ENVIRONMENTAL MONITORING USING IOT

PHASE 4: DEVELOPMENT

Environmental monitoring

INTRODUCTION :

Environmental monitoring refers to the process of tracking and assessing various aspects of the natural environment to understand changes, trends, and potential impacts on ecosystems, human health, and the planet as a whole. It involves the collection, analysis, and interpretation of data related to air, water, soil, and other environmental factors. Environmental monitoring is crucial for several reasons:

Leak Detection Systems

Techniques of Environmental Monitoring

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

PYTHON:

```
import network
import time
from machine importPin,ADC
importdht
importujson
fromumqtt.simpleimportMQTTClient

# 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', '')
whilenotsta_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 = ""
whileTrue:
  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()
  ifbutton_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(),
```

```
  })


  ifmessage != 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_Clientmqtt(&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.
  floattemperatureValue = dht.readTemperature();
  floathumidityValue = 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 for60 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_Clientmqtt(&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");
```

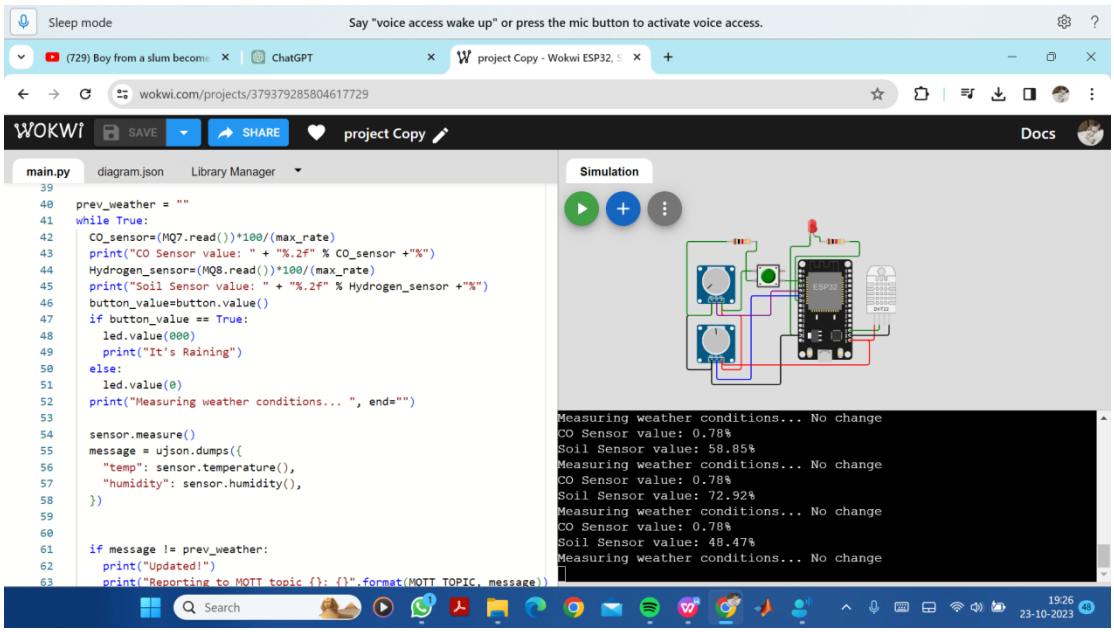
```
void setup() {
  Serial.begin(115200);

  // 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.
  floattemperatureValue = dht.readTemperature();
  floathumidityValue = 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 for60 seconds (adjust as needed).
}
```

MICROPROCESSOR PROGRAM

```
cpp
#include <Wire.h> // Include the Wire library for I2C communication

// Define the addresses of your sensors
const int humiditySensorAddress = 0x3F;
const int temperatureSensorAddress = 0x4A;
const int soundSensorPin = A0;
const int waterDetectorPin = 2;
const int airFlowSensorPin = A1;

void setup() {
Serial.begin(9600); // Initialize serial communication
Wire.begin();       // Initialize I2C communication
pinMode(waterDetectorPin, INPUT);
}
```

```
void loop() {
  float humidity = readHumidity(humiditySensorAddress);
  float temperature = readTemperature(temperatureSensorAddress);
  int soundLevel = analogRead(soundSensorPin);
  bool isWaterDetected = digitalRead(waterDetectorPin);
  int airFlow = analogRead(airFlowSensorPin);

  // Process and display the data
Serial.print("Temperature: ");
Serial.print(temperature);
Serial.println("°C");
Serial.print("Humidity: ");
Serial.print(humidity);
Serial.println("%");
Serial.print("Sound Level: ");
Serial.println(soundLevel);
Serial.print("Water Detected: ");
Serial.println(isWaterDetected ? "Yes" : "No");
Serial.print("Air Flow: ");
Serial.println(airFlow);

  // Add code to transmit data to your desired destination or take actions based on sensor
readings

delay(10000); // Delay for 10 seconds before taking the next reading
}

float readHumidity(int address) {
  // Implement code to read humidity from the I2C sensor
  // Return the humidity reading
}

float readTemperature(int address) {
  // Implement code to read temperature from the I2C sensor
  // Return the temperature reading
}

// Implement similar functions for other sensors

void processSensorsData(float humidity, float temperature, int soundLevel, bool
waterDetected, int airFlow) {
  // Add your logic for data processing here
}
```

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 systemharnesses the power of interconnected sensors, devices, and data analytics to collect, manage, and analyze environmental data in real-time.