

### Assignment -4

Assignment Date	28 October 2022
Student Name	Balaji L
Student Roll Number	722819106006
Maximum Marks	2 Marks

Write code and connections in wokwi for the ultrasonic sensor.

Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

Upload document with wokwi share link and images of IBM cloud

Program:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
const int T=2;
const int E=21;
long d;
float Distance;
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-----credentials of IBM Accounts-----

#define ORG "4iin2l" //IBM ORGANITION ID
#define DEVICE_TYPE "iot" //Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "a8@kKmmz05RydrYQPr" //Token
String data3;

//----- 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_TYPE ":" DEVICE_ID; //client id

//-----
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client id by
passing parameter like server id, port and wificredential
void setup() // configureing the ESP32
{
    Serial.begin(115200);
    pinMode(T, OUTPUT);
    pinMode(E, INPUT);
    Serial.println();
    wificonnect();
    mqttconnect();
}

void loop() // Recursive Function
```

```

{
    digitalWrite(T,LOW);
    delay(1000);
    digitalWrite(T,HIGH);
    delay(1000);
    digitalWrite(T,LOW);
    d=pulseIn(E,HIGH);
    Distance=d*(0.034/2);
    Serial.print("Distance in Cm:");
    Serial.println(Distance);
    if(Distance<100)
    {
        Serial.println("!!ALERT!!");
        delay(1000);
        PublishData(Distance);
        delay(1000);
        if (!client.loop()) {
            mqttconnect();
        }
    }
    delay(1000);
}

/*.....retrieving to Cloud.....*/

void PublishData(float dist) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSon to update the data to ibm cloud
    */
    String payload = "{\"Distance\":\"";
    payload += dist;
    payload += "\",\"!!ALERT!!\":\"\"Distance is less than 100 cms\"";
    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 ");

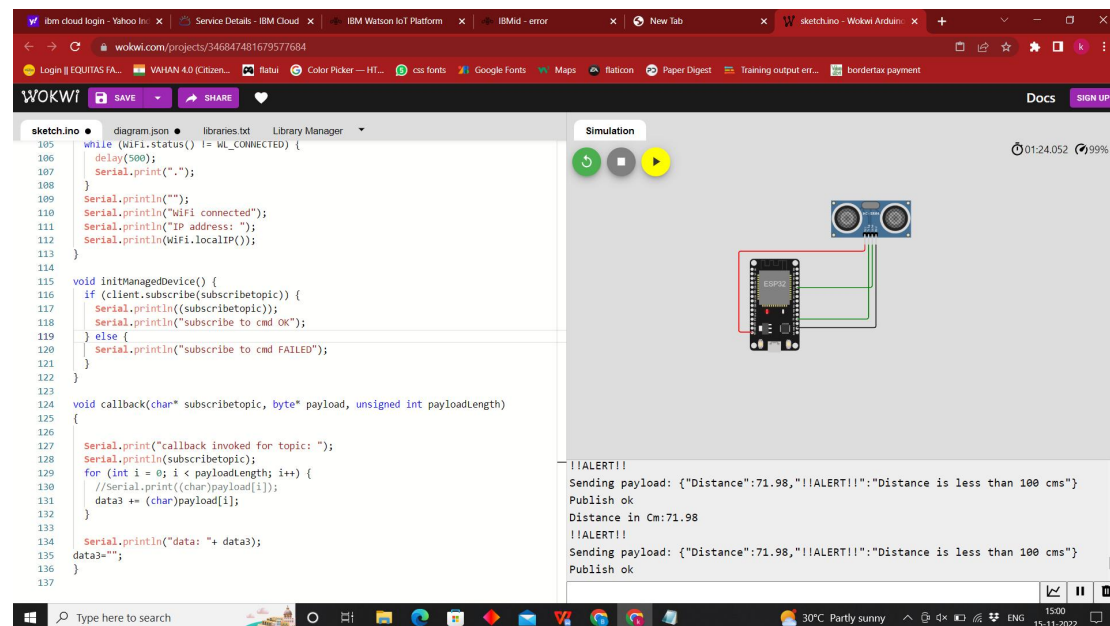
WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

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

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
    Serial.print("callback invoked for topic: ");
    Serial.println(subscribetopic);
    for (int i = 0; i < payloadLength; i++) {
        //Serial.print((char)payload[i]);
        data3 += (char)payload[i];
    }

    Serial.println("data: "+ data3);
    data3="";
}

```



Wokwi simulation link:

<https://wokwi.com/projects/346847481679577684>

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar labeled 'Search by Device ID' is present. The main content area shows a table of devices. The first device listed has ID 12345, status 'Disconnected', device type 'iot', class ID 'Device', and was added on '15 Nov 2022 14:50'. Below the table, the 'Recent Events' tab is selected, showing a stream of data events. The events are listed in a table with columns: Event, Value, Format, and Last Received. The events are JSON payloads containing distance and alert information.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Disconnected	iot	Device	15 Nov 2022 14:50	

Event	Value	Format	Last Received
Data	{\"Distance\":71.98,\"!ALERT!\":0,\"Distance is less th...	json	a few seconds ago
Data	{\"Distance\":71.98,\"!ALERT!\":0,\"Distance is less th...	json	a few seconds ago
Data	{\"Distance\":71.98,\"!ALERT!\":0,\"Distance is less th...	json	a few seconds ago
Data	{\"Distance\":71.98,\"!ALERT!\":0,\"Distance is less th...	json	a few seconds ago
Data	{\"Distance\":71.98,\"!ALERT!\":0,\"Distance is less th...	json	a few seconds ago