

Assignment 4

Name	Umamaheswari 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 "jeglvm"
#define DEVICE_TYPE
"umamaheswari" #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/uma/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

The screenshot displays the Wokwi IoT simulator interface. On the left, a code editor shows the following C++ code:

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "jeglvm"
6 #define DEVICE_TYPE "umamaheswari"
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/uma/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);
31 }
```

On the right, the 'Simulation' window shows an ESP32 board connected to an Ultrasonic Distance Sensor. The sensor's distance is displayed as 94cm. Below the sensor, a console output shows two successful publishes of a JSON payload:

```
Publish OK
Sending payload: {"Alert Distance":93.96}
Publish OK
Sending payload: {"Alert Distance":93.96}
Publish OK
```

IBM RECENT EVENTS:

The screenshot shows the IBM Watson IoT Platform dashboard. The main view displays a table of devices. One device, 'Assignment-4', is selected and its details are shown in a modal window. The modal window has tabs for Identity, Device Information, Recent Events, State, and Logs. The 'Recent Events' tab is active, showing a list of events with columns: Event, Value, Format, and Last Received. The events are all of type 'uma' and contain the value '{"Alert Distance":93.96}' in JSON format, received 'a few seconds ago'.

Event	Value	Format	Last Received
uma	{"Alert Distance":93.96}	json	a few seconds ago
uma	{"Alert Distance":93.96}	json	a few seconds ago
uma	{"Alert Distance":93.96}	json	a few seconds ago
uma	{"Alert Distance":93.96}	json	a few seconds ago
uma	{"Alert Distance":93.96}	json	a few seconds ago

2. When distance greater than 100 cm

The screenshot shows the Wokwi IDE interface. On the left, the Arduino sketch for 'esp32-dht22.ino' is displayed. The sketch includes libraries for WiFi and MQTT, defines device information (ORG, DEVICE_TYPE, DEVICE_ID, TOKEN), and sets up an MQTT client. It also defines pins for a trigpin and echopin, and sets up an LED. The main loop publishes data to an MQTT topic. On the right, the simulation window shows a 3D model of the ESP32 board connected to an HC-SR04 ultrasonic sensor. The simulation is running, and the console output shows the following messages:

```

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

IBM Watson IoT Platform

412619106014@smartinernz.com
ID: km27sq

Browse

Action

Device Types

Interfaces

Search by Device ID

Device Simulator

Add Device

	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
>	03062001	Disconnected	NodeMCU	Device	Oct 26, 2022 2:08 PM	
▼	Assignment_4	Connected	Ruibashree	Device	Nov 4, 2022 1:36 PM	→ ...

Identity

Device Information

Recent Events

State

Logs

×

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

Event	Value	Format	Last Received
rubashree	["Alert Distance":96.93]	json	a few seconds ago
rubashree	["Alert Distance":96.93]	json	a few seconds ago
rubashree	["Alert Distance":96.93]	json	a few seconds ago
rubashree	["Alert Distance":96.97]	json	a few seconds ago
rubashree	["Alert Distance":96.93]	json	a few seconds ago

Activate Windows

Go to Settings to activate Windows.