

FLOOD MONITORING EARLY WARNING SYSTEM

INTRODUCTION:

Floods are among the most devastating natural disasters, causing loss of lives, property damage, and economic disruptions worldwide. Flood monitoring and early warning systems are crucial tools for minimizing the impact of floods by providing timely information and alerts to vulnerable communities. This paper explores the key concepts and components of flood monitoring and early warning systems, their significance, and the technologies involved in their implementation.

HARDWARE REQUIRED:

- NODE MCU
- 16*2 LCD DISPLAY (12 CM MODULE)
- CSM 900A MODULE
- ULTRASONIC SENSOR
- FLOAT SENSOR
- ZERO PCB
- 5V 2A DC POWER SUPPLY

PROGRAM:

```
import time
import dht
import urequests
import network
from machine import Pin

# Define GPIO pins
TRIG_PIN = 2
ECHO_PIN = 3
BUZZER_PIN = 4
DHT_PIN = 5
```

```
LED_PIN = 6
```

```
# Define your Wi-Fi SSID and password
```

```
WIFI_SSID = "LIBIN"
```

```
WIFI_PASSWORD = "4651232960"
```

```
# Define your ThingSpeak API Key and Channel ID
```

```
THING_SPEAK_API_KEY = "GYZTW85RNGCZSLE9"
```

```
THING_SPEAK_CHANNEL_ID = "2316433"
```

```
# Function to establish a Wi-Fi connection
```

```
def connect_wifi(ssid, password):
```

```
    wlan = network.WLAN(network.STA_IF)
```

```
    wlan.active(True)
```

```
    if not wlan.isconnected():
```

```
        print("Connecting to Wi-Fi...")
```

```
        wlan.connect(ssid, password)
```

```
        while not wlan.isconnected():
```

```
            pass
```

```
    print("Connected to Wi-Fi:", wlan.ifconfig())
```

```
def distance_measurement():
```

```
    # Trigger ultrasonic sensor
```

```
    trigger = Pin(TRIG_PIN, Pin.OUT)
```

```
    trigger.on()
```

```
    time.sleep_us(10)
```

```
    trigger.off()
```

```
# Wait for echo to be HIGH (start time)
```

```
echo = Pin(ECHO_PIN, Pin.IN)
```

```
while not echo.value():
```

```

    pass

    pulse_start = time.time()

    # Wait for echo to be LOW (end time)
    while echo.value():
        pass

    pulse_end = time.time()

    # Calculate distance
    pulse_duration = time.time_diff(pulse_end, pulse_start)
    distance = pulse_duration / 58 # Speed of sound (343 m/s) divided by 2

    return distance

def read_dht_sensor():
    dht_sensor = dht.DHT22(Pin(DHT_PIN, Pin.IN))
    dht_sensor.measure()
    return dht_sensor.temperature(), dht_sensor.humidity()

buzz_start_time = None # To track when the buzzer started

# Connect to Wi-Fi
connect_wifi(WIFI_SSID, WIFI_PASSWORD)

while True:
    dist = distance_measurement()
    temp, humidity = read_dht_sensor()
    status = "No Flooding Detected"

    if dist < 50:
        # Turn on the buzzer and LED

```

```

Pin(BUZZER_PIN, Pin.OUT).on()

Pin(LED_PIN, Pin.OUT).on()

status = "Flooding Detected"

buzz_start_time = time.ticks_ms()

elif buzz_start_time is not None and time.ticks_diff(time.ticks_ms(), buzz_start_time) >= 60000: #
1 minute

    # Turn off the buzzer and LED after 1 minute

    Pin(BUZZER_PIN, Pin.OUT).off()

    Pin(LED_PIN, Pin.OUT).off()


print("Distance: {:.2f} cm".format(dist))

print("Temperature: {:.2f}°C, Humidity: {:.2f}%".format(temp, humidity))

print("Status:", status)


# Send data to ThingSpeak

try:

    data = {

        "api_key": THING_SPEAK_API_KEY,

        "field1": dist,

        "field2": temp,

        "field3": humidity,

    }

    response = urequests.post("https://api.thingspeak.com/update.json", json=data)

    response.close()

except Exception as e:

    print("Error sending data to ThingSpeak:", e)


time.sleep(2)

```

LINK:

<https://wokwi.com/projects/378661131954863105>

SIMULATION:

The screenshot displays the Wokwi web interface for a project titled "Flood Monitoring and Early Warning". The interface is split into two main sections: a code editor on the left and a simulation window on the right.

Code Editor (main.py):

```
1 import time
2 import dht
3 import urequests
4 import network
5 from machine import Pin
6
7 # Define GPIO pins
8 TRIG_PIN = 2
9 ECHO_PIN = 3
10 BUZZER_PIN = 4
11 DHT_PIN = 5
12 LED_PIN = 6
13
14 # Define your Wi-Fi SSID and password
15 WIFI_SSID = "LIBIN"
16 WIFI_PASSWORD = "4651232960"
17
18 # Define your ThingSpeak API Key and Channel ID
19 THING_SPEAK_API_KEY = "GVZTW85RNGCZSLE9"
20 THING_SPEAK_CHANNEL_ID = "2316433"
21
22 # Function to establish a Wi-Fi connection
23 def connect_wifi(ssid, password):
24     wlan = network.WLAN(network.STA_IF)
25     wlan.active(True)
26     if not wlan.isconnected():
27         print("Connecting to Wi-Fi...")
```

Simulation Window:

The simulation window shows a visual representation of the hardware components connected to a Raspberry Pi Zero. The components include:

- Ultrasonic Sensor (HC-SR04):** Connected to the Pi's GPIO pins (TRIG_PIN = 2, ECHO_PIN = 3).
- Buzzer:** Connected to the Pi's GPIO pin (BUZZER_PIN = 4).
- DHT22 Temperature and Humidity Sensor:** Connected to the Pi's GPIO pins (DHT_PIN = 5).
- LED:** Connected to the Pi's GPIO pin (LED_PIN = 6).

The simulation window also features a "Simulation" button (a green play icon) and a "Microsoft Store" button. The bottom of the screen shows a Windows taskbar with various application icons and system status indicators (time: 18:04, date: 01-11-2023).