Assignment 4

Student Name	R.SUBASHINI
Student Roll No	142219106107

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.

Program:

```
#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);
```

```cpp
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(dist100)

{ 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");

}

}

}
```
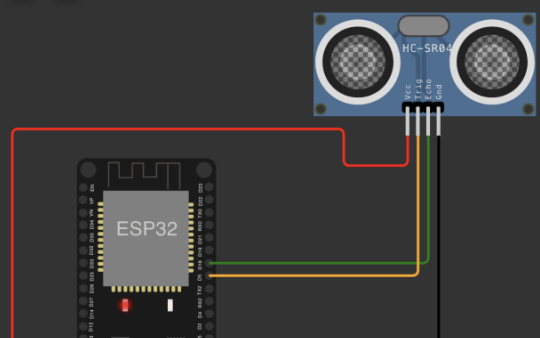
```
1    #include <WiFi.h>
2    #include <PubSubClient.h>
3
4
5    #define ORG "486ral"
6    #define DEVICE_TYPE "IOT"
7    #define DEVICE_ID "id07"
8    #define TOKEN "123456789"
9    #define trigpin 5
10   #define echopin 18
11
12
13
14   char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
15   char publishTopic[] = "iot-2/evt/data/fmt/json";
16   char authMethod[] = "use-token-auth";
17   char token[] = TOKEN;
18   char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19
20
21
22
23   WiFiClient wifiClient;
24   PubSubClient client(server, 1883, wifiClient);
25
26
27
28
29   long duration;
30   float dist;
31
32   void setup()
33   {
34       Serial.begin(9900);
35
```

Sending payload: {"Distance":29.99,"Status":"Alert"}
Publish OK

Sending payload: {"Distance":29.99,"Status":"Alert"}
Publish OK

Sending payload: {"Distance":29.99,"Status":"

Back

Device Drilldown - id07

Connection Information

Recent Events

State

Device Information

Metadata

Diagnostics

Connection Logs

Device Actions

Recent Events

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

Event	Value	Format	Last Received
data	{"Distance":29.99,"Status":"Alert"}	json	a few seconds ago
data	{"Distance":29.99,"Status":"Alert"}	json	a few seconds ago
data	{"Distance":29.99,"Status":"Alert"}	json	a few seconds ago
data	{"Distance":29.99,"Status":"Alert"}	json	a few seconds ago
data	{"Distance":29.99,"Status":"Alert"}	json	a few seconds ago