|  |
| --- |
| Phase-3 |

|  |  |
| --- | --- |
| **Date** | **25.10.2023** |
| **Team ID** | **Proj\_223985\_team** |
| **Project Name** | **Flood monitoring and Early warning** |
| **Members** | **Rasswanth G**  **Vishwa S**  **Saravanan S**  **Pragadheesh M**  **Yogesh R** |

**Project Overview:**

The flood monitoring and early warning system is designed to detect and alert users to potential flood conditions. It incorporates various sensors and an ESP32 microcontroller to measure temperature, humidity, and water levels, providing real-time data to a cloud-based platform for monitoring.

**Purpose and Scope:**

The primary purpose of this system is to monitor environmental conditions and, more specifically, water levels in flood-prone areas. By doing so, it serves as an early warning system, helping to prevent damage and improve safety in flood-prone regions.

**Objectives:**

* Develop a system for real-time monitoring of temperature, humidity, and water levels.
* Interface with ThingSpeak, a cloud-based platform, to store and display data.
* Implement a warning mechanism through a buzzer when water levels reach a critical threshold.

**Hardware Components:**

* ESP32 microcontroller
* DHT22 sensor used for temperature and humidity
* Ultrasonic sensor used for water level measurement
* Buzzer used for early warning

**Components Descriptions:**

* ESP32 Microcontroller: The ESP32 is a versatile microcontroller that provides Wi-Fi connectivity, making it suitable for IoT applications. In this project, it acts as the central processing unit.
* DHT11 Sensor: The DHT11 sensor is utilized for temperature and humidity measurements. It provides accurate environmental data.
* Ultrasonic Sensor: The ultrasonic sensor is responsible for measuring water levels in rivers, lakes, or other bodies of water.
* Buzzer: The buzzer serves as an audible warning device. When water levels reach a critical threshold, it alerts users with a loud sound.

**Software Components:**

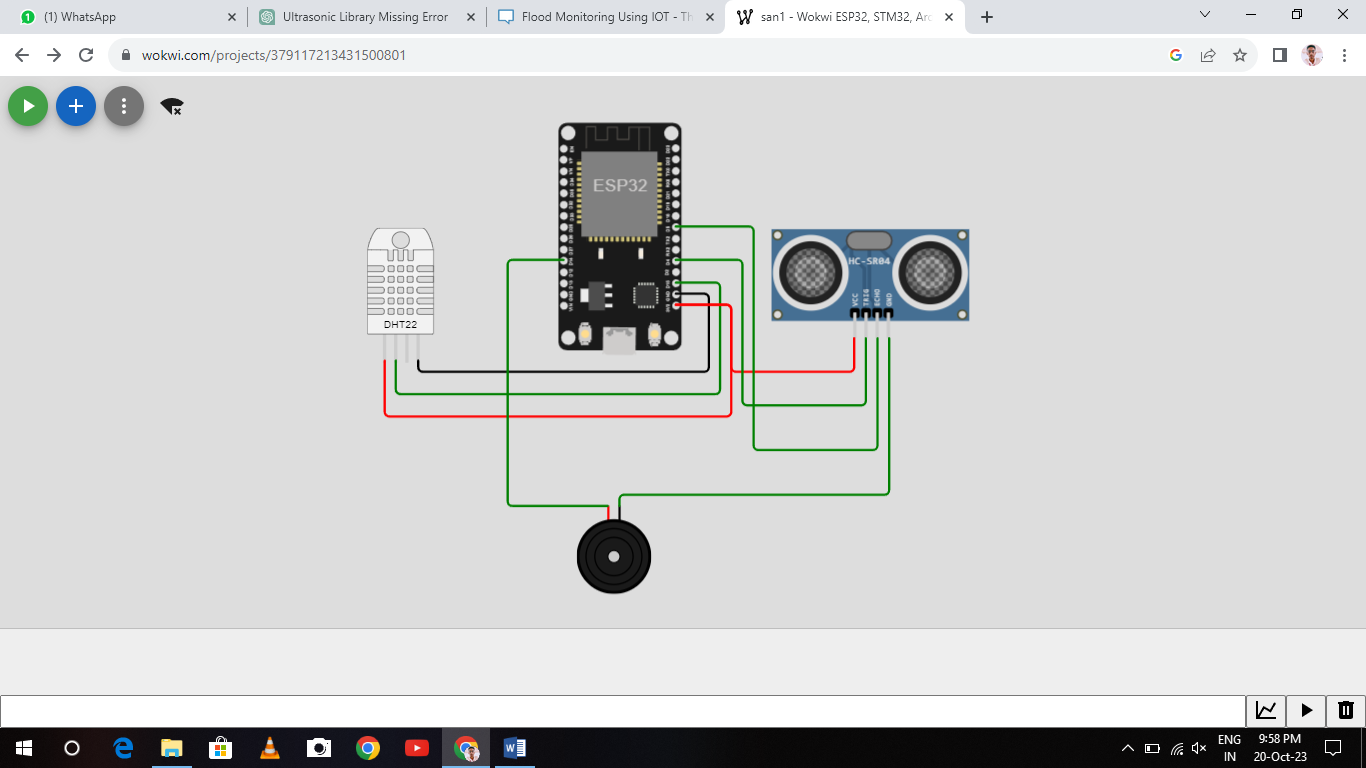
* DHTesp library for DHT22 sensor
* Ultrasonic library for ultrasonic sensor
* ThingSpeak library for cloud integration

