**ASSIGNMENT 4**

|  |  |
| --- | --- |
| STUDENT NAME: | A.SHOBIKA SRI |
| STUDENT ROLL NUMBER: | 142219106095 |

**Question :**

**Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less**

**than 100 cm send an "alert" to the IBM cloud and display in the device recent events.**

**Solution :**

#include <WiFi.h>

#include <PubSubClient.h>

#define ORG "486ral"

#define DEVICE\_TYPE "IOT"

#define DEVICE\_ID "id07"

#define TOKEN "123456789"

#define trigpin 5

#define echopin 18

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);

long duration;

float dist;

void setup()

{

Serial.begin(9900);

pinMode(trigpin, OUTPUT);

pinMode(echopin, INPUT);

wifiConnect();

mqttConnect();

}

void loop() {

publishData();

delay(500);

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();

}

}

void publishData()

{

digitalWrite(trigpin,LOW);

digitalWrite(trigpin,HIGH);

delayMicroseconds(10);

digitalWrite(trigpin,LOW);

duration=pulseIn(echopin,HIGH);

dist=duration\*0.034 /2;

if(dist<100)

{

String payload = "{\"Distance\":";

payload += dist;

payload += ",";

payload += "\"Status\":";

payload += "\"Alert\"}";

Serial.print("\n");

Serial.print("Sending payload: ");

Serial.println(payload);

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

{

Serial.println("Publish OK");

}

}

if(dist>100)

{

String payload = "{\"Distance\":";

payload += dist;

payload += ",";

payload += "\"Status\":";

payload += "\"Normal\"}";

Serial.print("\n");

Serial.print("Sending payload: ");

Serial.println(payload);

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

{

Serial.println("Publish OK");

}

else

{

Serial.println("Publish FAILED");

}

}

}

