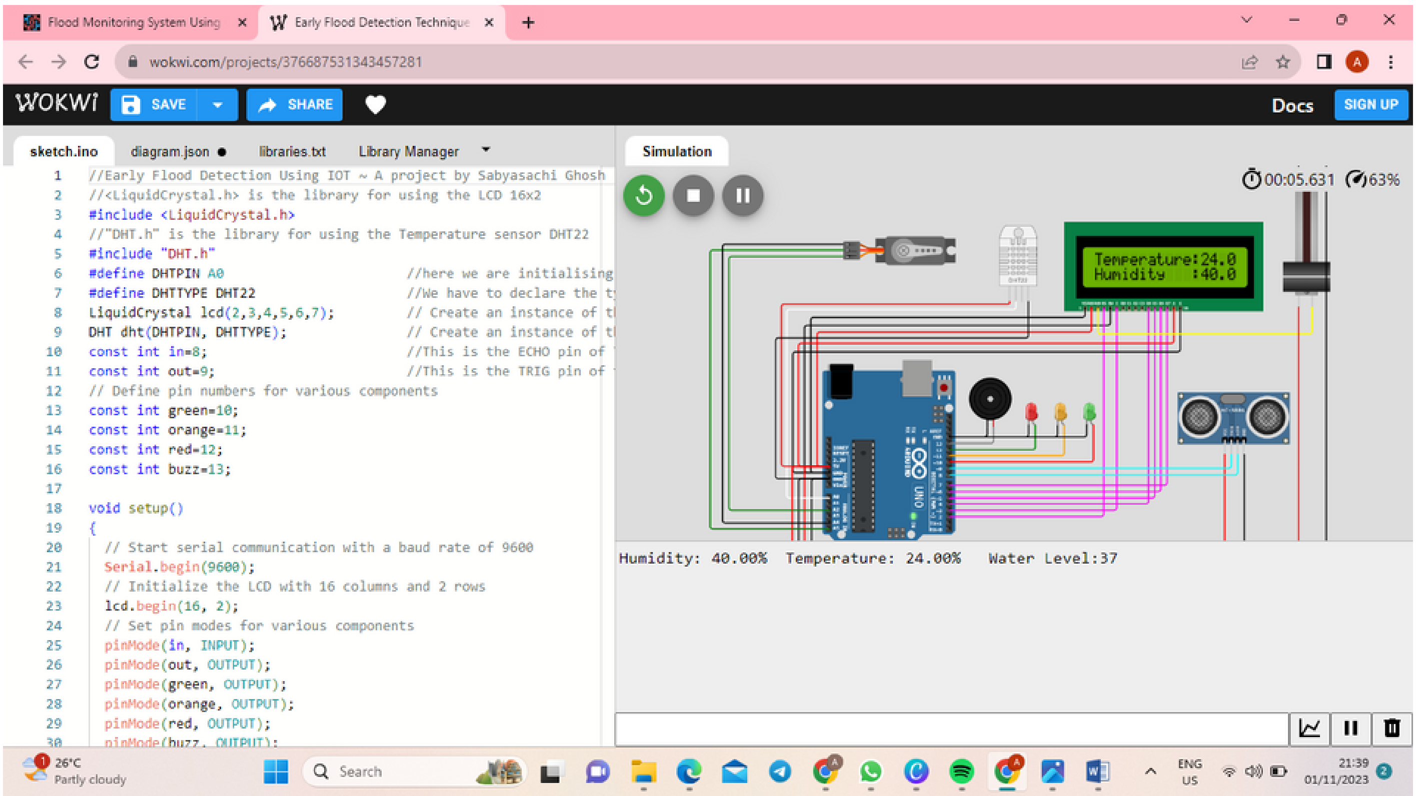
Float sensor:

