Environment Management system
Phase 4 submission

.

Phase 4 : Development Part 2
Topic: Environment Management system

Introduction:

An Environmental Monitoring System using the Internet of Things (IoT) is a cutting-edge solution that leverages interconnected devices, sensors, and data analytics to gather, manage, and analyze environmental data in real-time. This technology plays a crucial role in addressing environmental challenges, tracking the state of natural resources, and ensuring a sustainable future. It can be applied in various contexts, from urban areas to remote wilderness, helping to assess and manage environmental conditions more effectively.

Overview:

The use of IoT in an EMS allows for the monitoring and control of various environmental parameters, such as air quality, water quality, energy consumption, waste management, and more. Sensors and devices can be deployed to collect data on these parameters, which is then transmitted to a central system for analysis and action.

The real-time nature of IoT data allows organizations to respond quickly to environmental incidents or deviations from set targets. For instance, if a sensor detects a sudden increase in air pollution levels, an alert can be generated, enabling immediate action to be taken to mitigate the issue.

In summary, incorporating IoT into an EMS provides organizations with the ability to collect and analyze real-time environmental data, leading to improved environmental performance, resource efficiency, and sustainability. It allows for proactive environmental management, reduces costs, and enhances overall operational effectiveness.

VARIOUS TOOLS USED FOR THIS PROJECT :

Sensors: IoT environmental monitoring systems rely on a variety of sensors designed to measure parameters like temperature, humidity, air quality, water quality, soil moisture, radiation levels, and more. These sensors are strategically deployed to collect real-time data.

IoT Devices: These are the hardware components that house sensors, process data, and facilitate communication. Devices like Raspberry Pi, Arduino, or specialized IoT modules are commonly used to collect data from sensors and transmit it to central systems.

Communication Networks: Data from sensors is transmitted using various communication protocols, such as Wi-Fi, Bluetooth, LoRaWAN, Zigbee, or cellular networks. The choice of network depends on the specific application's range and data transfer requirements.

IoT Platform: Data collected by sensors is sent to an IoT platform or cloud service, such as AWS IoT, Google Cloud IoT, or Microsoft Azure IoT. These platforms provide storage, data processing, real-time monitoring, and visualization tools.

Data Processing and Analytics: The collected data is processed and analyzed to derive valuable insights. Advanced analytics techniques may be used to detect trends, anomalies, and patterns within the data.

Creating a real-time Environment Management platform involves a combination of front end and back end technologies. Here's a simplified outline using C and C++ and phython programing with wifi connection for the front end and Node.js for the back end:

Phython:

```
import network
import time
from machine import Pin,ADC
import dht
import ujson
from umqtt.simple import MQTTClient

# MQTT Server Parameters
MQTT_CLIENT_ID = "micropython-weather-demo"
MQTT_BROKER    = "broker.mqttdashboard.com"
MQTT_USER      = ""
MQTT_PASSWORD  = ""
MQTT_TOPIC     = "wokwi-weather"

sensor = dht.DHT22(Pin(15))
MQ7=ADC(Pin(35))
MQ8=ADC(Pin(32))
button=Pin(34,Pin.IN)
led=Pin(33,Pin.OUT)
min_rate=0
max_rate=4095

print("Connecting to WiFi", end="")
sta_if = network.WLAN(network.STA_IF)
sta_if.active(True)
sta_if.connect('Wokwi-GUEST', '')
while not sta_if.isconnected():
  print(".", end="")
  time.sleep(0.1)
print(" Connected!")

print("Connecting to MQTT server... ", end="")
client = MQTTClient(MQTT_CLIENT_ID, MQTT_BROKER, user=MQTT_USER, password=MQTT_PASSWORD)
client.connect()

print("Connected!")

prev_weather = ""
while True:
  CO_sensor=(MQ7.read())*100/(max_rate)
  print("CO Sensor value: " + "%.2f" % CO_sensor +"%")
  Hydrogen_sensor=(MQ8.read())*100/(max_rate)
  print("Soil Sensor value: " + "%.2f" % Hydrogen_sensor +"%")
  button_value=button.value()
  if button_value == True:
    led.value(000)
    print("It's Raining")
  else:
    led.value(0)
  print("Measuring weather conditions... ", end="")

  sensor.measure()
  message = ujson.dumps({
    "temp": sensor.temperature(),
    "humidity": sensor.humidity(),
  })


  if message != prev_weather:
    print("Updated!")
    print("Reporting to MQTT topic {}: {}".format(MQTT_TOPIC, message))
```

```
    client.publish(MQTT_TOPIC, message)
    prev_weather = message
  else:
    print("No change")
  time.sleep(1)
```

C++:

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>

// Replace these with your Wi-Fi credentials.
const char* WIFI_SSID = "YourWiFiSSID";
const char* WIFI_PASS = "YourWiFiPassword";

// Replace with your Adafruit IO credentials.
#define ADAFRUIT_IO_USERNAME "YourAdafruitUsername"
#define ADAFRUIT_IO_KEY "YourAdafruitAIOKey"

// Define the DHT sensor.
#define DHT_PIN 2           // The pin where your DHT sensor is connected.
#define DHT_TYPE DHT22      // DHT sensor type (DHT11, DHT22, AM2302, etc.)

