

Publish Data To The IBM Cloud

Team ID	PNT2022TMID47715
Project Name	Project - Smart Waste Management System For Metropolitan Cities

IBM Watson IoT Platform

IBM Watson IoT Platform

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
1234	Disconnected	Assignment	Device	Nov 7, 2022 7:25 PM	
1234	Disconnected	raspberrypi	Device	Nov 3, 2022 8:17 PM	
123456	Disconnected	123	Device	Nov 4, 2022 10:00 PM	

Items per page 50 | 1-3 of 3 items

0 Simulations running

WOKWI

sketch.ino

```
24 void setup() {
25   Serial.begin(115200);
26   pinMode(trigPin, OUTPUT);
27   pinMode(echoPin, INPUT);
28   wifiConnect();
29   mqttConnect();
30 }
31 void loop() {
32   {
33     digitalWrite(trigPin, LOW);
34     delayMicroseconds(2);
35     digitalWrite(trigPin, HIGH);
36     delayMicroseconds(10);
37     digitalWrite(trigPin, LOW);
38     duration = pulseIn(echoPin, HIGH);
39     distance = duration * SOUND_SPEED/2;
40     Serial.print("Distance (cm): ");
41     Serial.println(distance);
42     if(distance<100) {
43       {
44         Serial.println("ALERT!!");
45         delay(1000);
46         PublishData(distance);
47         delay(1000);
48         if (!client.loop()) {
49           mqttConnect();
50         }
51       }
52       delay(1000);
53     }
54     void PublishData(float dist) {
55       mqttConnect();
56       String payload = "{\"Distance\":\"";
57       payload += dist;
```

Simulation

01:48.039 99%

Editing Ultrasonic Distance Sensor

Distance: 87cm

Publish ok

Distance (cm): 86.96

ALERT!!

Sending payload: {"Distance":86.96,"ALERT!!":"Distance less than 100cms"}

Publish ok

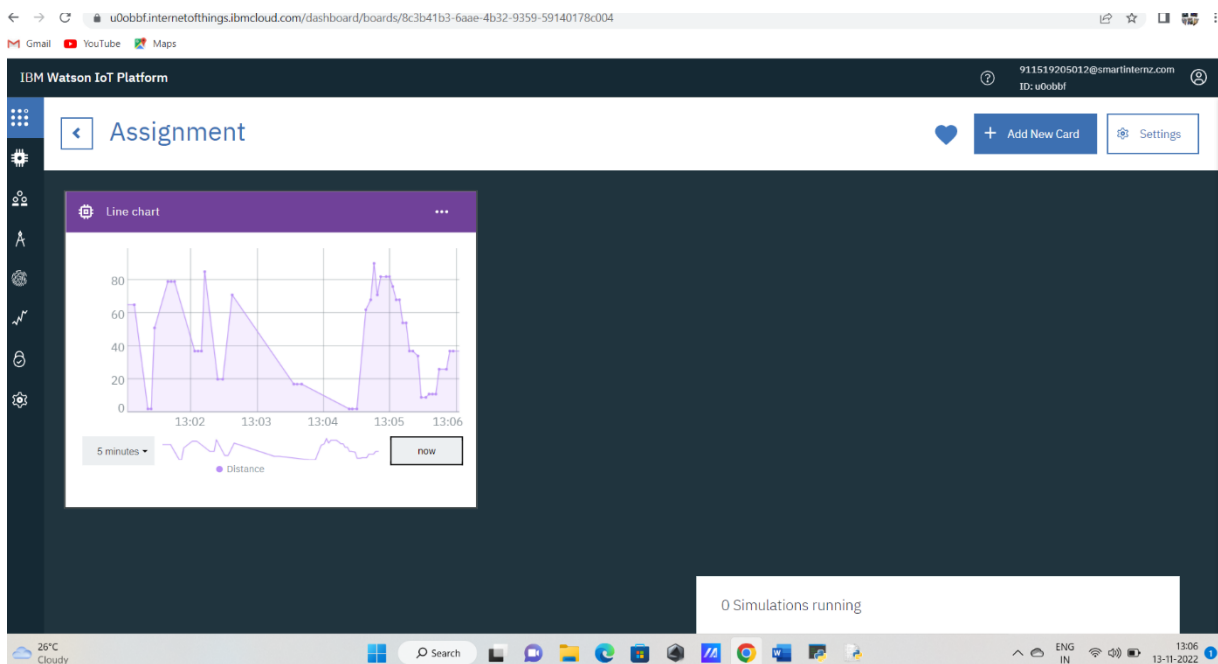
Distance (cm): 86.96

ALERT!!

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area displays a table of devices. The selected device has ID '1234', status 'Connected', and device type 'Assignment'. Below the table, the 'Recent Events' tab is active, showing a list of events. The events are as follows:

Event	Value	Format	Last Received
Data	{"Distance":84.95,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":84.95,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":84.95,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":70.96,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":70.96,"ALERT!!":"Distance less than ...	json	a few seconds ago

At the bottom of the dashboard, it indicates '0 Simulations running'.



Code

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "u0obbf"//IBM ORGANITION ID
```

```

#define DEVICE_TYPE "Assignment"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribtopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  wificonnect();
  mqttconnect();
}
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * SOUND_SPEED/2;
  Serial.print("Distance (cm): ");
  Serial.println(distance);
  if(distance<100)
  {
    Serial.println("ALERT!!");
    delay(1000);
    PublishData(distance);
    delay(1000);
    if (!client.loop()) {
      mqttconnect();
    }
  }
  delay(1000);
}

```

```

void PublishData(float dist) {
  mqttconnect();
  String payload = "{\"Distance\":\"";
  payload += dist;
  payload += "\",\"ALERT!!\":\"\"Distance less than 100cms\"";
  payload += "\"}";
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");
  } 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);
    }
    initManagedDevice();
    Serial.println();
  }
}

void wificonnect()
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd 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++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: " + data3);
  data3="";
}
```

“Alert Data Publish to IBM Cloud from Wokwi consist with ESP32 and Ultrasonic sensor”