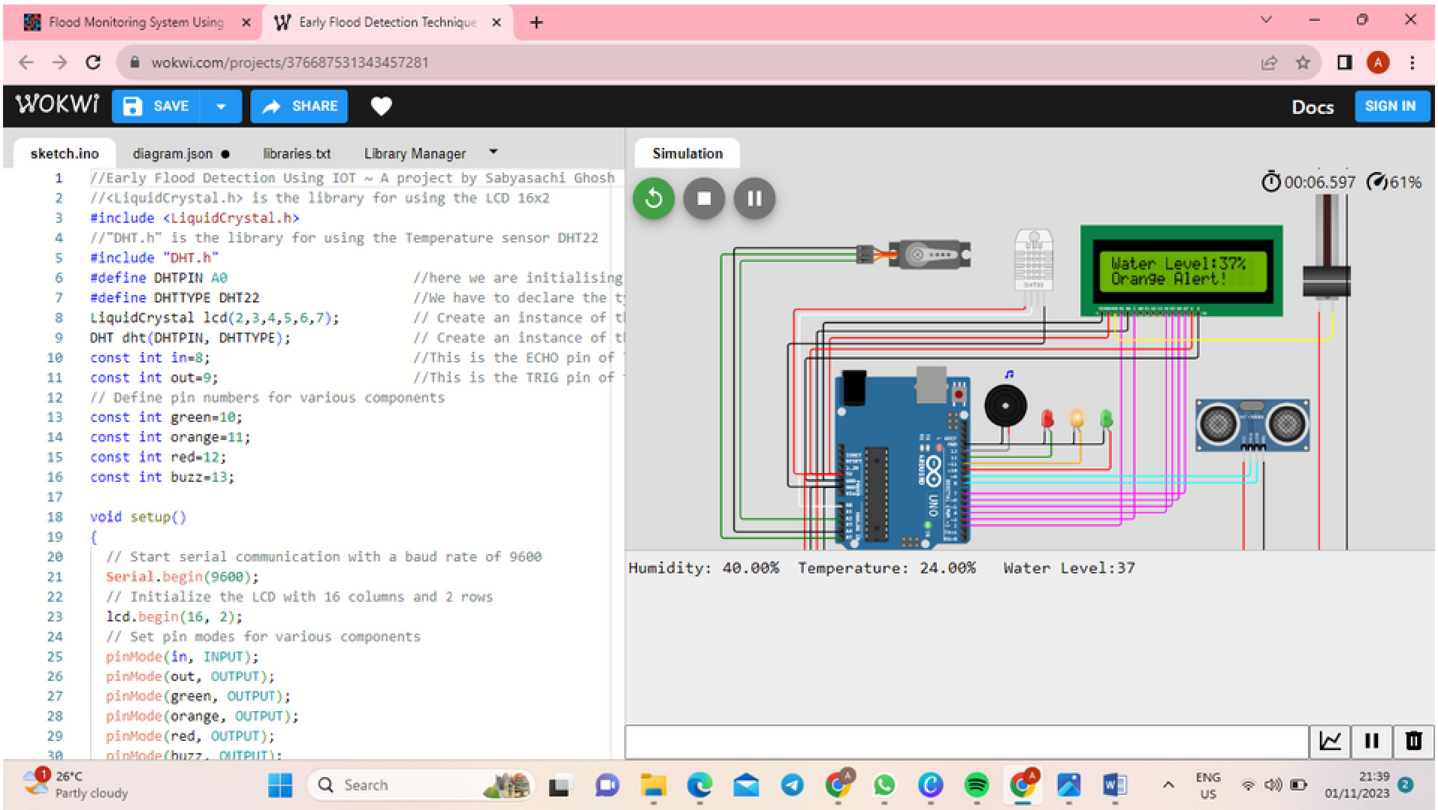
Float sensors to detect Water levels. They consist of a float, on the water and when the water level increases the float mechanism goes to the Top and is given the alert information.

