FLOOD MONITORING AND EARLY WARNING

To deploy IoT sensors for measuring water levels in flood-prone areas, follow these general steps:

1. **Select Sensors:**

Choose water level sensors suitable for your specific environment and requirements. Consider factors like accuracy, durability, and communication capabilities.

2. **Choose Communication Protocol:**

Decide on a communication protocol for the sensors. Common choices include MQTT, CoAP, or HTTP. Ensure compatibility with your sensor and data collection platform.

3. **IoT Platform Selection:**

Choose an IoT platform for data collection and management. Popular platforms include AWS IoT, Azure IoT, and Google Cloud IoT. Set up an account and create a project.

4. **Configure IoT Devices:**

Configure each IoT sensor to connect to your chosen IoT platform. Set up credentials, communication protocols, and ensure proper security measures are in place.

5. **Install Sensors:**

Deploy sensors in flood-prone areas, ensuring they are securely placed to withstand environmental conditions. Connect sensors to a power source and verify their functionality.

6. **Test Communication:**

Confirm that sensors can communicate with the IoT platform. Send test data and check if it's correctly received and logged on the platform.

7. **Data Processing and Storage:**

Set up data processing and storage mechanisms on your IoT platform. Define how data should be handled, stored, and analyzed.

8. **Early Warning System:**

Implement an early warning system that triggers alerts based on the received water level data. This could involve setting thresholds and sending notifications when critical levels are reached.

9. **Power Management:**

Implement power management strategies for the sensors, especially if they are in remote areas. Consider using solar power or low-power modes to extend sensor lifetimes.

10. **Monitoring and Maintenance:**

Regularly monitor the sensors' performance and address any issues promptly. Plan for routine maintenance to ensure sensors continue to operate effectively.

Remember to adapt these steps based on the specific requirements of your project and the characteristics of the flood-prone areas where the sensors will be deployed.

Certainly! Let's assume you're using a hypothetical water level sensor that communicates over MQTT. Here's a basic Python script using the Paho MQTT library to send water level data to a server:

```python
Import paho.mqtt.client as mqtt
Import json
Import random
Import time

# Simulating water level data for testing
Def get_water_level():
    Return random.uniform(0, 100)

# MQTT settings
Broker_address = "mqtt.eclipse.org"
Port = 1883
Topic = "water_level_data"
```

```
# Callback when connection is established
Def on_connect(client, userdata, flags, rc):
   Print(“Connected with result code “+str(rc))
   Client.subscribe(topic)

# Callback when a message is received from the server
Def on_message(client, userdata, msg):
   Print(f”Received message: {msg.payload}”)

# Create MQTT client
Client = mqtt.Client()
Client.on_connect = on_connect
Client.on_message = on_message

# Connect to the broker
Client.connect(broker_address, port, 60)

# Loop to continuously send water level data
Try:
   While True:
      Water_level = get_water_level()
      Payload = {“water_level”: water_level}

      # Convert dictionary to JSON
      Payload_json = json.dumps(payload)

      # Publish data to the topic
      Client.publish(topic, payload_json)
```

```
    Print(f"Sent water level data: {water_level}")
    Time.sleep(10)  # Adjust the interval based on your requirements

Except KeyboardInterrupt:
  Print("Script terminated by user")
  Client.disconnect()
```

This script generates random water level data for testing purposes. In a real-world scenario, you would replace the `get_water_level` function with the logic to read data from your actual water level sensor.

Also, make sure to replace the MQTT broker details (`broker_address`, `port`, `topic`) with the specifics of your early warning platform.

NAME: PACHAIYAPPAN S

REG NO: 610821106066