**Software Descriptions:**

* DHTesp Library: This library enables communication with the DHT22 sensor, making it possible to read temperature and humidity values.
* Ultrasonic Library: The Ultrasonic library provides functions for interfacing with the ultrasonic sensor, allowing accurate water level measurements.
* ThingSpeak Library: The ThingSpeak library is used for sending data to the ThingSpeak platform, where it can be visualized and monitored.

**System Architecture:**

The system architecture involves the ESP32 as the central controller, which interfaces with the DHT22 and ultrasonic sensors. The ESP32 collects data from these sensors, processes it, and sends the data to the ThingSpeak platform over Wi-Fi. ThingSpeak then displays the data, and when water levels surpass a predefined threshold, the ESP32 triggers the buzzer to issue a warning.

****

**Source Code:**

#include <WiFi.h>

#include <DHTesp.h>

#include <Ultrasonic.h>

#include <ThingSpeak.h>

const char\* ssid = "Wokwi-GUEST";

const char\* password = "";

const unsigned long channelID = 2314097;

const char\* writeAPIKey = "CYPMNXUMXZ4YM8U6";

#define DHT\_PIN 15

DHTesp dht;

#define TRIGGER\_PIN 4

#define ECHO\_PIN 5

Ultrasonic ultrasonic(TRIGGER\_PIN, ECHO\_PIN);

#define BUZZER\_PIN 14

const int WaterLevelThreshold = 50;

WiFiClient client;

void setup() {

**Serial**.begin(115200);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

**Serial**.println("Connecting to WiFi...");

}

**Serial**.println("Connected to WiFi");

ThingSpeak.begin(client);

dht.setup(DHT\_PIN, DHTesp::DHT22);

pinMode(BUZZER\_PIN, OUTPUT);

}

void loop() {

TempAndHumidity data = dht.getTempAndHumidity();

long distance = ultrasonic.read();

if (isnan(data.temperature) || isnan(data.humidity) || distance == 0) {

**Serial**.println("Failed to read from sensors.");

return;

}

**Serial**.print("Temperature (C): ");

**Serial**.println(data.temperature);

**Serial**.print("Humidity (%): ");

**Serial**.println(data.humidity);

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

**Serial**.println(distance);

ThingSpeak.setField(1, data.temperature);

ThingSpeak.setField(2, data.humidity);

ThingSpeak.setField(3, distance);

int status = ThingSpeak.writeFields(channelID, writeAPIKey);

if (status == 200) {

**Serial**.println("Data sent to ThingSpeak successfully");

} else {

**Serial**.print("Data send failed, status code: ");

**Serial**.println(status);

}

if (distance < WaterLevelThreshold) {

digitalWrite(BUZZER\_PIN, HIGH);

} else {

digitalWrite(BUZZER\_PIN, LOW);

