

DEVELOP A PYTHON SCRIPT

Date	03 November
Team ID	PNT2022TMID52636
Project Name	Signs with Smart Connectivity for Better Road Safety
Maximum Marks	4 Marks

```
#include <WiFi.h>
```

```
#include <PubSubClient.h>
```

```
#define ORG "4i2rfo"
```

```
#define DEVICE_TYPE "Traffic_Analyser"
```

```
#define DEVICE_ID "Ultrasonic_Sensor"
```

```
#define TOKEN "12345678"
```

```
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
```

```
char publishTopic[] = "iot-2/evt/UltraSonic_Sensor/fmt/json";
```

```
char subscribetopic[] = "iot-2/cmd/command/fmt/String";
```

```
char authMethod[] = "use-token-auth";
```

```
char token[] = TOKEN;
```

```
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
```

```
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
```

```
WiFiClient wifiClient; // creating the instance for wificlient
```

```
PubSubClient client(server, 1883, wifiClient); //calling the predefined client id by passing  
parameter like server id,portand wificredential
```

```
const int trigPin = 18;
```

```
const int echoPin = 5;
```

```
long duration;
```

```
float distanceCm;
```

```
String data3;
```

```
void setup()// configureing the ESP32
```

```
{  
  Serial.begin(115200);  
  pinMode(trigPin, OUTPUT);  
  pinMode(echoPin, INPUT);  
  wificonnect();  
  mqttconnect();  
}
```

```
void loop()// Recursive Function
```

```
{  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  
  duration = pulseIn(echoPin, HIGH);  
  
  // Calculate the distance  
  distanceCm = duration * 0.034/2;  
  
  // Prints the distance in the Serial Monitor  
  Serial.print("Distance (cm): ");  
  Serial.println(distanceCm);  
  
  delay(1000);  
  PublishData(distanceCm);
```

```

    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

/*.....retrieving to Cloud.....*/

void PublishData(float dist) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSon to update the data to ibm cloud
    */
    String payload = "{\"Distance\":\"";
    payload += dist;
    payload += "}";

    Serial.print("Sending payload: ");
    Serial.println(payload);

    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print
        publish ok in Serial monitor or else it will print publish failed
    } else {
        Serial.println("Publish failed");
    }
}

```

```
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    Serial.println();
  }
}

void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6); //passing the wifi credentials to establish the
connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```



```
DEVICE_ID = "OpenWeather"
```

```
server = ORG + ".messaging.internetofthings.ibmcloud.com";
```

```
pubTopic1 = "iot-2/evt/Weather/fmt/json";
```

```
pubTopic2 = "iot-2/evt/Visibility/fmt/json";
```

```
authMethod = "use-token-auth";
```

```
token = TOKEN;
```

```
clientId = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;
```

```
mqttc = mqtt.Client(client_id=clientId)
```

```
mqttc.username_pw_set(authMethod, token)
```

```
mqttc.connect(server, 1883, 60)
```

```
while True:
```

```
    try:
```

```
        mqttc.publish(pubTopic1, weather_main)
```

```
        mqttc.publish(pubTopic2, visibility)
```

```
        print ("Published")
```

```
    except RuntimeError as error:
```

```
        # Errors happen fairly often, DHT's are hard to read, just keep going
```

```
        print(error.args[0])
```

```
        time.sleep(2.0)
```

```
        continue
```

```
    except Exception as error:
```

```
        dhtDevice.exit()
```

```
        raise error
```

```
time.sleep(5.0)
```

```
mqttc.loop_forever()
```

OpenWeather

Connected

Weather_Analyser

Device

Nov 19, 2022 2:10 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
Visibility	5000	json	a few seconds ago
Weather	{"type":"Buffer","data":[72,97,122,101]}	json	a few seconds ago
Visibility	5000	json	a few seconds ago
Weather	{"type":"Buffer","data":[72,97,122,101]}	json	a few seconds ago
Visibility	5000	json	a few seconds ago