IOT_PHASE-4

DEVELOPMENT PART-2

INTRODUCTION

- Flood monitoring involves the continuous observation and collection of data related to rainfall, water levels, river discharge, weather conditions, and other relevant parameters. Various sensors, satellites, and monitoring stations are used to gather this data. Additionally, historical flood records and topographic information are considered in the monitoring process.
- Early warning systems use the data collected through monitoring to provide advance notice and alerts to communities and authorities when there is an imminent risk of flooding. These systems can be automated or human-operated and utilize various communication channels to reach those at risk. Early warning systems aim to provide timely and accurate information to enable preparedness and evacuation, reducing the impact of floods on lives and property.

PROJECT GOALS AND OBJECTIVES

- Develop a robust IOT system to continuously monitor water levels in flood-prone areas.
- Establish real-time data transmission to a central server for timely analysis.
- Implement predictive algorithms to issue early flood warnings based on data trends.
- Ensure community engagement and awareness through alerts and notifications.5.Reduce flood-related damages and enhance disaster preparedness.

COMPONENTS:

Flood Monitoring and Early Warning Consists the following components

1. ESP32 Development Board (e.g., ESP32-WROOM-32)
2. Water level sensor (e.g., a float switch or water level sensor)
3. Buzzer or speaker for the alarm
4. Wokwi account (for simulating the project online)
5. Internet connection for Wokwi simulation

SYSTEM IMPLEMENTATION

- Purchase the necessary hardware components (ESP32, water level sensor, buzzer).
- Set up your ESP32 development environment (Arduino IDE or Platformio).
- Connect the real hardware following the same connections as in the Wokwi simulation.
- Upload the Arduino code to your ESP32.
- Deploy the system in the area you want to monitor for floods.
- Consider adding additional features like sending alerts to a server or smartphone.

PROCEDURE:

Step 1: setting up the Wokwi environment

1. Visit the Wokwi website (https://wokwi.Com/).
2. Create an account if you don't have one already.
3. Once logged in, click on "create a new simulation."
4. Select "ESP32 devkit" as your board.

Step 2: connecting the hardware in Wokwi*

1. In the Wokwi simulator, you can add components like the ESP32, water level sensor, and buzzer/speaker by dragging them from the components panel onto the virtual breadboard.
2. Connect the components using virtual jumper wires. Connect the power and ground pins appropriately.
3. Connect the water level sensor to an Analog input pin on the ESP32.
4. Connect the buzzer/speaker to a digital output pin on the ESP32.

Step 3: writing the Arduino code*

Here's a simple Arduino code example to get you started with flood monitoring and early warning. This code reads the water level from the sensor and triggers the alarm when the water level exceeds a certain threshold.

