

## ASSIGNMENT 4

Date	2 Nov 22
Name	Gokul M
Team ID	PNT2022TMID38273
Project Name	IOT Based Smart Crop Protection System for Agriculture

### QUESTION :

Write code and connection in wovki for ultrasonic sensor. Whenever distance is less than 100 cms send “alert” to IBM cloud and display in device recent events

### CODE :

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
WiFiClient wifiClient;
String data3;
#define ORG "p08uvj"

#define DEVICE_TYPE "Gokul"
#define DEVICE_ID "Gokul_assignment_4"
#define TOKEN "O&kB_-*lqm6i*WnlH8"

#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Gokul/fmt/json";
char topic[] = "iot-2/cmd/event_1/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);

const int trigpin=5;
const int echopin=18;
String command;
```

```

String data="";

long duration;
float dist;


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

void loop(){
  bool isNearby = dist < 100;
  digitalWrite(led, isNearby);

  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(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");
    }
}

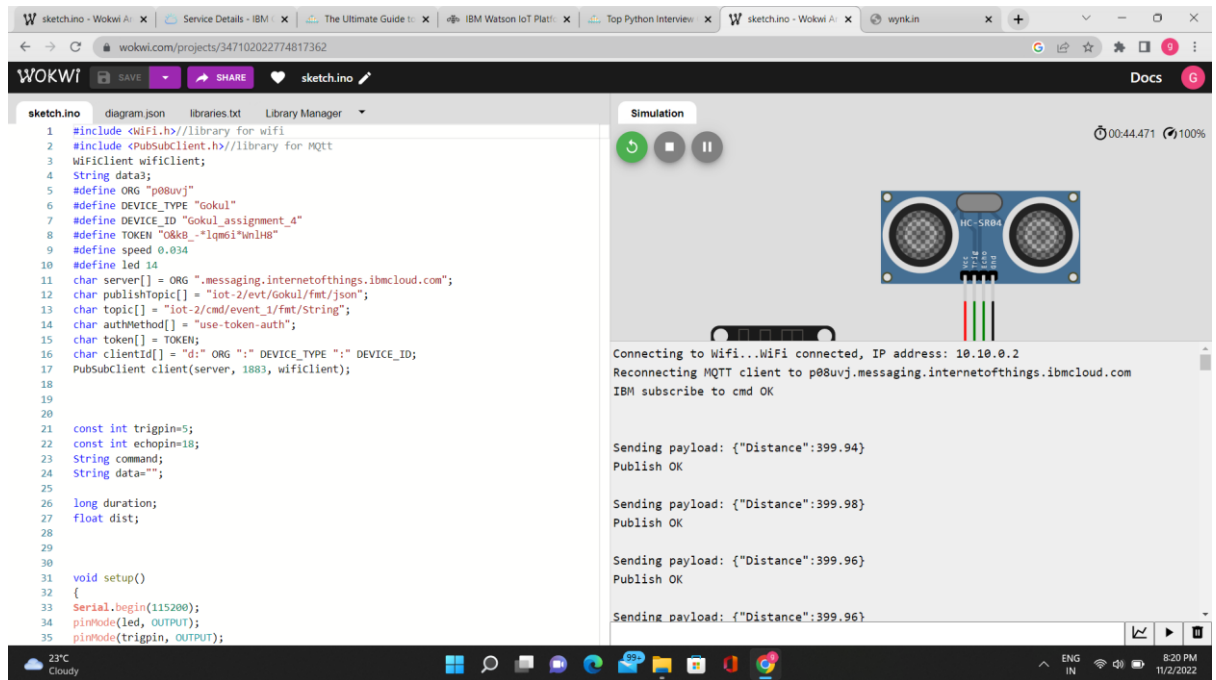
void publishData(){
    digitalWrite(trigpin,LOW);
    digitalWrite(trigpin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin,LOW);
    duration=pulseIn(echopin,HIGH);
    dist=duration*speed/2;
    if(dist<100){
        String payload = "{\"Alert Distance\":";
        payload += dist;
        payload += "}";
        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);
        if (client.publish(publishTopic, (char*) payload.c_str())){
            Serial.println("Publish OK");
        }
    }
    if(dist>100){
        String payload = "{\"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");
        }
    }
    else{
        Serial.println("Publish FAILED");
    }
}
}

```

## OUTPUT :

### 1) When Distance greater than 100 cm



The screenshot shows the Wokwi IoT simulator interface. On the left, the sketch code is displayed, which includes headers for WiFi and MQTT, defines device parameters (ORG, DEVICE\_TYPE, DEVICE\_ID, TOKEN, speed), and sets up an MQTT client. The main loop publishes distance data as JSON payloads. On the right, the simulation output shows the device connecting to WiFi (IP: 10.10.0.2) and the MQTT client, then sending several payloads: {"Distance":399.94}, {"Distance":399.98}, {"Distance":399.96}, {"Distance":399.96}.

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "p08uvj"
6 #define DEVICE_TYPE "Gokul"
7 #define DEVICE_ID "Gokul_assignment_4"
8 #define TOKEN "O&K8_-1qm6iWnIH8"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/gokul/fmt/json";
13 char topic[] = "iot-2/cmd/event_1/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wificlient);
18
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);
34   pinMode(led, OUTPUT);
35   pinMode(trigpin, OUTPUT);
```