}

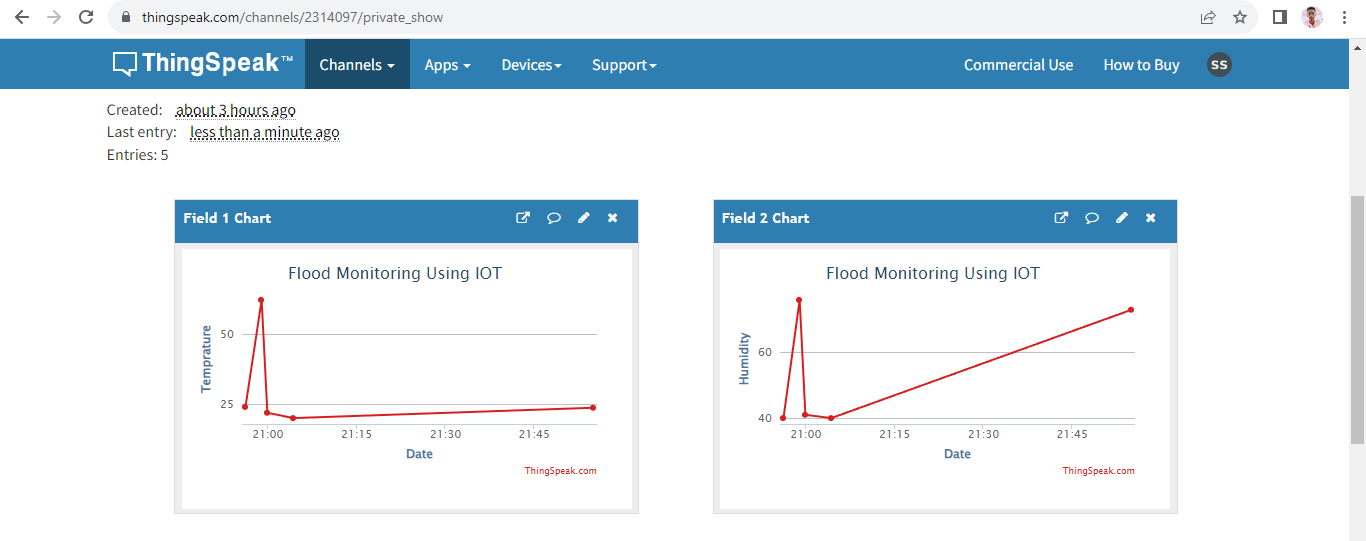
delay(900000); }

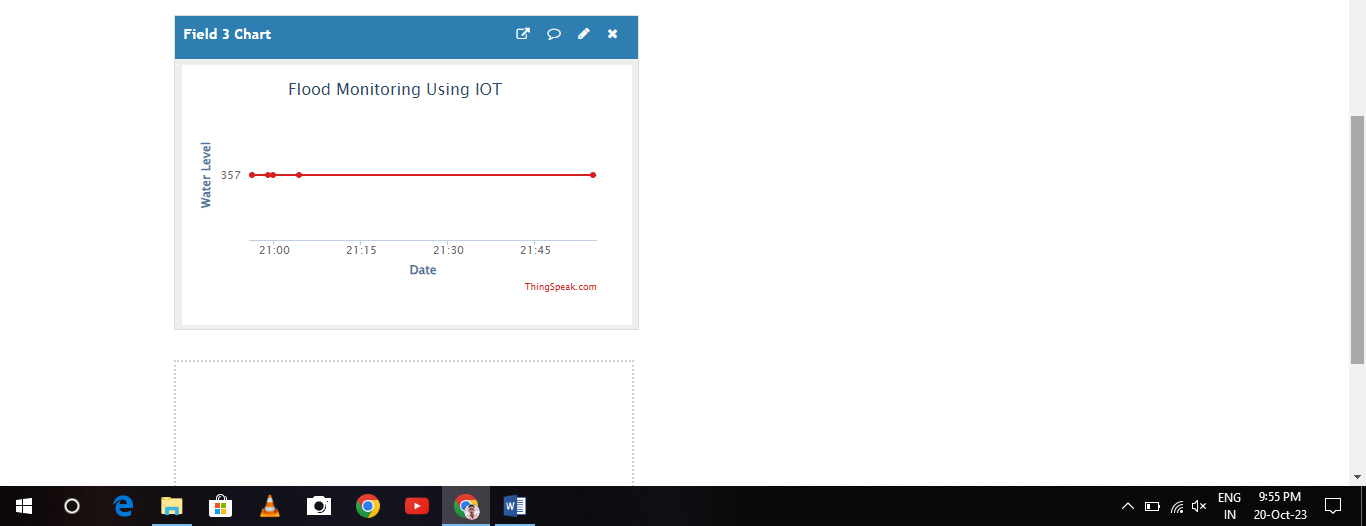
**Operation:**

The system operates by continuously reading temperature, humidity, and water level data from the DHT22 and ultrasonic sensors. It sends this data to ThingSpeak via Wi-Fi for remote monitoring. If the water level surpasses a predefined threshold, the buzzer is triggered to issue an early warning.

The significance of the water level threshold lies in defining the point at which a flood warning is triggered.

**Digital Outputs:**

****

****

**Conclusion:**

The flood monitoring and early warning system successfully monitors temperature, humidity, and water levels. It combines an ESP32 microcontroller, DHT22 sensor, ultrasonic sensor, and a buzzer. The system uses the libraries like DHTesp, Ultrasonic, and ThingSpeak for seamless data collection and cloud-based monitoring. Testing confirmed the system's accuracy and responsiveness. Future improvements may involve additional sensors and enhanced functionality. This project exemplifies the potential of IoT solutions for environmental monitoring and disaster mitigation.