ultrasonic sensor

ASSIGNMENT-4

code

#include <WiFi.h> #inc1ude <PubSubC1ient.h> void callback(char* subscribetopic,

byte* payload, unsigned int payloadLength);

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

#define ORG "yt1use"//IBM ORGANITION ID

#define DEVICE TYPE "2702"//Device type mentioned in ibm watson IOT Platform #define DEVICE ID
"12345"//Device ID mentioned in ibm watson IOT Platform #define TOKEN
"O+n)Eh+lNX0y3?rG!8"
//Token String

data3;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; char publishTopic[] = "iot-2/evt/Data/fmt/json"; char subscribetopic[] = "iot-2/cmd/test/fmt/String”; char authMethod[]

= "use-token-auth"; char token[] = TOKEN; char clientId[] = "d:" ORG ":" DEVICE TYPE

":" DEVICE_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback ,wifiClient); const int trigPin = 5;

const int echoPin = 18; #define SOUND_SPEED 0.034 long duration; float

distance; void setup() {

Serial.begin(115200); pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT);

wificonnect(); mqttconnect();

} void

loop()

{ digita1Write(trigPin,

LOW);

```
digita1Write(trigPin,

HIGH);

de1ayMicroseconds(10);

digita1Write(trigPin,

LOW); duration =

pulseIn(echoPin, HIGH);

distance = duration * SOUND_SPEED/2; Serial.print("Distance (cm): "); Serial.println(distance);
if(distance<100)


Serial.print1n("ALERT!!"); delay(1000); Pub1ishData(distance); delay(1000);

if(!client.loop()) { mqttconnect();


} delay(1000);

} void PublishData(float dist) { mqttconnect(); String

payload =

"(\"Distance\":"; payload += dist; payload += ",\"ALERT!!\":""\"Distance less

than 100cms\""; payload += "}";

Serial.print("Sending payload: ");

Serial.println(payload);


if(client.publish(publishTopic, (char*) payload.c_str())) (

Serial.println("Publish ok");

} else (

Serial.print1n("Publish failed");
```

```
void mqttconnect() { if(!c1ient.connected()) { Serial.print("Reconnecting client to ");
Serial.print1n(server); while (!!!client.connect(clientId, authMethod, token)) (
Serial.print("."); de1ay(500);

initManagedDevice();
Serial.print1n();

} void
wificonnect()

Serial.println(); Serial.print("Connecting to "); WiFi.begin("Wokwi-GUEST", "",
6); while (WiFi.status() != WL CONNECTED) { delay(500);
Serial.print(".");

Serial.print1n(""); Serial.println("WiFi connected"); Serial.println("IP address: ");
Serial.print1n(WiFi.localIP );

void initManagedDevice() (
if(client.subscribe(subscribetopic)) { Serial.println((subscribetopic)); Serial.println("subscribe to cmd
OK");
} else (
Serial.print1n("subscribe to cmd FAILED");
```

} void callback(char* subscribetopic, byte* payload, unsigned int

payloadLength)

Serial.print("callback invoked for topic:

"); Serial.print1n(subscribetopic); for (int i

= 0; i < payloadLength; i++) {

//Serial.print((char)pay1oad[i]); data3 += (char)payload[i];

Serial.print1n("data: "+ data3); data3="";

DIAGRAM.JSON

"version": 1,

"author": "IRFANA FATHIMA A 19IT007",

"editor": "wokwi", "parts": [

{ "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 6, "left": -66, "attrs": (} },

("type": "wokwi-hc-sr04", "id": "ultrasonicl", "top": 32.56, "left": 81.02, "attrs": {} }

"connections": [

["esp:TX0", "$serialMonitor:RX", ”", []],

["esp:RX0", "$serialMonitor:TX", "", []],

["esp:VIN", "u1trasonic1:VCC", "red", ["h-31.67", "v-176.8", "h152", "v163.33"]

["esp:D18", "ultrasonic1:ECHO", "green", ["h11.37", "v64.67", "h121.33"]],

["esp:D5", "ultrasonic1:TRIG", "green", ["h16.7", "v45.07", "h4"]],

["esp:GND.1", "ultrasonic 1:GND", "black", ["h8.7", "v14.7", "h138.67"]]