

# ASSIGNMENT 4

- Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.
- Upload document with wokwi share link and images of IBM cloud

## PROGRAM :

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "y95bny"
#define DEVICE_TYPE "NODEMCU"
#define DEVICE_ID "1234"
#define TOKEN "12345678"
#define speed 0.034
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();
const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration;
int dist;
void setup()
{
```

```

Serial.begin(115200);
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(30);
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(1000);
}
initManagedDevice();
Serial.println();
}
}

```

```

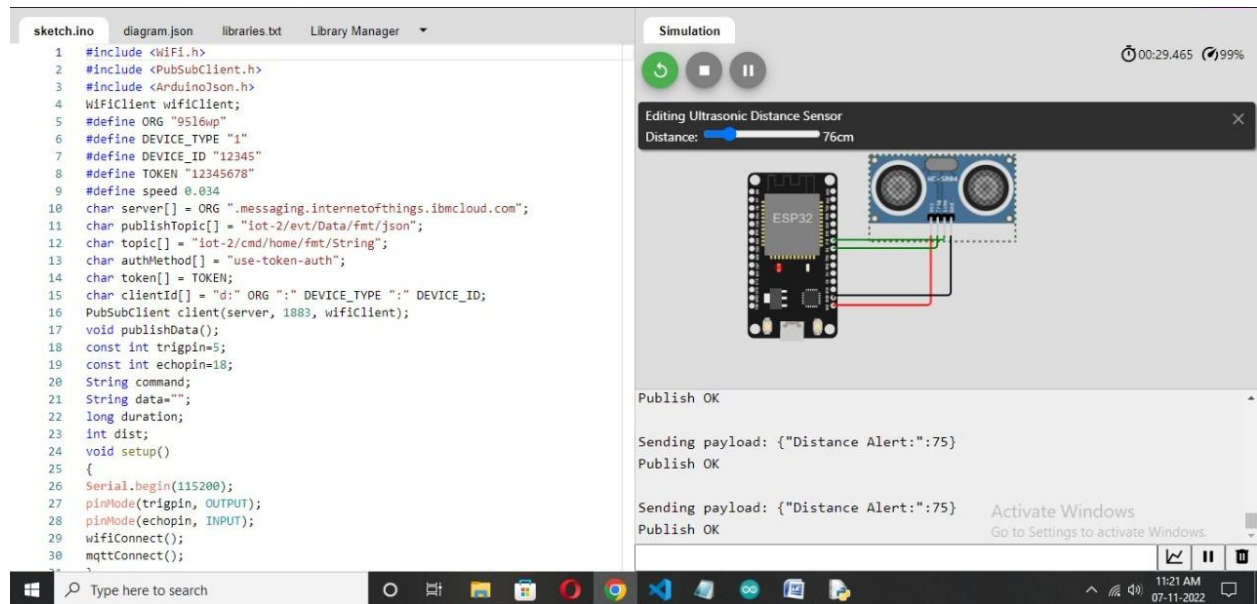
}
void initManagedDevice()
{
    if (client.subscribe(topic))
    {
        Serial.println(client.subscribe(topic));
        Serial.println("subscribe to cmd OK");
    }
    else
    {
        Serial.println("subscribe to cmd FAILED");
    }
}
void publishData()
{
    digitalWrite(trigpin, LOW);
    digitalWrite(trigpin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin, LOW);
    duration=pulseIn(echopin, HIGH);
    dist=duration*speed/2;
    if(dist<100)
    {
        DynamicJsonDocument doc(1024);
        String payload;
        doc["Distance Alert:"]=dist;
        serializeJson(doc, payload);
        delay(30);
        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");
        }
    }
}
}

```

WOKWI LINK:

<https://wokwi.com/projects/347594971923087956>

Wokwi\_output :



IBM\_watson\_output:

The screenshot shows a web application interface for managing devices. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A blue header bar displays the device ID '12345', status 'Connected', a count '1', the name 'Device', and the timestamp '6 Nov 2022 16:16'. Below this, a tabbed interface shows 'Identity', 'Device Information', 'Recent Events' (selected), 'State', and 'Logs'. A message states: 'The recent events listed show the live stream of data that is coming and going from this device.'

| Event   | Value                 | Format | Last Received     |
|---------|-----------------------|--------|-------------------|
| event_1 | {}                    | json   | a few seconds ago |
| event_1 | {}                    | json   | a few seconds ago |
| Data    | {"Distance Alert":75} | json   | a few seconds ago |
| event_1 | {}                    | json   | a few seconds ago |
| Data    | {"Distance Alert":75} | json   | a few seconds ago |

At the bottom of the browser window, a Windows taskbar is visible with the search bar 'Type here to search' and various application icons. A Windows watermark 'Activate Windows Go to Settings to activate Windows.' is present in the bottom right corner.

THANKING YOU !!!