

SPRINT 4

TEAM ID	PNT2022TMID17727
PROJECT NAME	Real-Time River water Quality Monitoring and Control System

CODE:

```
#include "DHTesp.h"
#include <cstdlib>
#include <time.h>
#include <WiFi.h>
#include <PubSubClient.h>

#define ORG "pfrli"
#define DEVICE_TYPE "Rasp"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
float temperature = 0;
int pH = 0;

String quality_status = "";
String temperture_status = "";

void setup() {
  Serial.begin(99900);

  wifiConnect();
  mqttConnect();
}
```

```

void loop() {

    srand(time(0));

    //initial variable
    int p;

    temperature = random(-20,40);
    pH = random(0,14);
    if(pH > 6.5 && pH < 8.5){
        p = 0;
    }
    else{
        p = 1;
    }

    //set a quality status

    switch (p) {
    case 0:
        quality_status = "Drinkable";
        break;
    case 1:
        quality_status = "Not Drinkable";
        break;
    }

    //Obviously the output.It is like json format 'cause it will help us for future sprints
    String payload = "{";
    payload+="\"pH level is \":";
    payload+=pH;
    payload+=",";
    payload+="\"Temperature of Water\":";
    payload+=(int)temperature;
    payload+=",";
    payload+="\"Alert\":"+"\""+quality_status+"\"}";
    Serial.println(payload);

    if(client.publish(publishTopic, (char*) payload.c_str()))
    {
        Serial.println("Publish OK");
    }
    else{

```

```

    Serial.println("Publish failed");
}
delay(1000);

if (!client.loop())
{
    mqttConnect();
}

}

void wifiConnect()
{
    Serial.print("Connecting to ");
    Serial.print("Wifi");
    WiFi.begin("Wokwi-GUEST", "", 6);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    Serial.print("WiFi connected, IP address: ");
    Serial.println(WiFi.localIP());
}

void mqttConnect()
{
    if (!client.connected())
    {
        Serial.print("Reconnecting MQTT client to ");
        Serial.println(server);
        while (!client.connect(clientId, authMethod, token))
        {
            Serial.print(".");
            delay(500);
        }

        Serial.println();
    }
}

```

```
}
```

DIAGRAM.JSON:

```
{
  "version": 1,
  "author": "PNT2022TMID51903",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -16.32, "left": -0.82, "attrs": { } },
    {
      "type": "wokwi-dht22",
      "id": "dht1",
      "top": -30.22,
      "left": 165.89,
      "attrs": { "temperature": "59.3" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [ ] ],
    [ "dht1:SDA", "esp:D15", "green", [ "v0" ] ],
    [ "dht1:VCC", "esp:3V3", "red", [ "v0" ] ],
    [ "dht1:GND", "esp:GND.1", "black", [ "v0" ] ]
  ]
}
```

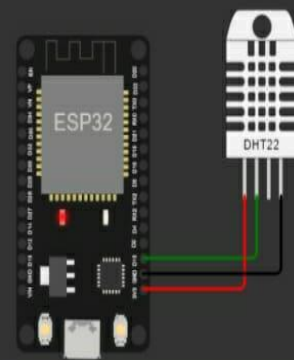
OUTPUT:

WOKWI SAVE SHARE Docs

esp32-dht22.ino diagram.json libraries.txt Library Manager

```
1 #include "DHTesp.h"
2 #include <stdlib>
3 #include <time.h>
4 #include <WiFi.h>
5 #include <PubSubClient.h>
6
7 #define ORG "pfrlli"
8 #define DEVICE_TYPE "Rasp"
9 #define DEVICE_ID "12345"
10 #define TOKEN "12345678"
11 #define speed 0.034
12
13 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/data/fmt/json";
15 char authMethod[] = "use-token-auth";
16 char token[] = TOKEN;
17 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
18
19 WiFiClient wifiClient;
20 PubSubClient client(server, 1883, wifiClient);
21 float temperature = 0;
22 int pH = 0;
23
24 String quality_status = "";
```

Simulation 02:27.406 99%



{ "pH level is ":2,"Temperature of Water":11,"Alert":"Not Drinkable" }

Publish OK

{ "pH level is ":11,"Temperature of Water":-10,"Alert":"Not Drinkable" }

WOKWI SAVE SHARE Docs

esp32-dht22.ino diagram.json libraries.txt Library Manager

```
1 #include "DHTesp.h"
2 #include <stdlib>
3 #include <time.h>
4 #include <WiFi.h>
5 #include <PubSubClient.h>
6
7 #define ORG "pfrlli"
8 #define DEVICE_TYPE "Rasp"
9 #define DEVICE_ID "12345"
10 #define TOKEN "12345678"
11 #define speed 0.034
12
13 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/data/fmt/json";
15 char authMethod[] = "use-token-auth";
16 char token[] = TOKEN;
17 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
18
19 WiFiClient wifiClient;
20 PubSubClient client(server, 1883, wifiClient);
21 float temperature = 0;
22 int pH = 0;
23
24 String quality_status = "";
25 String temperture_status = "";
```

Simulation 02:16.985 99%

Connected to wifi..WiFi connected, IP address: 10.10.0.2
Connecting MQTT client to
pfrlli.messaging.internetofthings.ibmcloud.com

{ "pH level is ":4,"Temperature of Water":-9,"Alert":"Not Drinkable" }

Publish OK

{ "pH level is ":5,"Temperature of Water":6,"Alert":"Not Drinkable" }

Publish OK

{ "pH level is ":10,"Temperature of Water":-13,"Alert":"Not Drinkable" }

Publish OK

{ "pH level is ":10,"Temperature of Water":32,"Alert":"Not Drinkable" }

Publish OK

{ "pH level is ":3,"Temperature of Water":27,"Alert":"Not Drinkable" }

Publish OK

{ "pH level is ":0,"Temperature of Water":-1,"Alert":"Not Drinkable" }

Publish OK

{ "pH level is ":1,"Temperature of Water":2,"Alert":"Not Drinkable" }

Publish OK

{ "pH level is ":8,"Temperature of Water":26,"Alert":"Drinkable" }

Publish OK