DHT dht(DHT_PIN, DHT_TYPE);

WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, "io.adafruit.com", 1883, ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY);

// Define MQTT feeds.
Adafruit_MQTT_Publish temperature = Adafruit_MQTT_Publish(&mqtt, ADAFRUIT_IO_USERNAME "/feeds/temperature");
Adafruit_MQTT_Publish humidity = Adafruit_MQTT_Publish(&mqtt, ADAFRUIT_IO_USERNAME "/feeds/humidity");

void setup() {
  Serial.begin(115200);

  // Connect to Wi-Fi.
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi...");
  }

  Serial.println("Connected to WiFi");

  // Connect to Adafruit IO.
  mqtt.connect();
  Serial.println("Connected to Adafruit IO");
}

void loop() {
  // Read temperature and humidity data from the DHT sensor.
  float temperatureValue = dht.readTemperature();
  float humidityValue = dht.readHumidity();

  // Publish data to Adafruit IO.
  if (!isnan(temperatureValue)) {
    temperature.publish(temperatureValue);
    Serial.print("Temperature: ");
    Serial.println(temperatureValue);
  } else {
    Serial.println("Failed to read temperature");
  }

  if (!isnan(humidityValue)) {
    humidity.publish(humidityValue);
    Serial.print("Humidity: ");
```

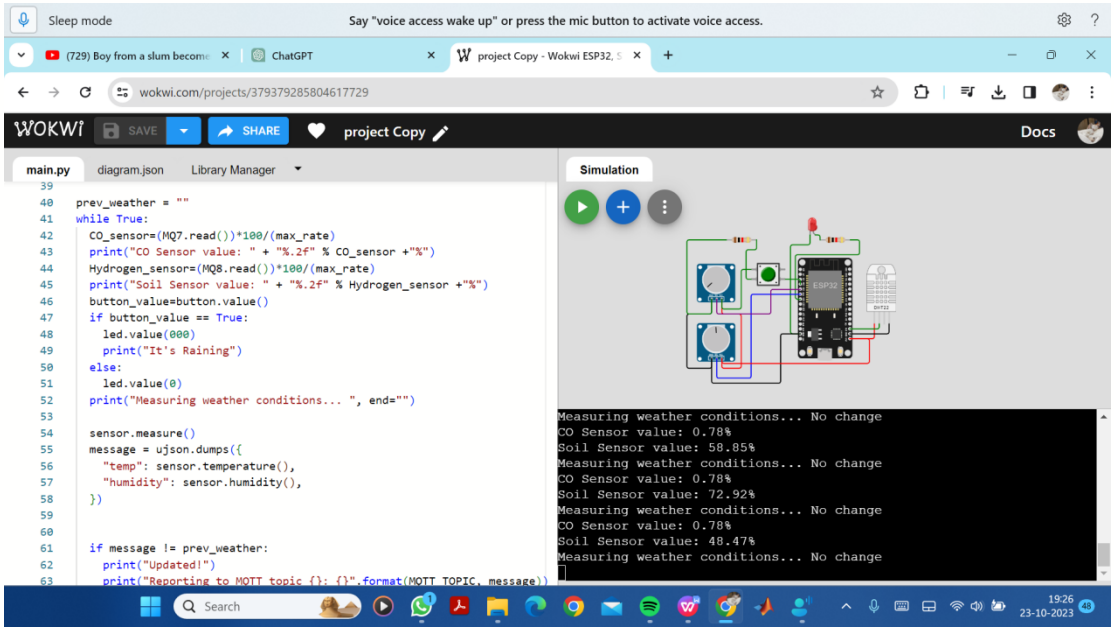
```
    Serial.println(humidityValue);
  } else {
    Serial.println("Failed to read humidity");
  }

  delay(60000); // Delay for 60 seconds (adjust as needed).
}
```

C program:

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>

// Replace these with your Wi-Fi credentials.
const char* WIFI_SSID = "YourWiFiSSID";
const char* WIFI_PASS = "YourWiFiPassword";

// Replace with your Adafruit IO credentials.
#define ADAFRUIT_IO_USERNAME "YourAdafruitUsername"
#define ADAFRUIT_IO_KEY "YourAdafruitAIOKey"

// Define the DHT sensor.
#define DHT_PIN 2           // The pin where your DHT sensor is connected.
#define DHT_TYPE DHT22      // DHT sensor type (DHT11, DHT22, AM2302, etc.)

DHT dht(DHT_PIN, DHT_TYPE);

WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, "io.adafruit.com", 1883, ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY);

// Define MQTT feeds.
Adafruit_MQTT_Publish temperature = Adafruit_MQTT_Publish(&mqtt, ADAFRUIT_IO_USERNAME "/feeds/temperature");
Adafruit_MQTT_Publish humidity = Adafruit_MQTT_Publish(&mqtt, ADAFRUIT_IO_USERNAME "/feeds/humidity");

void setup() {
  Serial.begin(115200);

  // Connect to Wi-Fi.
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi...");
  }

  Serial.println("Connected to WiFi");

  // Connect to Adafruit IO.
  mqtt.connect();
  Serial.println("Connected to Adafruit IO");
}

void loop() {
  // Read temperature and humidity data from the DHT sensor.
  float temperatureValue = dht.readTemperature();
  float humidityValue = dht.readHumidity();

  // Publish data to Adafruit IO.
  if (!isnan(temperatureValue)) {
    temperature.publish(temperatureValue);
    Serial.print("Temperature: ");
    Serial.println(temperatureValue);
  } else {
    Serial.println("Failed to read temperature");
  }
```

```
  if (!isnan(humidityValue)) {
    humidity.publish(humidityValue);
    Serial.print("Humidity: ");
    Serial.println(humidityValue);
  } else {
    Serial.println("Failed to read humidity");
  }

  delay(60000); // Delay for 60 seconds (adjust as needed).
}
```

Result :



Conclusion:

In conclusion, an Environmental Monitoring System using the Internet of Things (IoT) represents a transformative and highly valuable technology for addressing a wide range of environmental challenges. This system harnesses the power of interconnected sensors, devices, and data analytics to collect, manage, and analyze environmental data in real-time.