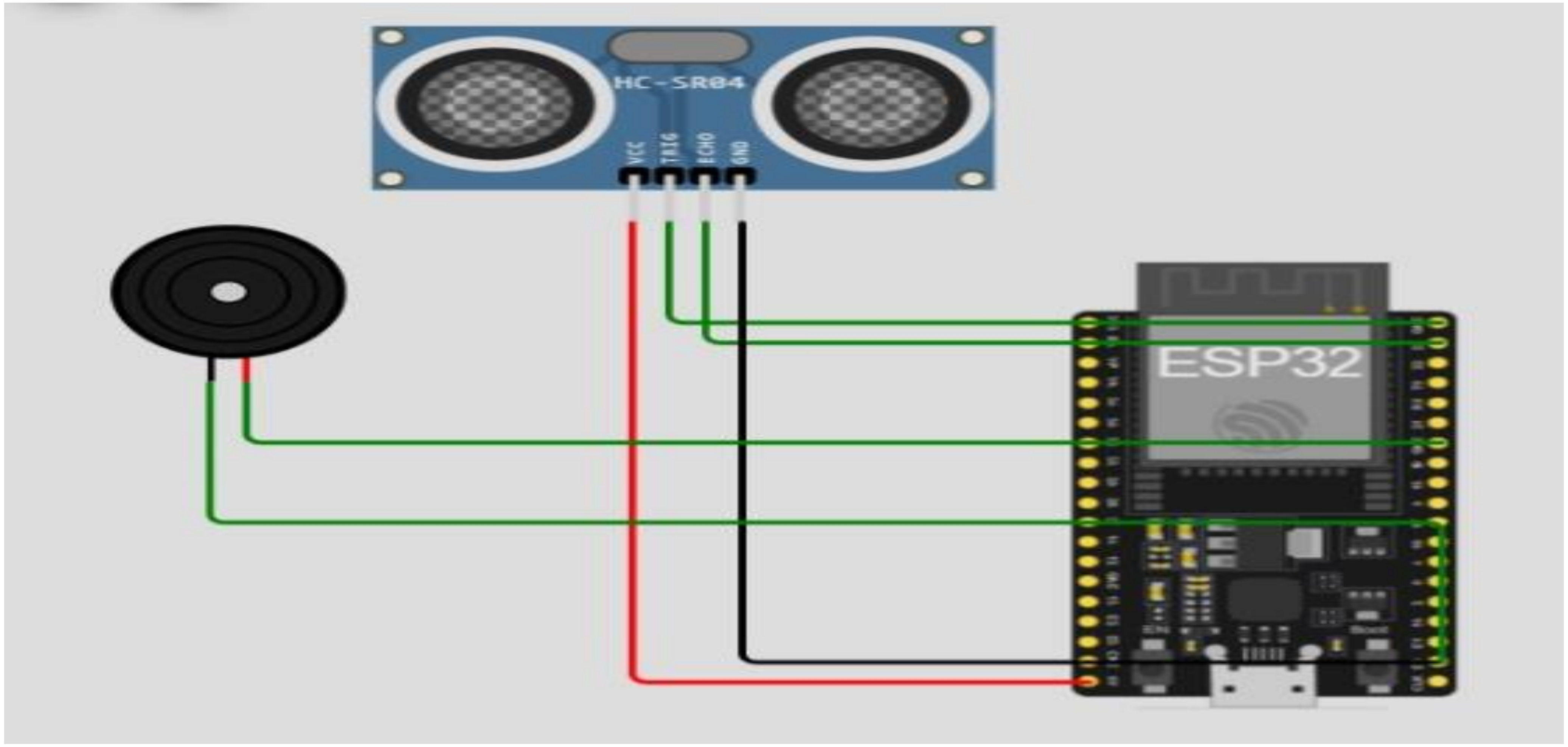
Step 4: Simulate the Flood Monitoring**

1. In the Wokwi simulator, click on the "Start Simulation" button to run your project.

2. Simulate the water level sensor by clicking on it and changing its value to simulate rising water levels.

3. Observe how the buzzer/speaker activates when the water level exceeds the threshold.

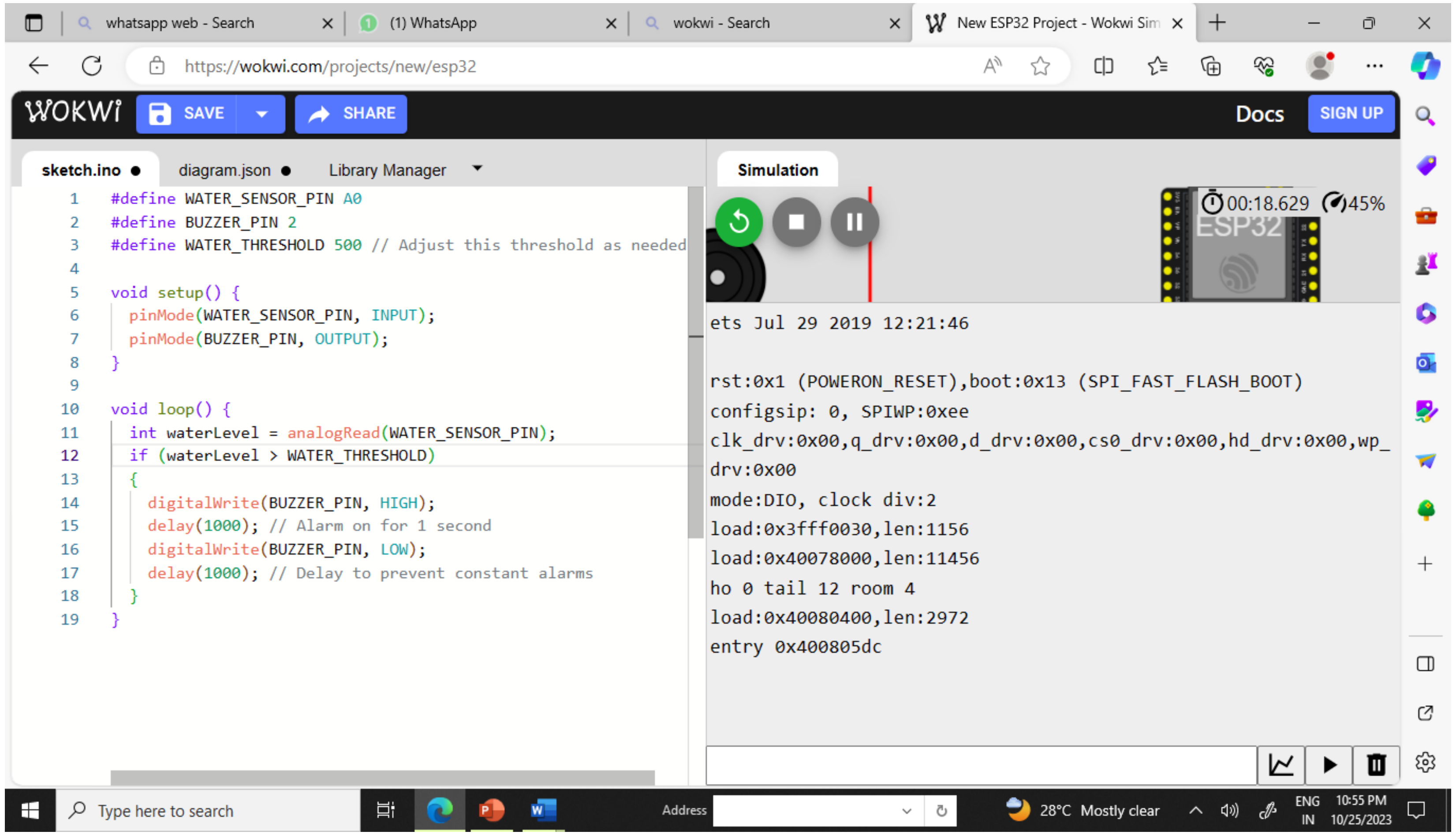
CIRCUIT:



PYTHON CODE:

```
#Define WATER_SENSOR_PIN a0
#define BUZZER_PIN 2
#define WATER_THRESHOLD 500 //
Adjust this threshold as needed
void setup() {
Pin mode(water,sensor,pin, INPUT);
Pin mode(buzzer,pin, output);
}
void loop() {
 int water level = Analog read(water,sensor,pin);
 if (water level > WATER_THRESHOLD)
 {
Digital write(buzzer,pin, HIGH);
delay(1000); // alarm on for 1 second    Digital write(buzzer,pin, LOW);
delay(1000);
// delay to prevent constant alarms
}}
```

OUTPUT:



```
#define WATER_SENSOR_PIN A0
#define BUZZER_PIN 2
#define WATER_THRESHOLD 500 // Adjust this threshold as needed

void setup() {
  pinMode(WATER_SENSOR_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
}

void loop() {
  int waterLevel = analogRead(WATER_SENSOR_PIN);
  if (waterLevel > WATER_THRESHOLD)
  {
    digitalWrite(BUZZER_PIN, HIGH);
    delay(1000); // Alarm on for 1 second
    digitalWrite(BUZZER_PIN, LOW);
    delay(1000); // Delay to prevent constant alarms
  }
}
```

```
ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_
drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
```

CONCLUSION

The flood monitoring and early warning system using IOT represents a crucial advancement in disaster management and public safety.

By improving preparedness and response, they help save lives, reduce property damage, and enhance resilience in flood-prone areas. As climate change continues to increase the frequency and severity of floods, investing in robust flood monitoring and early warning systems is not only advisable but essential for the safety and well-being of communities worldwide.