

Assignment 4

Name	Abimanyu S
Team ID	PNT2022TMID38327
Project Name	IOT Based smart Crop production system for Agriculture

Question:

Write a 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

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
WiFiClient wifiClient;
String data3;
#define ORG "z84rrz"
#define DEVICE_TYPE "Abimanyu"
#define DEVICE_ID "Assignment-4"
#define TOKEN "12345678"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/abi/fmt/json"; char
topic[] = "iot-2/cmd/status/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);

const int trigpin=19;
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();
}
}
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");
}
}

```

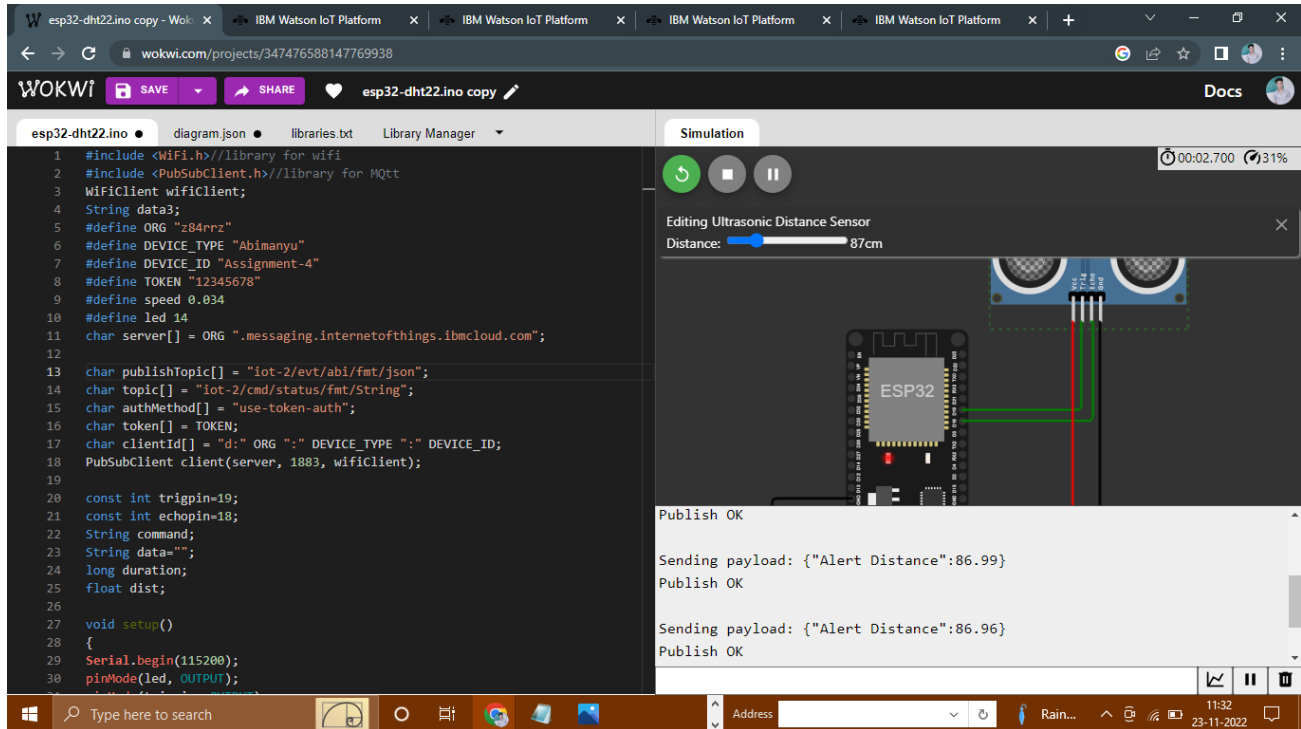
```

}
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 = "{\"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("Publish OK");
        }
    }
    if(dist>100){
        String payload = "{\"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");
        }
    }
    else
    {
        Serial.println("Publish FAILED");
    }
}
}
}

```

Output: 1. When distance less than 100 cm



IBM RECENT EVENTS:

The screenshot shows the IBM Watson IoT Platform dashboard. The user is logged in as 'ebiaravi1812@gmail.com' with ID 'z84rrz'. The dashboard displays a table of recent events for a device named 'Assignment-4' (Abimanyu). The table has columns for Event, Value, Format, and Last Received. The events show a stream of 'Alert Distance' data points in JSON format, all received 'a few seconds ago'.

Event	Value	Format	Last Received
abi	{"Alert Distance":86.96}	json	a few seconds ago
abi	{"Alert Distance":86.99}	json	a few seconds ago
abi	{"Alert Distance":86.96}	json	a few seconds ago
abi	{"Alert Distance":86.96}	json	a few seconds ago
abi	{"Alert Distance":86.96}	json	a few seconds ago

2. When distance greater than 100 cm

The screenshot shows the Wokwi IDE interface. On the left, the Arduino code for the ESP32 is displayed. The code includes headers for WiFi and MQTT, defines the device type as 'Abimanyu', and sets up the MQTT client. It also defines a trigger pin (19) and an echo pin (18). The setup function initializes the serial port and the ultrasonic sensor. The main loop sends the distance data to the MQTT broker.

```

1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "z84rrz"
6 #define DEVICE_TYPE "Abimanyu"
7 #define DEVICE_ID "Assignment-4"
8 #define TOKEN "12345678"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12
13 char publishTopic[] = "iot-2/evt/abi/fmt/json";
14 char topic[] = "iot-2/cmd/status/fmt/String";
15 char authMethod[] = "use-token-auth";
16 char token[] = TOKEN;
17 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
18 PubSubClient client(server, 1883, wifiClient);
19
20 const int trigpin=19;
21 const int echopin=18;
22 String command;
23 String data="";
24 long duration;
25 float dist;
26
27 void setup()
28 {
29   Serial.begin(115200);
30   pinMode(led, OUTPUT);

```

On the right, the simulation window shows an ESP32 board connected to an Ultrasonic Distance Sensor. The sensor's distance is set to 400cm. The simulation log shows the following output:

```

Publish OK
Sending payload: {"Distance":399.94}
Publish OK
Sending payload: {"Distance":399.96}
Publish OK

```

IBM RECENT EVENTS:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A user profile is visible in the top right corner with the email 'abiaravi1812@gmail.com' and ID 'z84rrz'. The main content area shows a device named 'Abimanyu' with a status of 'Connected'. The 'Recent Events' tab is selected, displaying a table of live data streams. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. The data shows five events, all from 'abi', with values representing distance measurements in JSON format, received a few seconds ago.

Event	Value	Format	Last Received
abi	{"Distance":399.96}	json	a few seconds ago
abi	{"Distance":399.94}	json	a few seconds ago
abi	{"Distance":399.96}	json	a few seconds ago
abi	{"Distance":399.98}	json	a few seconds ago
abi	{"Distance":399.96}	json	a few seconds ago