IOT_Phase3

FLOOD MONITORING & EARLY WARNING

NAME:G.V.DEEPAK KUMAR

REG NO:610821106015

Deploying IoT sensors in flood-prone areas to measure water levels is a proactive approach to flood monitoring and management. To do this:

1.Select Sensor Type: Choose the appropriate water level sensors compatible with IoT technology. Ultrasonic sensors, pressure transducers, or capacitance sensors are commonly used for this purpose.

2.Sensor Placement: Install the sensors at strategic locations within the flood-prone areas, such as near rivers, streams, or floodplains.

3.IoT Connectivity: Configure the sensors to connect to an IoT network. This may involve using Wi-Fi, cellular, LoRa, or other suitable communication methods.

4.Data Transmission: Set up the sensors to regularly transmit water level data to a central server or cloud platform. Ensure data security and integrity.

5.Data Processing: Implement data processing algorithms to filter and analyze the incoming data for anomalies and flood potential.

6.Alert System: Develop an alert system that triggers notifications when water levels rise to a certain predefined threshold, indicating a potential flood event.

7.Integration: Integrate the sensor data with local weather information and other relevant data sources for a more comprehensive flood monitoring system.

IOT_Phase3

8.Monitoring and Maintenance: Regularly monitor the sensor network for functionality and perform maintenance as needed.

9.Community Outreach: Inform local authorities and communities about the system and its capabilities, so they can take timely actions in response to flood alerts.

10.Emergency Response Plan: Develop an emergency response plan that outlines actions to be taken when flood alerts are triggered, such as evacuation procedures and resource allocation.

This IoT-based flood monitoring system can greatly improve flood preparedness and response, potentially saving lives and reducing flood-related damages.

Python script on the IoT sensors to send collected water level data to the early warning platform.

```
import paho.mqtt.client as mqtt

import random

import time


# MQTT broker settings

mqtt_broker = "mqtt.eclipse.org"  # Replace with your MQTT broker address

mqtt_port = 1883


# MQTT topic to publish to

mqtt_topic = "water_level"


# Initialize MQTT client
```

```
client = mqtt.Client("WaterLevelSensor")
client.connect(mqtt_broker, mqtt_port)

try:
   while True:
      # Simulate collecting water level data (replace with real sensor data)
      water_level = round(random.uniform(0.0, 100.0), 2)  # Simulated value between 0 and 100

      # Prepare the data payload
      payload = {
         "sensor_id": "sensor123",  # Replace with your sensor ID
         "water_level": water_level,
         "timestamp": int(time.time())
      }

      # Publish the data to the MQTT broker
      client.publish(mqtt_topic, str(payload))
      print(f"Published data: {payload}")

      # Adjust the interval to your desired data sending frequency (e.g., every 5 minutes)
      time.sleep(300)  # 300 seconds (5 minutes)

except KeyboardInterrupt:
```

```
print("Data publishing stopped.")

client.disconnect()
```