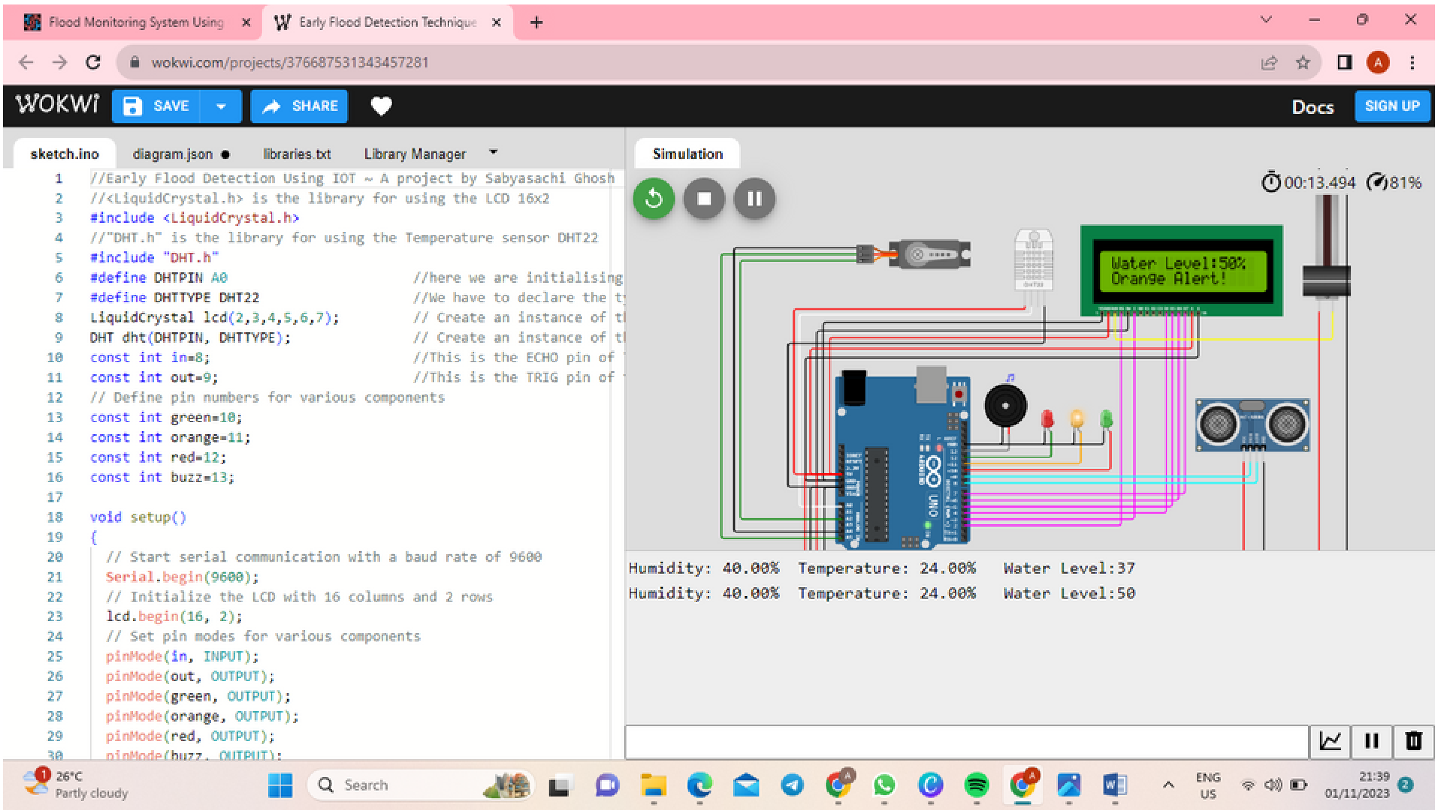
Specifications of Float Sensors

- 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


Flood Monitoring System Using
Early Flood Detection Technique
wokwi.com/projects/376687531343457281
WOKWI
SAVE
SHARE
Docs
SIGN IN
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:04.764
91%
Flood Monitoring
Alerting System
//Early Flood Detection Using IOT ~ A project by Sabyasachi Ghosh
//<LiquidCrystal.h> is the library for using the LCD 16x2
#include <LiquidCrystal.h>
//"DHT.h" is the library for using the Temperature sensor DHT22
#include "DHT.h"
#define DHTPIN A0 //here we are initialising
#define DHTTYPE DHT22 //We have to declare the t
LiquidCrystal lcd(2,3,4,5,6,7); // Create an instance of t
DHT dht(DHTPIN, DHTTYPE); // Create an instance of t
const int in=8; //This is the ECHO pin of
const int out=9; //This is the TRIG pin of
// 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);
26°C
Partly cloudy
Search
ENG
US
21:39
01/11/2023


Flood Monitoring System Using
Early Flood Detection Technique
wokwi.com/projects/376687531343457281
WOKWI
SAVE
SHARE
Docs
SIGN UP
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:05.631
63%
Temperature:24.0
Humidity :40.0
Humidity: 40.00% Temperature: 24.00% Water Level:37
//Early Flood Detection Using IOT ~ A project by Sabyasachi Ghosh
//<LiquidCrystal.h> is the library for using the LCD 16x2
#include <LiquidCrystal.h>
//"DHT.h" is the library for using the Temperature sensor DHT22
#include "DHT.h"
#define DHTPIN A0 //here we are initialising
#define DHTTYPE DHT22 //We have to declare the t
LiquidCrystal lcd(2,3,4,5,6,7); // Create an instance of t
DHT dht(DHTPIN, DHTTYPE); // Create an instance of t
const int in=8; //This is the ECHO pin of
const int out=9; //This is the TRIG pin of
// 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);
26°C
Partly cloudy
Search
ENG
US
21:39
01/11/2023