Connecting to Wifi...WiFi connected, IP address: 10.10.0.2  
Reconnecting MQTT client to p08uvj.messaging.internetofthings.ibmcloud.com  
IBM subscribe to cmd OK

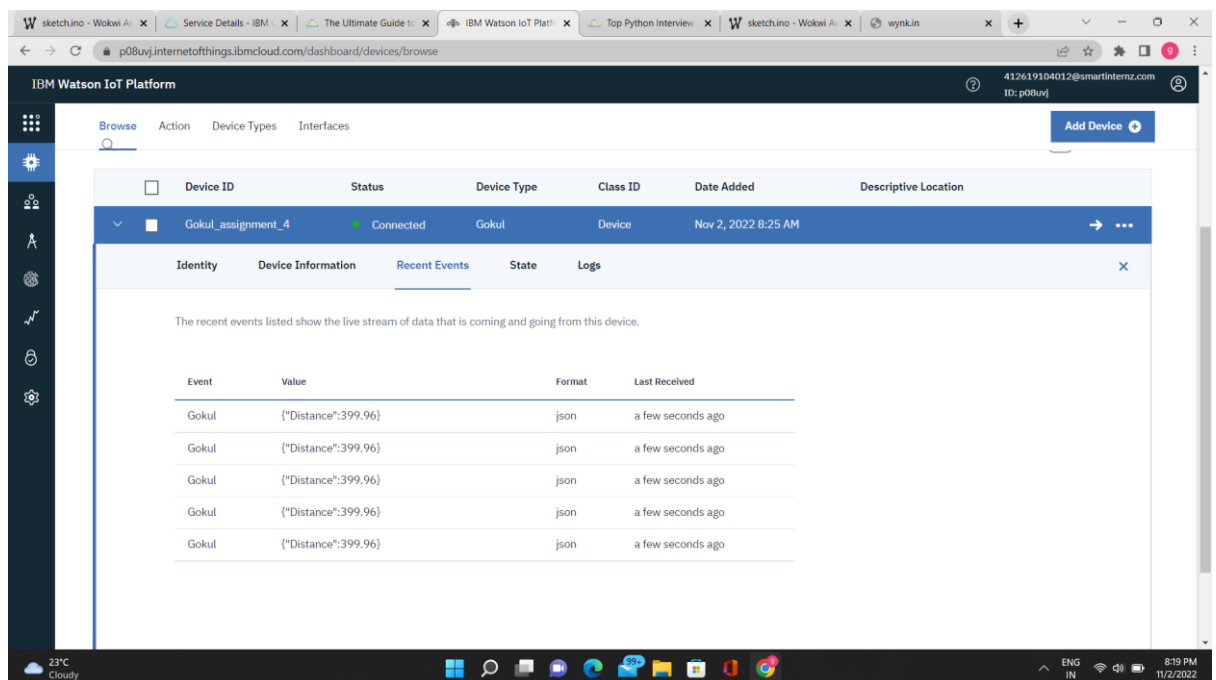
Sending payload: {"Distance":399.94}  
Publish OK

