

Assignment -4

Python Programming

Student Name	Thaila Uma A
Student Roll Number	720319106024

Question-1:

Solution:

Link : <https://wokwi.com/projects/348220398448411220>

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;

#define trigpin    18
#define echopin    5

String data3;

#define ORG "mx4qhh"//IBM ORGANITION ID
#define DEVICE_TYPE "River"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "Water"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "rTjnsdE5@eVgnVJwW4"

#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();

String command;
String data="";
long duration;
```

```

float dist;

void setup()
{
    Serial.begin(115200);
    wifiConnect();
    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);

    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 loop()
{
    int pulseWidth = 0;
    digitalWrite(trigpin, HIGH);
    delayMicroseconds(100);
    digitalWrite(trigpin, LOW);
    pulseWidth = pulseIn(echopin, HIGH);
    Serial.print("AlertDistance: ");
    Serial.println(pulseWidth/58);

    publishData();
    if (!client.loop()) {
        mqttConnect();
    }
}

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 automatically of the loop");
        }
    }
}

```

```

    }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);
    }
    data3="";
}
}

```

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area is titled 'Recent Events' and shows a table of data events. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. The events listed are all 'Data' events with values like '["Normal Distance":52]' and '["Normal Distance":51.99]', all in 'json' format, and received 'a few seconds ago'. The bottom of the dashboard shows '0 Simulations running' and a status bar with system icons and the date '13/11/2022'.

Event	Value	Format	Last Received
Data	["Normal Distance":52]	json	a few seconds ago
Data	["Normal Distance":51.99]	json	a few seconds ago
Data	["Normal Distance":52.02]	json	a few seconds ago
Data	["Normal Distance":52.02]	json	a few seconds ago
Data	["Normal Distance":51.99]	json	a few seconds ago

Output : <https://mx4qhh.internetofthings.ibmcloud.com/dashboard/devices/browse>

IBM Cloud x W esp32-dht22.ino - Wokwi Arduino x (9) WhatsApp x +

wokwi.com/projects/322410731508073042

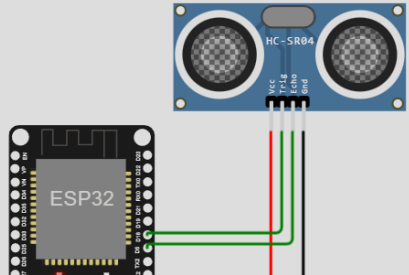
WOKWI SAVE SHARE esp32-dht22.ino by urish Docs

esp32-dht22.ino • diagram.json • libraries.txt • Library Manager

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4
5 #define trigpin 18
6 #define echopin 5
7
8 String data3;
9
10 #define ORG "mx4qhh"//IBM ORGANITION ID
11 #define DEVICE_TYPE "River"//Device type mentioned in ibm watson IOT Platform
12 #define DEVICE_ID "Water"//Device ID mentioned in ibm watson IOT Platform
13 #define TOKEN "rTjnsdES@eVgnVJwM4"
14
15 #define speed 0.034
16 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
17 char publishTopic[] = "iot-2/evt/Data/fmt/json";
18 char topic[] = "iot-2/cmd/home/fmt/String";
19 char authMethod[] = "use-token-auth";
20 char token[] = TOKEN;
21 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
22 PubSubClient client(server, 1883, wifiClient);
23 void publishData();
24
25 String command;
26 String data="";
27 long duration;
28 float dist;
29
30 void setup()
```

Simulation

00:07.263 99%



Sending payload: {"Normal Distance":51.99}
Publish OK
AlertDistance: 52

Sending payload: {"Normal Distance":51.99}
Publish OK
AlertDistance: 52

Type here to search

19:05
13/11/2022