Flood Monitoring System Using
Early Flood Detection Technique
wokwi.com/projects/376687531343457281
WOKWI
SAVE
SHARE
Docs
SIGN IN
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
//Early Flood Detection Using IOT ~ A project by Sabyasachi Ghosh
//<LiquidCrystal.h> is the library for using the LCD 16x2
#include <LiquidCrystal.h>
//"DHT.h" is the library for using the Temperature sensor DHT22
#include "DHT.h"
#define DHTPIN A0 //here we are initialising
#define DHTTYPE DHT22 //We have to declare the t
LiquidCrystal lcd(2,3,4,5,6,7); // Create an instance of t
DHT dht(DHTPIN, DHTTYPE); // Create an instance of t
const int in=8; //This is the ECHO pin of
const int out=9; //This is the TRIG pin of
// 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);
00:06.597
61%
Water Level:37%
Orange Alert!
Humidity: 40.00% Temperature: 24.00% Water Level:37
26°C
Partly cloudy
Search
ENG
US
21:39
01/11/2023


Flood Monitoring System Using
Early Flood Detection Technique
wokwi.com/projects/376687531343457281
WOKWI
SAVE
SHARE
Docs
SIGN UP
sketch.ino
diagram.json
libraries.txt
Library Manager
Simulation
//Early Flood Detection Using IOT ~ A project by Sabyasachi Ghosh
//<LiquidCrystal.h> is the library for using the LCD 16x2
#include <LiquidCrystal.h>
//"DHT.h" is the library for using the Temperature sensor DHT22
#include "DHT.h"
#define DHTPIN A0 //here we are initialising
#define DHTTYPE DHT22 //We have to declare the t
LiquidCrystal lcd(2,3,4,5,6,7); // Create an instance of t
DHT dht(DHTPIN, DHTTYPE); // Create an instance of t
const int in=8; //This is the ECHO pin of
const int out=9; //This is the TRIG pin of
// 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);
00:13.494
81%
Water Level:50%
Orange Alert!
Humidity: 40.00% Temperature: 24.00% Water Level:37
Humidity: 40.00% Temperature: 24.00% Water Level:50
26°C
Partly cloudy
Search
ENG
US
21:39
01/11/2023

Benefits of this project:

The flood monitoring and early warning project offers numerous significant benefits, including:

1. **Lives Saved:** Early warnings enable timely evacuation and preparedness, reducing the risk to human life during flooding.
2. **Property Protection:** Minimizing flood damage to infrastructure, homes, and businesses through early alerts and preparedness measures.
3. **Economic Preservation:** Reducing economic losses caused by flood damage to properties, agriculture, and infrastructure.
4. **Community Resilience:** Empowering communities with the knowledge and tools to respond effectively to flood risks, fostering resilience.
5. **Resource Efficiency:** Optimizing emergency response resource allocation by providing lead time for preparation.
6. **Environmental Conservation:** Enhancing environmental protection by enabling better planning to reduce the impact of floods on natural habitats.

7.**Data-Driven Decision-Making:** Utilizing data for informed decision-making in disaster management and infrastructure development.
8.**Technological Innovation:** Driving advancements in sensor technology, data analysis, and early warning systems.
9.**Improved Policy:** Supporting the development of policies for better disaster preparedness and management.
10.**Global Impact:** Contributing to global efforts in disaster risk reduction and climate change adaptation.
11.**Long-Term Sustainability:** Encouraging sustainable practices in flood management, considering environmental impacts and long-term solutions.
12.**Community Engagement and Education:** Raising awareness and educating the public on flood risks, safety measures, and emergency responses.

By implementing a flood monitoring and early warning system, these benefits collectively contribute to saving lives, protecting property, and bolstering community resilience in the face of natural disasters.

Testing and Deployment:

The testing and deployment of a flood monitoring and early warning project involve several crucial steps:

Testing Phase:

1. **Sensor Calibration:** Ensure the accuracy of sensors by calibrating them to collect precise data.
2. **Data Validation:** Verify and validate the data collected by sensors to confirm its accuracy and reliability.
3. **System Integration:** Test the integration of various sensors, communication systems, and data processing infrastructure to ensure seamless operation.
4. **Model Testing:** Validate predictive models used for flood risk assessment against historical data and known flood events.
5. **Response Simulation:** Simulate response scenarios based on warnings to ensure the effectiveness of communication and emergency protocols.

Deployment Phase:

- **Installation:** Deploy and install sensors in strategic flood-prone areas according to the project plan.
- **Data Transmission and Processing:** Set up communication systems and data processing infrastructure for real-time data transmission and analysis.
- **Community Awareness:** Educate and inform the local community about the system, emergency procedures, and the significance of early warnings.
- **Collaboration with Authorities:** Coordinate with local authorities, emergency services, and relevant stakeholders for effective response planning and communication protocols.
- **Real-time Monitoring:** Activate the system to collect and monitor real-time data for flood risks and warnings.
- **Continuous Evaluation:** Regularly evaluate and fine-tune the system based on performance, user feedback, and environmental changes.