

```
import time

import random

import paho.mqtt.client as mqtt


# MQTT Broker Settings

broker_address = "your_broker_address" # Replace with your MQTT broker
address

port = 1883

topic = "water_level"


# Simulated Water Level Sensor

def read_water_level():

    # Simulate water level measurement, replace this with actual sensor reading

    return random.uniform(0.0, 10.0)


# Initialize MQTT Client

client = mqtt.Client("WaterLevelSensor")

client.connect(broker_address, port)


while True:

    water_level = read_water_level()

    payload = f'{{"water_level": {water_level}}}'
```

Publish data to the MQTT broker

client.publish(topic, payload)

print(f"Published: {payload}")

time.sleep(60) # Send data every 60 seconds (adjust as needed)

IoT Flood Monitoring and Early Warning System

Project Overview

The IoT Flood Monitoring and Early Warning System is designed to monitor water levels in flood-prone areas and provide early flood warnings. This system utilizes IoT sensors and an MQTT-based communication protocol to transmit water level data to a central early warning platform.

Components of the System

IoT Sensors: Water level sensors are deployed in flood-prone areas to measure water levels. These sensors provide real-time data that is crucial for flood monitoring.

Python Script: A Python script is installed on the IoT sensors. This script is responsible for reading water level measurements from the sensors and sending this data to the central early warning platform using MQTT.

MQTT Broker: An MQTT broker acts as a central messaging hub. It receives data from the sensors and distributes it to the relevant components of the early warning platform.

Early Warning Platform: The early warning platform receives the water level data from the MQTT broker. It processes the data, analyzes it, and triggers alerts or warnings when water levels indicate a potential flood risk.

Python Script for IoT Sensors

The provided Python script in this document serves as an example of how IoT sensors can send water level data to the early warning platform using MQTT. Please note that this script is for demonstration purposes and includes a simulated water level measurement. In a real-world scenario, you should replace the simulated measurement with code to read data from an actual water level sensor.

Configuration

MQTT Broker Address: Replace "your_broker_address" in the Python script with the actual address of your MQTT broker or use a suitable MQTT broker service.

Data Interval: You can adjust the time interval for sending data (currently set to every 60 seconds) based on your project's requirements.

Conclusion

The IoT Flood Monitoring and Early Warning System is a crucial tool for monitoring and responding to flood situations. It provides real-time data that can

be used to trigger alerts and take preventive measures in flood-prone areas. To fully implement this system, you'll need to integrate the sensor data reception and alerting logic into your early warning platform.