<div align="center">**ASSIGNMENT-4**</div>

Assignment Date	29 october 2022
Student Name	Ashokkumar M
Student Roll Number	711619104004
Maximum Marks	2 Marks

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.Upload document with wokwi share link and images of ibm cloud.

SOLUTION:

  PROGRAM:

```
#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "ik6mgw"

#define DEVICE_TYPE "assign4"

#define DEVICE_ID "928451"

#define TOKEN "12345678"

#define speed 0.034

#define led 14

char server[] = ORG
".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-
2/evt/Data/fmt/json";

char topic[] = "iot-2/cmd/home/fmt/String";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE
":" DEVICE_ID;

PubSubClient client(server, 1883,
wifiClient);

void publishData();
```

```cpp
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();
  }
```

```cpp
}

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(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){
    String payload = "{\"Normal
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");
    }


  }
    if(dist>101 && dist<111){
    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())) {

      Serial.println("Warning crosses 110cm
-- it automaticaly of the loop");

      digitalWrite(led,HIGH);

    }else {

      Serial.println("Publish 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++){

    dist += (char)payload[i];

  }

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

  if(data3=="lighton"){

    Serial.println(data3);

    digitalWrite(led,HIGH);

  }

  data3="";

}
```
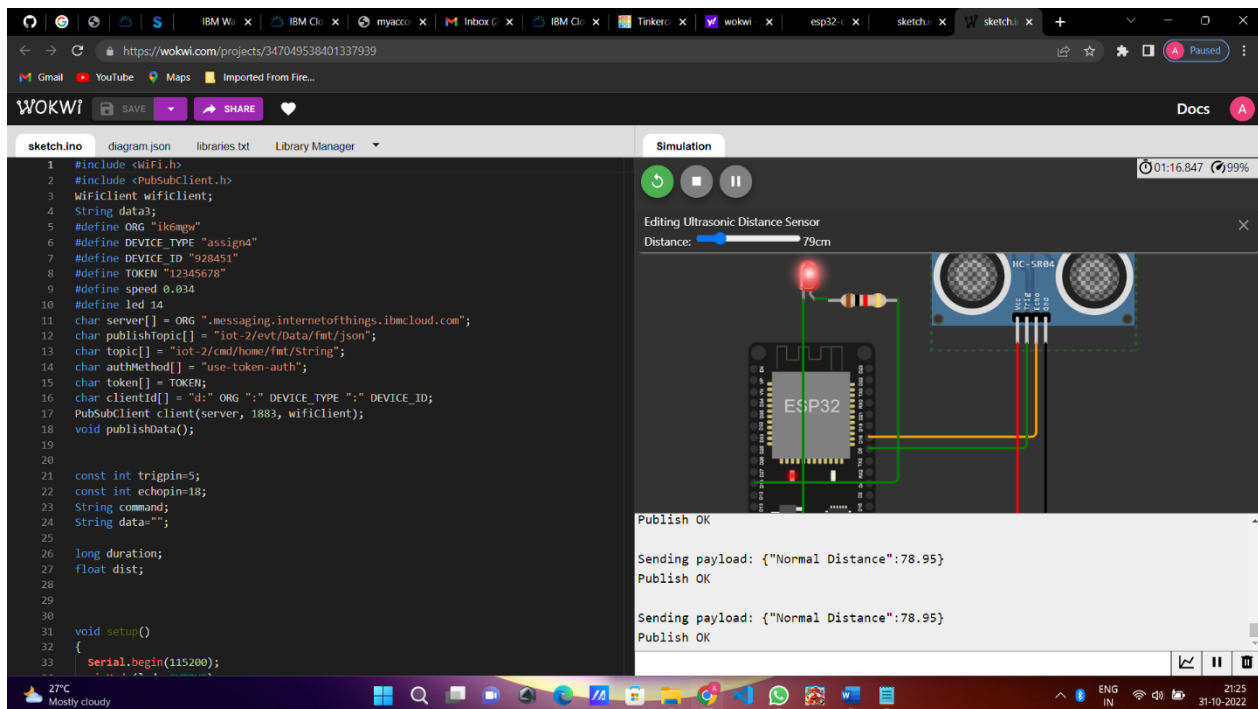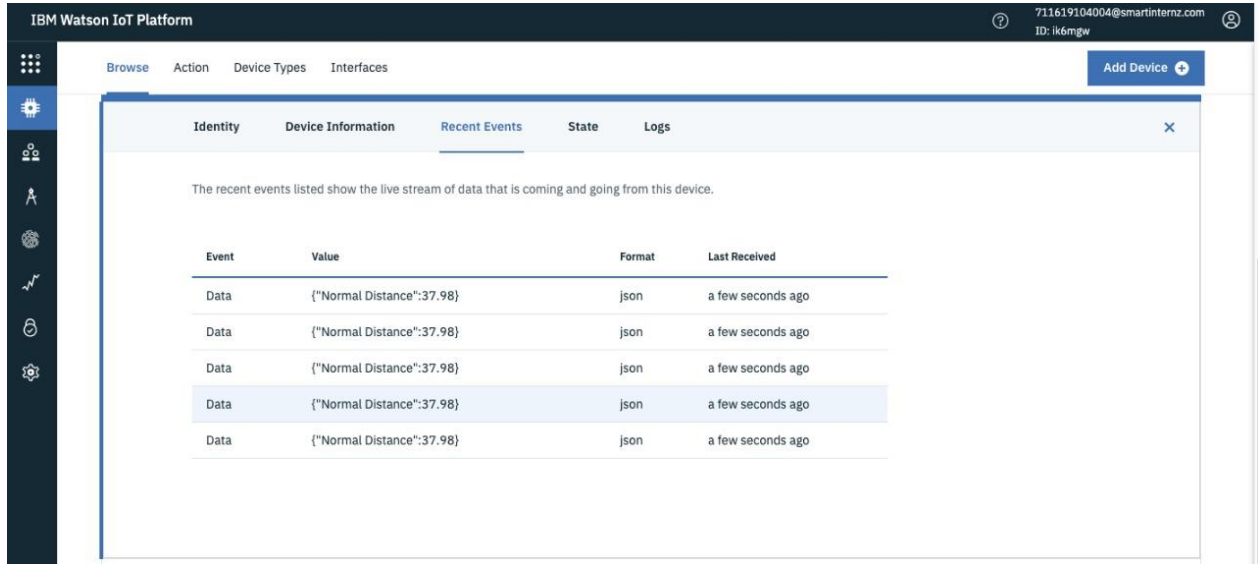
SIMULATION SCREENSHOTS:

```
1   #include <WiFi.h>
2   #include <PubSubClient.h>
3   WiFiClient wifiClient;
4   String data3;
5   #define ORG "ik6mgw"
6   #define DEVICE_TYPE "assign4"
7   #define DEVICE_ID "928451"
8   #define TOKEN "12345678"
9   #define speed 0.034
10  #define led 14
11  char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12  char publishTopic[] = "iot-2/evt/Data/fmt/json";
13  char topic[] = "iot-2/cmd/home/fmt/String";
14  char authMethod[] = "use-token-auth";
15  char token[] = TOKEN;
16  char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17  PubSubClient client(server, 1883, wifiClient);
18  void publishData();
19
20
21  const int trigpin=5;
22  const int echopin=18;
23  String command;
24  String data="";
25
26  long duration;
27  float dist;
28
29
30
31  void setup()
32  {
33    Serial.begin(115200);
```

Editing Ultrasonic Distance Sensor

Distance: 79cm

HC-SR04

ESP32

Publish OK

Sending payload: {"Normal Distance":78.95}
Publish OK

Sending payload: {"Normal Distance":78.95}
Publish OK

WOKWI LINK:

https://wokwi.com/projects/347049538401337939