

IBM ASSIGNMENT 3 – IOT DOMAIN

1. Link:

<https://wokwi.com/projects/364540012436747265>

2. Code:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT

const int trigPin = 12; // trig pin of the ultrasonic sensor
const int echoPin = 13; // echo pin of the ultrasonic sensor

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

//-----credentials of IBM Accounts-----

#define ORG "okjiu4" //IBM ORGANITION ID
#define DEVICE_TYPE "abcd" //Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "87654321" //Token
String data3;
long duration, distance_cm;

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command
type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth"; // authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id

//-----
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient);

void setup() {
  Serial.begin(9600); // initialize serial communication
  pinMode(trigPin, OUTPUT); // set trig pin as output
```

```

    pinMode(echoPin, INPUT); // set echo pin as input

    wificonnect();
    mqttconnect();
}

void loop() {

    digitalWrite(trigPin, LOW); // send a low pulse to trig pin
    delayMicroseconds(2);

    digitalWrite(trigPin, HIGH); // send a high pulse to trig pin for 10
microseconds
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH); // read the duration of the pulse from echo
pin

    distance_cm = duration / 1000; // calculate distance in cm
    if(distance_cm <=100){
        Serial.print("Distance: ");
        Serial.print(distance_cm);
        Serial.println(" cm");
        PublishData(distance_cm);
        delay(1000);
        if (!client.loop()) {
            mqttconnect();
        }
    }
    else{
        Serial.println("Distance is greater than 100 ,we cannot print and sent to
cloud");
    }
    delay(1000); // wait for 1 second before taking the next measurement
}

void PublishData(int distance) {
    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 += distance;

```

```

payload += "}";

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

if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it successfully 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);
        }

        initManagedDevice();
        Serial.println();
    }
}

void wificonnect() //function definition 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());
}

```

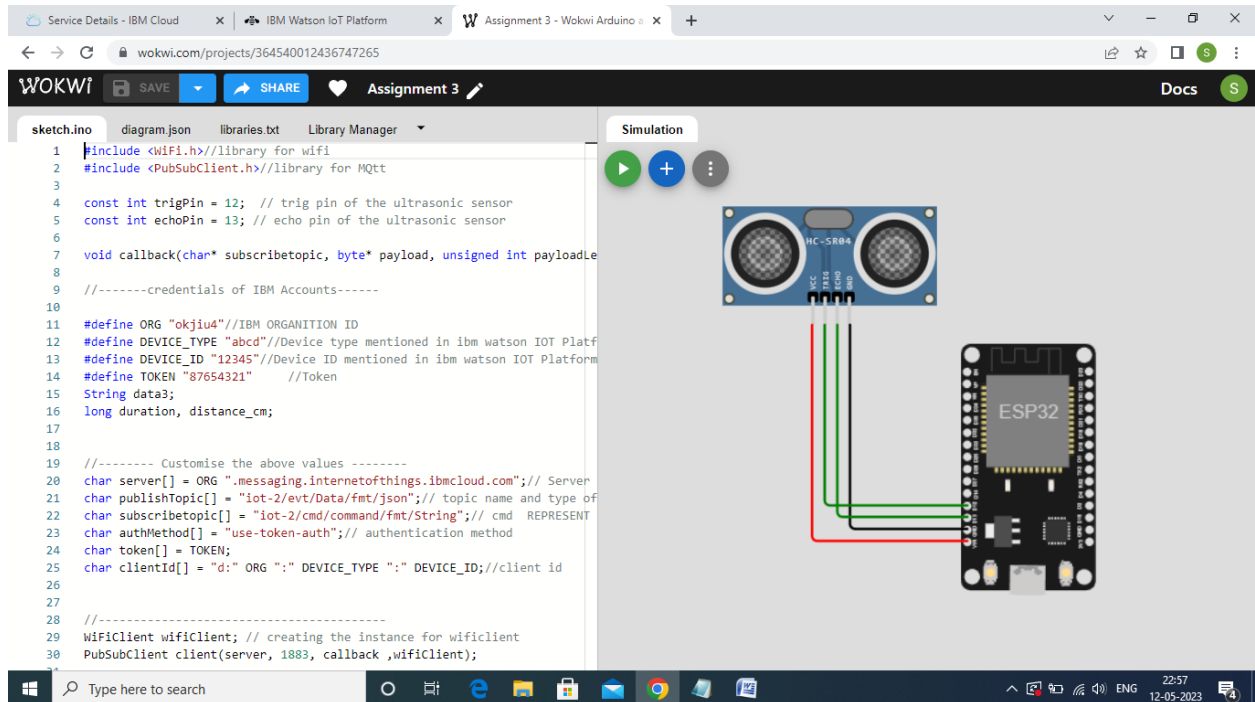
```

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);
}

```

3. Connections Screenshot:



4. IBM Cloud Recent Events Screenshot:

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for 'Service Details - IBM Cloud', 'IBM Watson IoT Platform', and 'Assignment 3 - Wokwi Arduino'. The main content area shows a list of devices. One device, '12345', is highlighted as 'Connected' with a green dot. Below the device list, a modal window titled 'Recent Events' is open, showing a table of events. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The events listed are all 'Data' events with a value of '{"distance":23}' in 'json' format, received 'a few seconds ago'.

Event	Value	Format	Last Received
Data	{"distance":23}	json	a few seconds ago
Data	{"distance":23}	json	a few seconds ago
Data	{"distance":23}	json	a few seconds ago
Data	{"distance":23}	json	a few seconds ago
Data	{"distance":23}	json	a few seconds ago