

ASSIGNMENT4

NAME	Yogarani S
REGISTER NUMBER	113119UG04117
TEAM ID	PNT2022TMID22430
DATE	25 October 2022
MAXIMUM MARKS	4 Marks

QUESTION: Write code and connections in wokwi for ultrasonic sensor.
Whenever distance is less than 100 cms send "alert" to ibm cloud and display in device recent events.

CODE:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#define ORG "q1wscz"
#define DEVICE_E "sampledevice"
#define DEVICE_D "24052002"
#define TOKEN "K9)lI1C@tX6y0(J6L1"
const int T_PIN = 5;
const int E_PIN = 4;
//----- Customise the above values -----
char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; // Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic
name and type of event perform and
format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String"; // cmd
REPRESENT command type
AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth"; // authentication
method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_E ":"
DEVICE_D; //client id
//
WiFiClient wifiClient; // creating the instance for
wificlient
PubSubClient client(server, 1883, wifiClient); //calling
the predefined client id by passing
parameter like server id, port and wificredential
```

```

void setup() {
  Serial.begin(115200);
  pinMode(T_PIN, OUTPUT);
  pinMode(E_PIN, INPUT);
  wificonnect();
  mqttconnect();
}
float readDistanceCM() {
  digitalWrite(T_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(T_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(T_PIN, LOW);
  int duration = pulseIn(E_PIN, HIGH);
  return duration * 0.034 / 2;
}
void loop() {
  float distance = readDistanceCM();
  Serial.print("Measured distance: ");
  Serial.println(distance);
  if(distance<=100){
    PublishData(distance);
  }
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}
void PublishData(float distance) {
  mqttconnect();//function call for connecting to ibm
  /*
  creating the String in in form JSon to update the data to
  ibm cloud
  */
  bool status=true;
  String payload = "{\"ALERT_MESSAGE\":";
  payload += status;
  payload += "," " \"DISTANCE\":";
  payload += distance;
  payload += "}";
  Serial.print("Sending payload: ");
  Serial.println(payload);
}

```

```

if (client.publish(publishTopic, (char*) payload.c_str()))
{
    Serial.println("Publish ok");// if it sucessfully upload
    data on the cloud then it will print
    publish ok in Serial monitor or else it will print publish
    failed
} else {
    Serial.println("Publish failed");
}
}

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

void wificonnect() //function defination for wificonnect
{
    Serial.println();
    Serial.print("Connecting to ");
    WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi
    credentials to establish the
    connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void initManagedDevice() {
    if (client.subscribe(subscribetopic)) {
        Serial.println((subscribetopic));
        Serial.println("subscribe to cmd OK");
    }
}

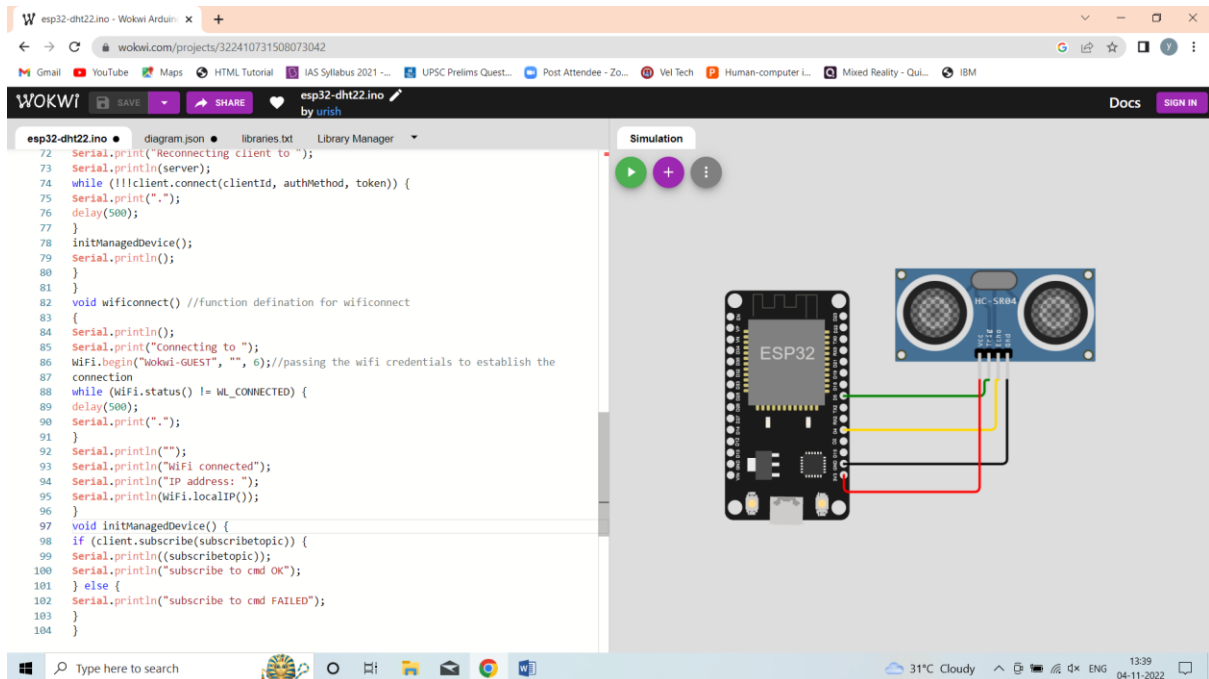
```

```

} else {
Serial.println("subscribe to cmd FAILED");
}
}

```

Output:



IBM CLOUD IMAGE:

