**SPRINT2**

**CODE:**

|  |  |
| --- | --- |
| Date | 8 November 2022 |
| Team id | PNT2022TMID50683 |
| Project name | Real Time River Water Quality Monitoring And Control System |
| Maximum marks | 5 marks |

#include <time.h>

#include <WiFi.h>

#include <PubSubClient.h>

#define ORG "bxsnm3"

#define DEVICE\_TYPE "nodemcu"

#define DEVICE\_ID "12345"

#define TOKEN "CvF9tldLEy0-)U&0B0"

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 water\_status="";

void setup() {

**Serial**.begin(99900);

   wifiConnect();

   mqttConnect();

}

void loop() {

  srand(time(0));

    //initial variables and random generated data

    temperature = random(0,100);

    ph = random(0,14);

    //set a flame status

    if(ph <= 8 && ph >= 6){

        water\_status = "Water is good!!";

    }

    else{

        water\_status = "Water is bad :(";

    }

    //json format for IBM Watson

    String payload = "{";

    payload+="\"ph\":";

    payload+=ph;

    payload+=",";

    payload+="\"temp\":";

    payload+=(int)temperature;

    payload+=",";

    payload+="\"water\_status\":\""+water\_status+"\"}";

    if(client.publish(publishTopic, (char\*) payload.c\_str()))

    {

**Serial**.println("Publish OK");

    }

    else{

**Serial**.println("Publish failed");

    }

delay(100);

    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();

  }

}