

ASSIGNMENT-4

WOKWI SIMULATOR AND IBM CLOUD

Assignment Date	19 November 2022
Student Name	Shreelakshmi R I
Student Roll Number	312319106145
Team ID	PNT2022TMID00340
Maximum Marks	2 Marks

QUESTION:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cm send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud.

SOLUTION:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
#define ECHO_GPIO 12
#define TRIGGER_GPIO 13
#define MAX_DISTANCE_CM 100 // Maximum of 0.1 meters
#include "Ultrasonic.h"
Ultrasonic ultrasonic(13, 12);
int distance;
void callback(char* subscribtopic, byte* payload, unsigned int payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "0K11gf" //IBM ORGANITION ID
#define DEVICE_TYPE "Outputesp" //Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "a22" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "pEjAlXjmhk8pWMF-gz" //Token
String data3;
float h, t;

//----- 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 subscribtopic[] = "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); //calling the predefined client id by passing
parameter like server id, port and wificredential
void setup() // configureing the ESP32
```

```

{
  Serial.begin(115200);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}
void loop()// Recursive Function
{
  distance = ultrasonic.read(CM);
  if(distance < 100){
    Serial.print("Distance in Centimeters: ");
    Serial.println(distance);
    PublishData(distance);
    delay(1000);
    if (!client.loop()) {
      mqttconnect();
    }
  }
  delay(1000);
}
/*.....retrieving to Cloud.....*/

void PublishData(float temp) {
  mqttconnect();//function call for connecting to ibm
  /*
    creating the String in in form JSon to update the data to ibm cloud
  */

  String payload = "{\"Alert Distance\":\"";
  payload += temp;
  payload += "\"";
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully 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();
  }
}

```

```

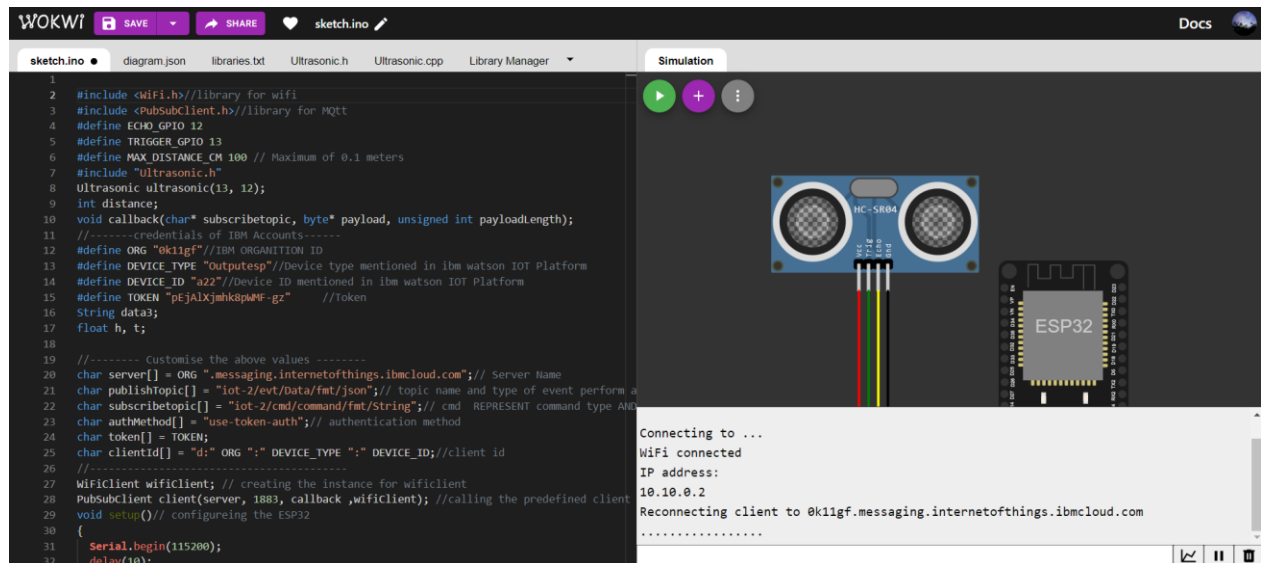
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");
  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);

