

ASSIGNMENT 4

Ultrasonic sensor simulation in wowki

The image displays two overlapping screenshots from a Windows desktop environment, illustrating the setup for an ultrasonic sensor simulation.

Top Screenshot: IBM Watson IoT Platform

The browser window shows the "Device Drilldown - ESP32" page. The "Device Credentials" section is active, displaying the following information:

Field	Value
Organization ID	sbg7eg
Device Type	ESP32_sensor
Device ID	ESP32
Authentication Method	use-token-auth
Authentication Token	fc*R@kqF1g8XtLyg4

A warning message states: "Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the device to generate a new authentication token." A link "Find out how to add these credentials to your device" is provided.

Bottom Screenshot: Wokwi Simulation Environment

The Wokwi interface shows a sketch of an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The "Simulation" window is open, displaying the following log output:

```
Connecting to ..
WiFi connected
IP address:
10.10.0.2
Reconnecting client to sbg7eg.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe.to.cmd.OK
```

The sketch code in the background includes the following key sections:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* topic, byte* payload, unsigned int payloadlength);
//-----credentials of IBM Accounts-----
#define ORG "sbg7eg"//IBM ORGANIZATION ID
#define DEVICE_TYPE "ESP32_sensor"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "ESP32"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "fc*R@kqF1g8XtLyg4" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char subscribeTopic[] = "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);
```

WOKWI

sketch.ino

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int
4 payloadLength);
5 //-----credentials of IBM Accounts-----
6 #define ORG "sbg7eg"//IBM ORGANIZATION ID
7 #define DEVICE_TYPE "ESP32_sensor"//Device type mentioned in ibm watson IOT Platform
8 #define DEVICE_ID "ESP32"//Device ID mentioned in ibm watson IOT Platform
9 #define TOKEN "fcRkqF!jg8Xtlyg4" //Token
10 String data;
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 WiFiClient wificlient;
18 PubSubClient client(server, 1883, callback, wificlient);
19 const int trigPin = 5;
20 const int echoPin = 18;
21 #define SOUND_SPEED 0.034
22 long duration;
23 float distance;
24 void setup() {
25   Serial.begin(115200);
26   pinMode(trigPin, OUTPUT);
27   pinMode(echoPin, INPUT);
28   wificlient.connect();
29   mqttconnect();
30 }
31 void loop()
32 {
33   digitalWrite(trigPin, LOW);
34   delayMicroseconds(2);
```

Simulation

10.10.0.2
Reconnecting client to sbg7eg.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
subscribe to cmd OK

Distance (cm): 400.01
Distance (cm): 399.96

WOKWI

sketch.ino

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int
4 payloadLength);
5 //-----credentials of IBM Accounts-----
6 #define ORG "sbg7eg"//IBM ORGANIZATION ID
7 #define DEVICE_TYPE "ESP32_sensor"//Device type mentioned in ibm watson IOT Platform
8 #define DEVICE_ID "ESP32"//Device ID mentioned in ibm watson IOT Platform
9 #define TOKEN "fcRkqF!jg8Xtlyg4" //Token
10 String data;
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/data/fmt/json";
13 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 WiFiClient wificlient;
18 PubSubClient client(server, 1883, callback, wificlient);
19 const int trigPin = 5;
20 const int echoPin = 18;
21 #define SOUND_SPEED 0.034
22 long duration;
23 float distance;
24 void setup() {
25   Serial.begin(115200);
26   pinMode(trigPin, OUTPUT);
27   pinMode(echoPin, INPUT);
28   wificlient.connect();
29   mqttconnect();
30 }
31 void loop()
32 {
33   digitalWrite(trigPin, LOW);
34   delayMicroseconds(2);
```

Simulation

Distance (cm): 72.98
ALERT!!
Sending payload: {"Distance":72.98,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 72.96
ALERT!!
Sending payload: {"Distance":72.96,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 72.96

The screenshot shows the IBM Watson IoT Platform interface. The main panel displays a list of devices. The 'ESP32' device is selected, and its details are shown in a sidebar. The device is connected and its status is 'Connected'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
15548	Disconnected	15548_device	Device	18 Nov 2022 23:56	
ESP32	Connected	ESP32_sensor	Device	19 Nov 2022 14:58	

Details for ESP32:

- Device ID: ESP32
- Device Type: ESP32_sensor
- Date Added: 19 Nov 2022 14:58
- Added By: shinedavid2002@gmail.com
- Connection Status: Connected
- Connection Time: 19 Nov 2022 16:07
- Client Address: 50.31.197.64 Insecure