

ASSIGNMENT - 4

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

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

#define ORG "prbqrn"                            // IBM organisation id
#define DEVICE_TYPE "Ultrasonic"                // Device type mentioned in ibm watson iot platform
#define DEVICE_ID "Assignment"                  // Device ID mentioned in ibm watson iot platform

#define TOKEN "6qL3DUu-zuo8yPI7tS"             // Token
#define speed 0.034
#define led 14 String
data3;
int LED = 4;

//.....customise above values .....

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // server name
char publishTopic[] = "iot-2/evt/sreedhar/fmt/json";             // topic name and type of event perform and format in which data
to be send
char topic[] = "iot-2/cmd/led/fmt/String";                       // cmd Represent type and command is test format of strings
char authMethod[] = "use-token-auth";                           // authentication method char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;       //Client id

//.....

WiFiClient wifiClient;                                         // creating instance for wificlient
PubSubClient client(server, 1883, wifiClient);                 // calling the predefined client id by passing parameter like server id,port
and wifi credential

const int trigpin=5; const
int echopin=18;
String command;
String data="";

long duration;
float dist;

void setup()

{
```

```

    Serial.begin(115200);
    pinMode(led, OUTPUT);
    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);
    wifiConnect();
    mqttConnect();
}

void loop() { bool isNearby
= dist < 100;
    digitalWrite(led, isNearby);

    publishData();
    delay(500);

    if (!client.loop())
    {
        mqttConnect(); // function call to connect to ibm
    }
}

/* _____retrieving to cloud_____ */

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);
        }
        initManagedDevice();
        Serial.println();
    }
}

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

void publishData()
{

```

```

    digitalWrite(trigpin,LOW);
    digitalWrite(trigpin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin,LOW);
    duration=pulseIn(echopin,HIGH);
    dist=duration*speed/2;
    if(dist<100)
    {
        digitalWrite(LED,HIGH);
        String payload = "{\"Alert Distance\".:\"";
        payload += dist;
        payload += "}";

        Serial.print("\n");
        Serial.print("Sending          payload:          ");
        Serial.println(payload);
        if (client.publish(publishTopic, (char*) payload.c_str())) // if data is uploaded to cloud successfully,prints publish ok else prints
        publish failed
        {
            Serial.println("Publish OK");
        }

    }
    if(dist>100)
    {
        digitalWrite(LED,HIGH);
        String payload = "{\"Distance\".-\"";
        payload += dist;
        payload += "}";

        Serial.print("\n");
        Serial.print("Sending          payload:          ");
        Serial.println(payload);
        if(client.publish(publishTopic, (char*) payload.c_str()))
        {
            Serial.println("Publish OK");
        }
    else
    {
        digitalWrite(LED,LOW);
        Serial.println("Publish FAILED");
    }

}

}

```

OUTPUT :

Code simulation on wokwi

WOKWI

SAVE

SHARE

esp32-dht22.ino by urish

Docs

esp32-dht22.ino

diagram.json

libraries.txt

Library Manager

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 //----- credentials of IBM Accounts -----
5
6 #define ORG "prbrqn"
7 #define DEVICE_TYPE "ultrasonic"
8 #define DEVICE_ID "Assignment"
9 #define TOKEN "6ql300u-zuo8yPl7tS"
10 #define speed 0.034
11 #define led id
12 String data3;
13 int LED = 4;
14
15 //----- customise above values -----
16
17 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
18 char publishTopic[] = "iot-2/evt/sreedhar/fmt/json";
19 char topic[] = "iot-2/cmd/led/fmt/String";
20 char authMethod[] = "use-token-auth";
21 char token[] = TOKEN;
22 char clientId[] = "dt:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
23
24 //-----
25
26 WiFiClient wificlient;
27 PubSubClient client(server, 1883, wificlient);
28
29 const int trigpin=5;
30 const int echopin=18;
31 String command;
32 String data="";

```

Simulation

00:51.441

101%

Publish OK

Sending payload: {"Distance":400.01}

Publish OK

Sending payload: {"Distance":399.94}

Data sent to IBM Cloud with distance:

Browse

Action

Device Types

Interfaces

Add Device

Assignment

Disconnected

GENSON

Device

Oct 29, 2022 0:51:44

Identity

Device Information

Recent Events

State

Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
distance	{"distance":141.32}	json	a few seconds ago
distance	{"distance":141.32}	json	a few seconds ago
distance	{"distance":141.32}	json	a few seconds ago
distance	{"distance":141.32}	json	a few seconds ago
distance	{"distance":141.32}	json	a few seconds ago

1 Simulation running