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);
  if(data3=="lighton")
  {
Serial.println(data3);
  }
  else
  {
Serial.println(data3);
  }
  data3="";
}

```

Output Error: Not connecting to IBM Watson device

WOKWI SIMULATION: (Error)



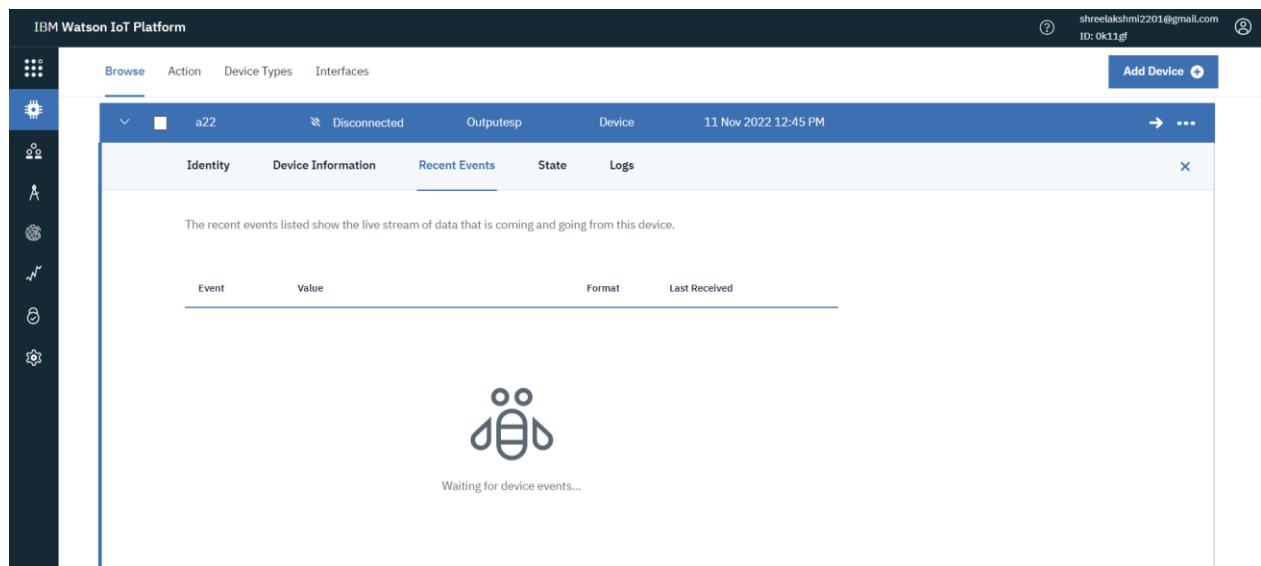
The screenshot shows the Wokwi web interface. On the left, a sketch of an Arduino Uno is displayed with the following code:

```
1
2 #include <WiFi.h> //library for wifi
3 #include <PubSubClient.h> //library for MQTT
4 #define ECHO_GPIO 12
5 #define TRIGGER_GPIO 13
6 #define MAX_DISTANCE_CM 100 // Maximum of 0.1 meters
7 #include "Ultrasonic.h"
8 Ultrasonic ultrasonic(13, 12);
9 int distance;
10 void callback(char* topic, byte* payload, unsigned int payloadLength);
11 //-----credentials of IBM Accounts-----
12 #define ORG "0k11gf" //IBM ORGANIZATION ID
13 #define DEVICE_TYPE "Outputesp" //Device type mentioned in ibm watson IOT Platform
14 #define DEVICE_ID "a22" //Device ID mentioned in ibm watson IOT Platform
15 #define TOKEN "pejAlXjmhk8pWF-gz" //Token
16 String data;
17 float h, t;
18
19 //----- Customise the above values -----
20 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
21 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform a
22 char subscribeTopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND
23 char authMethod[] = "use-token-auth"; // authentication method
24 char token[] = TOKEN;
25 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID //client id
26 //-----
27 WiFiClient wifiClient; // creating the instance for wifiClient
28 PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client
29 void setup() // configuring the ESP32
30 {
31   Serial.begin(115200);
32   delay(10);
33 }
```

On the right, the simulation window shows a visual representation of the hardware (Arduino Uno and HC-SR04 ultrasonic sensor) and a console output with the following text:

```
Connecting to ...
Wifi connected
IP address:
10.10.0.2
Reconnecting client to 0k11gf.messaging.internetofthings.ibmcloud.com
.....
```

IBM Watson Platform-Device Event Log: (Error)



The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area displays the 'Recent Events' tab for device 'a22'. The device is shown as 'Disconnected' and 'Outputesp'. The 'Recent Events' section is currently empty, with a message stating 'The recent events listed show the live stream of data that is coming and going from this device.' Below this, there is a table with columns: 'Event', 'Value', 'Format', and 'Last Received'. The table is currently empty, and a large 'Waiting for device events...' message is displayed at the bottom.

WOKWI SIMULATION LINK:

<https://wokwi.com/projects/347771913148301906>