

PHASE 3: DEVELOPMENT PART -1

- In our project we are using python script on the IOT sensors to send collected water level data to the early warnings platform. It imports libraries for MQTT communication and Ada fruit BME280 library for the temperature and humidity sensor.
- The script sets up configuration variables, including the MQTT broker's address and MQTT topics for different sensor data (temperature, humidity, water flow, and water level).It initializes the BME280 sensor using the Ada fruit library.
- It collects data from the BME280 sensor (temperature and humidity).
- In sensor , it creates a data payload in the form of a Python dictionary with sensor ID, timestamp, and sensor-specific data.
- It publishes these payloads to the respective MQTT topics using the MQTT client.
- After sending the data, it prints "Sent sensor data" as a confirmation.
- It then sleeps for 60 seconds before collecting and sending data again.

PROGRAM

```
import time
import paho.MQTT.client as MQTT
import board
import adafruit_bme280

MQTT_broker = "MQTT.early warning.com"
MQTT_topic_water_level = "water_level"
MQTT_topic_temperature = "temperature"
MQTT_topic_humidity = "humidity"
MQTT_topic_water_flow = "water_flow"

client = MQTT.Client()
client.connect(MQTT_broker)

bme280 = adafruit_bme280.Adafruit_BME280_I2C(board.I2C())

while True:
    temperature = bme280.temperature
    humidity = bme280.humidity
    water_flow = read_water_flow_sensor()
    water_level = read_water_level_sensor()

    payload_temperature = {
        "sensor_id": "temperature_sensor",
```

```

        "timestamp": int(time.time()),
        "temperature": temperature
    }
    payload_humidity = {
        "sensor_id": "humidity_sensor",
        "timestamp": int(time.time()),
        "humidity": humidity
    }
    payload_water_flow = {
        "sensor_id": "water_flow_sensor",
        "timestamp": int(time.time()),
        "water_flow": water_flow
    }
    payload_water_level = {
        "sensor_id": "water_level_sensor",
        "timestamp": int(time.time()),
        "water_level": water_level
    }

    client.publish(MQTT_topic_temperature, payload_temperature)
    client.publish(MQTT_topic_humidity, payload_humidity)
    client.publish(MQTT_topic_water_flow, payload_water_flow)
    client.publish(MQTT_topic_water_level, payload_water_level)
    print("Sent sensor data")
    time.sleep(60)

```

Output:

The script provides a framework for sending data from various sensors to an MQTT broker . In the output we see periodic messages indicating that the sensors data has been sent, assuming that MQTT broker connection is successful.