Sending payload: {"Distance":399.98}  
Publish OK

Sending payload: {"Distance":399.96}  
Publish OK

Sending payload: {"Distance":399.96}

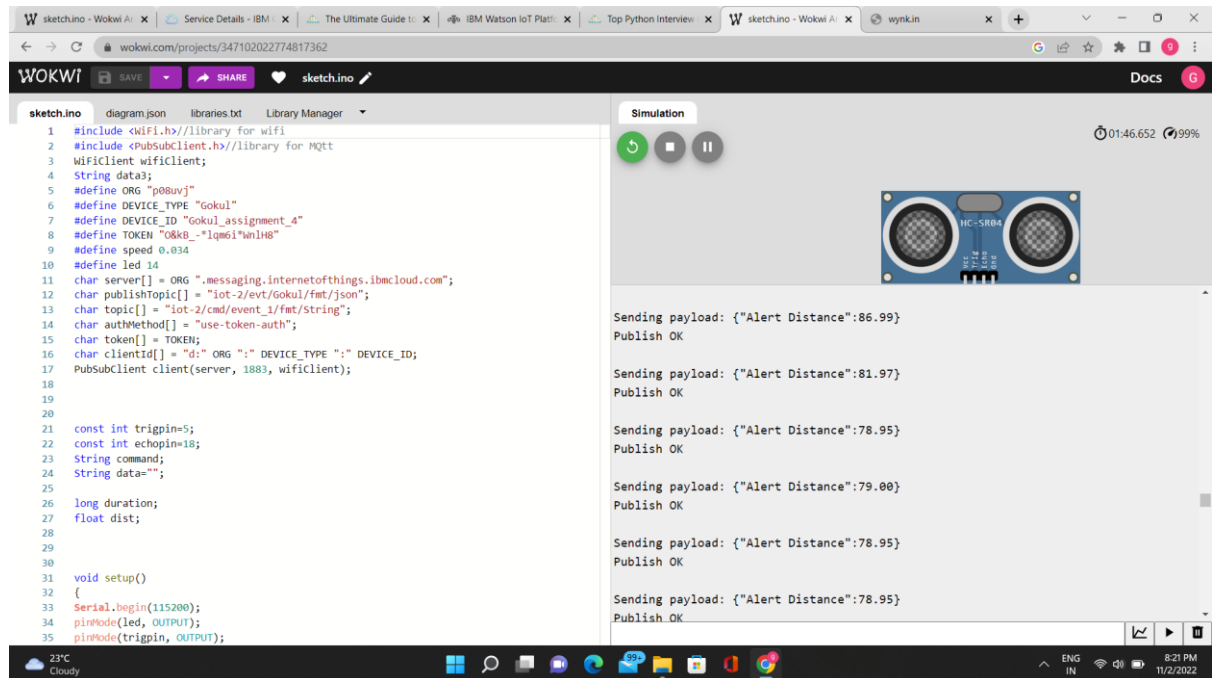
## IBM RECENT EVENTS



The screenshot shows the IBM Watson IoT Platform dashboard. The device 'Gokul\_assignment\_4' is listed as 'Connected'. The 'Recent Events' tab is selected, showing a table of events. The table has columns for Event, Value, Format, and Last Received. The events are all of type 'Gokul' with a value of '{"Distance":399.96}' in 'json' format, received 'a few seconds ago'.

Event	Value	Format	Last Received
Gokul	{"Distance":399.96}	json	a few seconds ago
Gokul	{"Distance":399.96}	json	a few seconds ago
Gokul	{"Distance":399.96}	json	a few seconds ago
Gokul	{"Distance":399.96}	json	a few seconds ago
Gokul	{"Distance":399.96}	json	a few seconds ago

