

## ASSIGNMENT 4

Date	2 Nov 22
Name	Sneka catherin V
Team ID	PNT2022TMID38287
Project Name	IOT Based Smart Solution For Railways

### 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 "f33ebq"
#define DEVICE_TYPE "sneka"
#define DEVICE_ID "assignment_4"
#define TOKEN "12345678"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/sneka/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");
    }
}

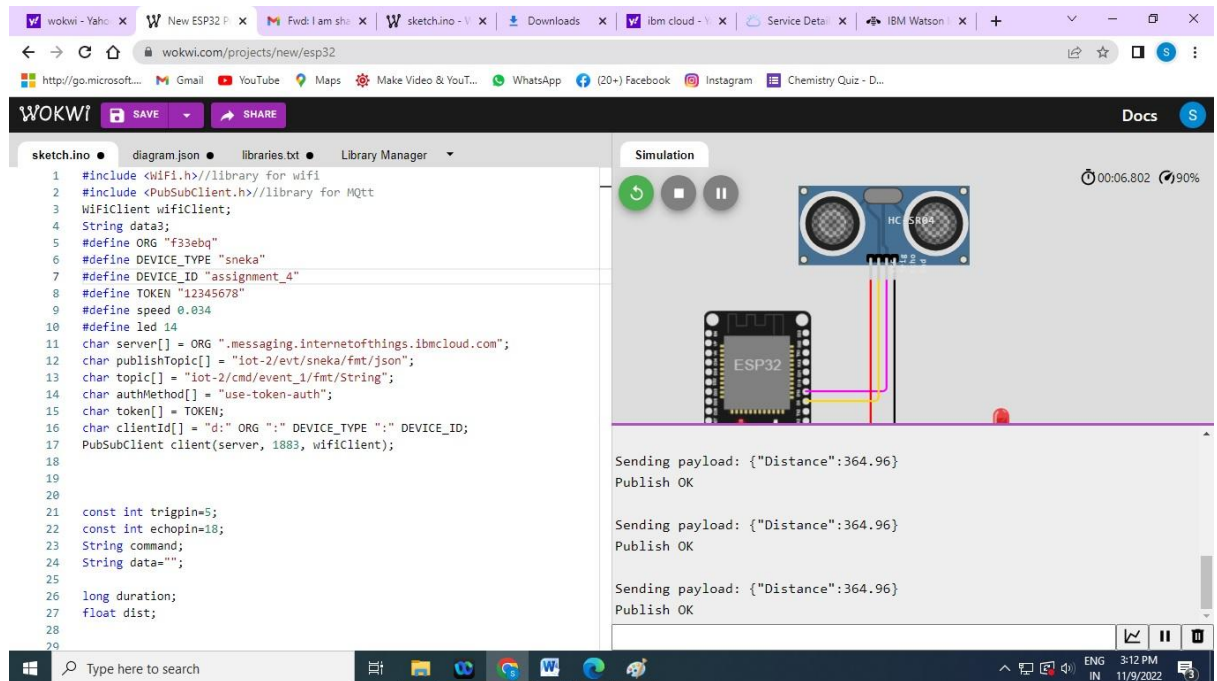
void publishData(){
    digitalWrite(trigpin,LOW);
    digitalWrite(trigpin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin,LOW);
    duration=pulseIn(echopin,HIGH);
    dist=duration*speed/2;
    if(dist<100){
        String payload = "{\"Alert Distance\":";
        payload += dist;
        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");
    }
}
}

```

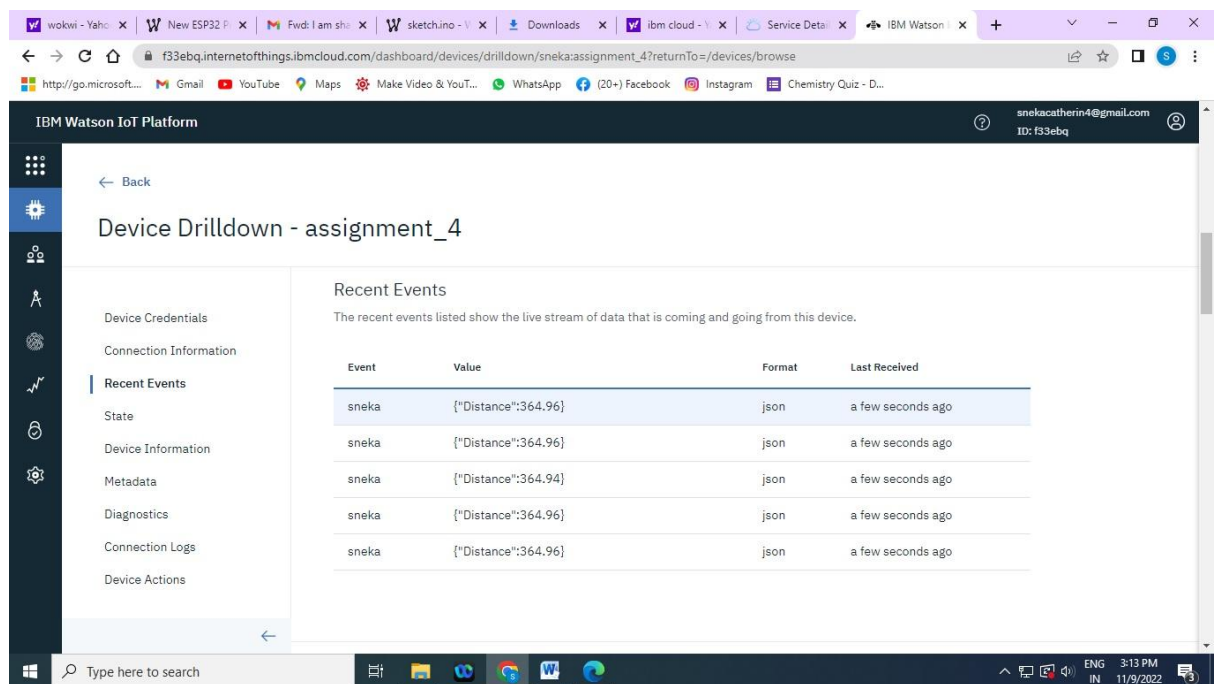
