

CODE :

```
#include <WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

WiFiClient wifiClient;

#define ORG "1la4d3"

#define DEVICE_TYPE "OVIYA117"

#define DEVICE_ID "oviyasri117"

#define TOKEN "OVIYAOVIYA117"

#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() {
  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(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["Distance Alert:"]=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/347479301071831634>

The screenshot shows the Wokwi IDE interface. On the left, the sketch.ino file contains the following code:

```
1 #include <Wifi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4 WiFiClient wifiClient;
5 #define ORG "1la4d3"
6 #define DEVICE_TYPE "OVIYA117"
7 #define DEVICE_ID "oviya117"
8 #define TOKEN "OVIYA0VIYA117"
9 #define speed 0.034
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
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 }
```

The simulation window on the right shows a breadboard with an ESP32 microcontroller and an HC-SR04 ultrasonic sensor. The console output displays the following messages:

```
Publish OK
Sending payload: {"Distance Alert":33}
Publish OK
Sending payload: {"Distance Alert":33}
Publish OK
```

The screenshot shows the IBM Watson IoT Platform dashboard. The 'Recent Events' tab is selected, displaying a table of events:

Event	Value	Format	Last Received
Data	{"Distance Alert":33}	json	a few seconds ago
Data	{"Distance Alert":33}	json	a few seconds ago
Data	{"Distance Alert":33}	json	a few seconds ago
Data	{"Distance Alert":34}	json	a few seconds ago
Data	{"Distance Alert":33}	json	a few seconds ago

A notification at the bottom right states: "1 Simulation running".