

#### Assignment-4

Team ID	PNT2022TMID06114
Project	Industry Specific intelligent fire management system
Date	17 Nov 2022

#### Code:

```
#include <WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

WiFiClient wifiClient;

#define ORG "wa9fzx"

#define DEVICE_TYPE "NodeMCUESP32"

#define DEVICE_ID "123456"

#define TOKEN "123456789"

#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() {
  ublishData();
  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(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["AlertDistance:"]=dist;
serializeJson(doc, payload);
delay(3000);
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/348595243786961491>

wokwi.com/projects

WOKWI SAVE SHARE sketch.ino copy Docs

sketch.ino diagram.json libraries.txt Library Manager Simulation

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4 WiFiClient wifiClient;
5 #define ORG "wa9fzx"
6 #define DEVICE_TYPE "NodeMCUESP32"
7 #define DEVICE_ID "123456"
8 #define TOKEN "123456789"
9 #define speed 0.034
10 char server[] = ORG ".messaging.internetofthings";
11 char publishTopic[] = "iot-2/evt/Data/fmt/json";
12 char topic[] = "iot-2/cmd/home/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
16 PubSubClient client(server, 1883, wifiClient);
17 void publishData();
18 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 long duration;
23 int dist;
24 void setup()
25 {
26   Serial.begin(115200);
27   pinMode(trigpin, OUTPUT);
28   pinMode(echopin, INPUT);
29   wifiConnect();
30   mqttConnect();
31 }
32 void loop() {
33   publishData();
34   delay(500);
35   if (!client.connected()) {
36     mqttConnect();
37   }
38 }
39 void wifiConnect() {
40   Serial.print("Connecting to "); Serial.print(WiFi.hostname());
41   WiFi.begin("Wokwi-GUEST", "", 6);
42   while (WiFi.status() != WL_CONNECTED) {
43     delay(500);
44     Serial.print(".");
45   }
46   Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
47 }
48 void mqttConnect() {
49   if (!client.connected()) {
50     Serial.print("Reconnecting MQTT client to "); Serial.println(server);
51     while (!client.connect(clientId, authMethod, token)) {
52       Serial.print(".");
53       delay(1000);
54     }
55     initManagedDevice();
56     Serial.println();
57   }
58 }
59 void initManagedDevice() {
60   if (client.subscribe(topic)) {
61     Serial.println(client.subscribe(topic));
62     Serial.println("subscribe to cmd OK");
63   } else {
64     Serial.println("subscribe to cmd FAILED");
65   }
66 }
67 void publishData()
68 {
69   digitalWrite(trigpin,LOW);
70   digitalWrite(trigpin,HIGH);
71   delayMicroseconds(10);
72   digitalWrite(trigpin,LOW);
73   duration=pulseIn(echopin,HIGH);
74   dist=duration*speed/2;
75   if(dist<100){
76     DynamicJsonDocument doc(1024);
77     String payload;
78     doc["AlertDistance:"]=dist;
79     serializeJson(doc, payload);
80     delay(3000);
81     Serial.print("\n");
82     Serial.print("Sending payload: ");
83     Serial.println(payload);
84     if (client.publish(publishTopic, (char*) payload)) {
85       Serial.println("Publish OK");
86     } else {
87       Serial.println("Publish FAILED");
88     }
89   }
90 }
```