## OUTPUT :

### 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 the necessary libraries and configuration for an ESP32 to connect to an IBM Watson IoT cloud. The code defines the device name as 'sneka', the device ID as 'assignment\_4', and the token as '12345678'. It also sets the server URL to 'messaging.internetofthings.ibmcloud.com'. The main loop of the code triggers the sending of a JSON payload containing the distance measured by the ultrasonic sensor. On the right, the simulation output shows three consecutive JSON payloads being sent: {"Distance":364.96}. The simulation interface also displays a visual representation of the ESP32 and the HC-SR04 sensor connected by wires.

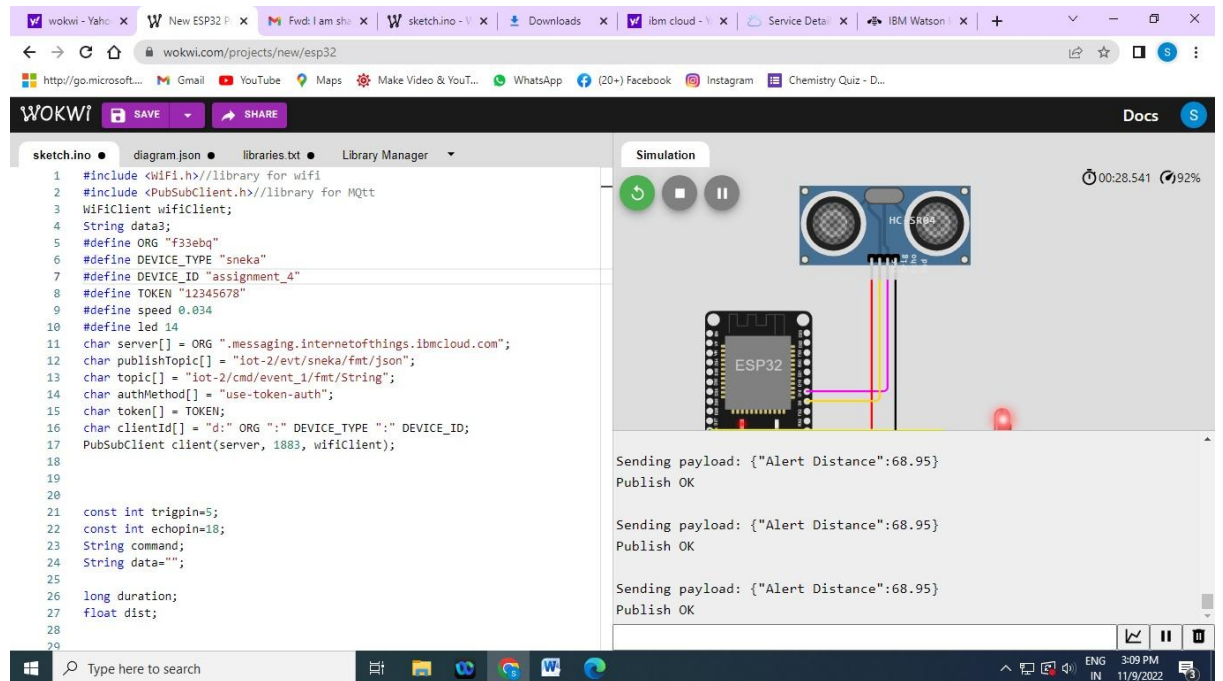
## IBM RECENT EVENTS



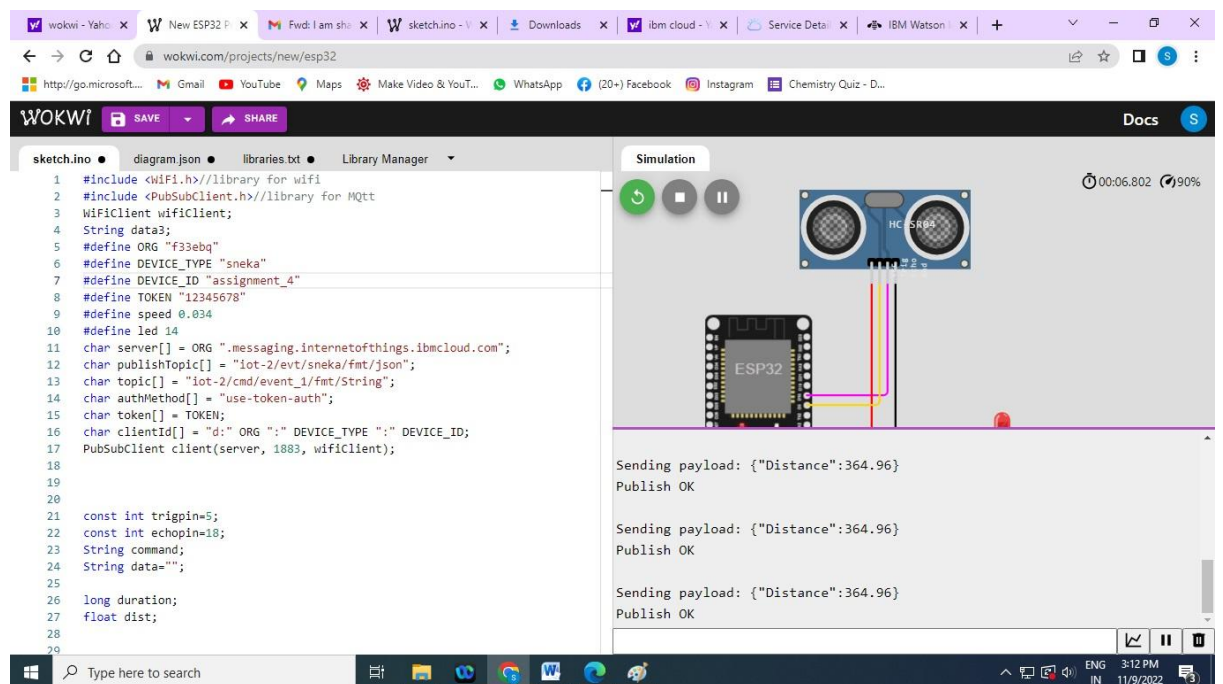
The screenshot shows the IBM Watson IoT Platform dashboard. The page is titled 'Device Drilldown - assignment\_4'. On the left, there is a sidebar with navigation options: Device Credentials, Connection Information, Recent Events, State, Device Information, Metadata, Diagnostics, Connection Logs, and Device Actions. The 'Recent Events' section is active, displaying a table of live data streams from the device 'sneka'. The table has four columns: Event, Value, Format, and Last Received. The data shows three consecutive JSON payloads being sent: {"Distance":364.96}.

Event	Value	Format	Last Received
sneka	{"Distance":364.96}	json	a few seconds ago
sneka	{"Distance":364.96}	json	a few seconds ago
sneka	{"Distance":364.94}	json	a few seconds ago
sneka	{"Distance":364.96}	json	a few seconds ago
sneka	{"Distance":364.96}	json	a few seconds ago

## 2) When distance less than 100



## IBM RECENT EVENTS



**WOKWI LINK-** <https://wokwi.com/projects/new/esp32>