ENVIRONMENTAL MONITORING PHASE_4

INTRODUCTION

Environmental monitoring is a critical practice that involves the collection and analysis of data to assess and manage various aspects of our natural surroundings. It plays a pivotal role in ensuring the sustainability and protection of ecosystems, human health, and the well-being of our planet. With the rapid advancements in technology, the Internet of Things (IoT) has emerged as a powerful tool to revolutionize environmental monitoring. IoT has enabled us to gather real-time, accurate, and comprehensive data, which has led to more effective decision-making and proactive measures for environmental conservation and protection.

CODE FOR THE REAL-TIME MONITORING OF 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)
```

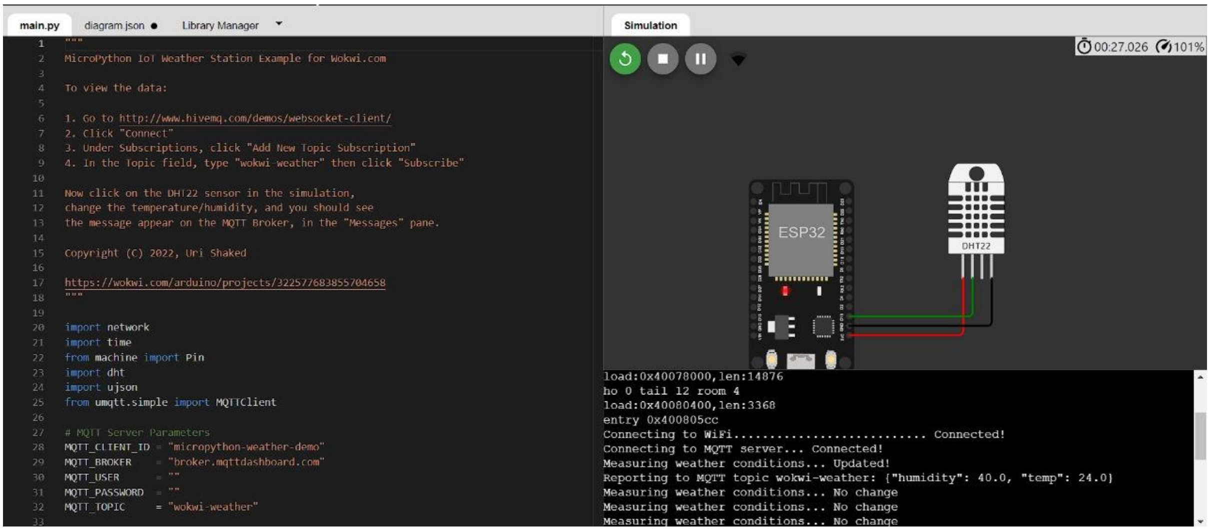
C++:

```
#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).
}
```



```
"""
MicroPython IoT Weather Station Example for Wokwi.com

To view the data:

1. Go to http://www.hivemq.com/demos/websocket-client/
2. Click "Connect"
3. Under Subscriptions, click "Add New Topic Subscription"
4. In the Topic field, type "wokwi-weather" then click "Subscribe"

Now click on the DHT22 sensor in the simulation,
change the temperature/humidity, and you should see
the message appear on the MQTT Broker, in the "Messages" pane.

Copyright (C) 2022, Uri Shaked

https://wokwi.com/arduino/projects/322577683855704658
"""

import network
import time
from machine import Pin
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"
```

```
load:0x40078000,len:14876
ho 0 tail 12 room 4
load:0x40080400,len:3368
entry 0x400805cc
Connecting to WiFi........................... Connected!
Connecting to MQTT server... Connected!
Measuring weather conditions... Updated!
Reporting to MQTT topic wokwi-weather: {"humidity": 40.0, "temp": 24.0}
Measuring weather conditions... No change
Measuring weather conditions... No change
Measuring weather conditions... No change
```

CONCLUSION

In conclusion, environmental monitoring using the Internet of Things (IoT) represents a transformative approach to understanding and mitigating the numerous challenges that our planet faces today. This technology has the potential to revolutionize the way we track and manage environmental parameters, contributing to the preservation of our ecosystems and the well-being of future generations.

IoT-enabled environmental monitoring offers several key advantages, including real-time data collection, enhanced accuracy, and cost-efficiency. It enables us to monitor a wide range of environmental factors such as air quality, water quality, climate conditions, biodiversity, and more.