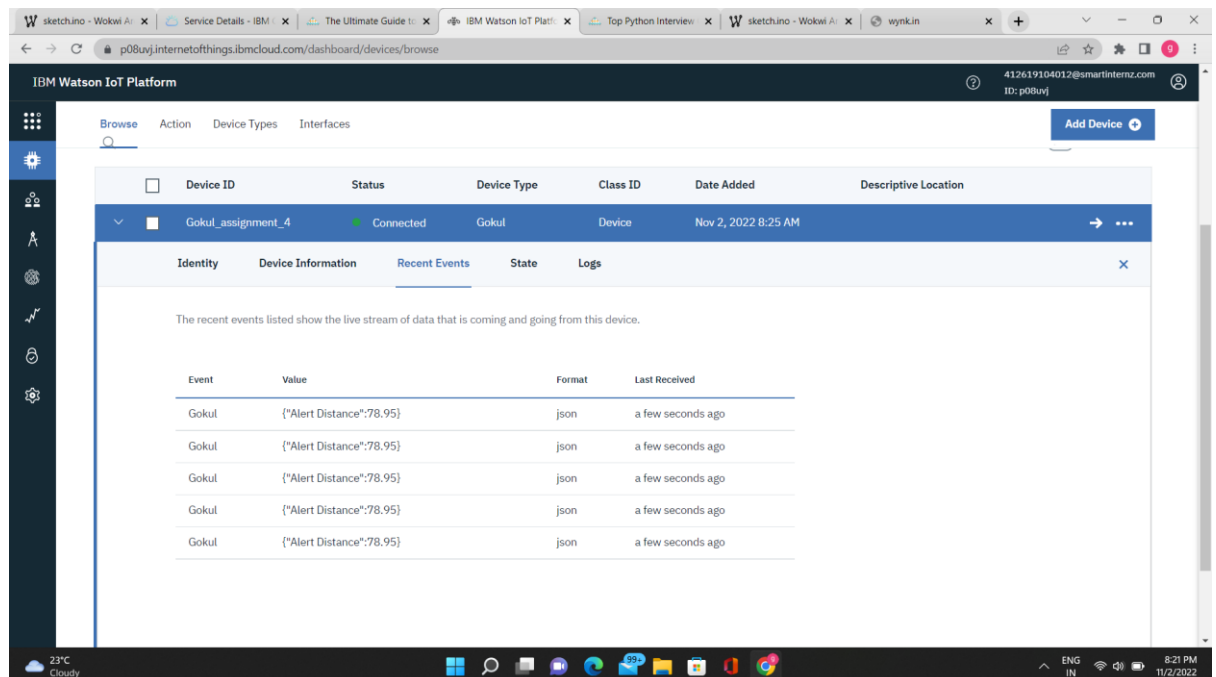
## 2) When distance less than 100



The screenshot shows the Wokwi IoT Platform interface. On the left, the 'sketch.ino' file is open, displaying the Arduino code for a GoKul device. The code includes the necessary libraries, defines the device type and ID, and sets up the MQTT client. The main loop publishes the 'Alert Distance' value to the 'iot-2/cmd/event\_1/fmt/string' topic. On the right, the 'Simulation' panel shows a 3D model of the GoKul device and a log of the published payloads. The log shows the following sequence of events:

- Sending payload: {"Alert Distance":86.99} Publish OK
- Sending payload: {"Alert Distance":81.97} Publish OK
- Sending payload: {"Alert Distance":78.95} Publish OK
- Sending payload: {"Alert Distance":79.00} Publish OK
- Sending payload: {"Alert Distance":78.95} Publish OK
- Sending payload: {"Alert Distance":78.95} Publish OK

## IBM RECENT EVENTS



The screenshot shows the IBM Watson IoT Platform dashboard. The 'Recent Events' tab is selected, displaying a table of events for the 'Gokul\_assignment\_4' device. The table has the following columns: Event, Value, Format, and Last Received. The events listed are:

Event	Value	Format	Last Received
Gokul	{"Alert Distance":78.95}	json	a few seconds ago
Gokul	{"Alert Distance":78.95}	json	a few seconds ago
Gokul	{"Alert Distance":78.95}	json	a few seconds ago
Gokul	{"Alert Distance":78.95}	json	a few seconds ago
Gokul	{"Alert Distance":78.95}	json	a few seconds ago

**WOVKI LINK-**

**<https://wokwi.com/projects/347102022774817362>**