

Assignment 4

Wokwi

Date	30 October 2022
Team ID	PNT2022TMID48312
Project Name	Real - time river water quality monitoring and control system
Maximum Marks	2 Marks

Question-1:

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

Solution: Program

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribtopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "bb0w3d"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "s8@L6@ahNqyZPO!?ET" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribtopic[] = "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);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * SOUND_SPEED/2;
  Serial.print("Distance (cm): ");
  Serial.println(distance);
  if(distance>100)
  {
    Serial.println("ALERT!!");
    delay(100);
    PublishData(distance);
    delay(100);
    if (!client.loop()) {
      mqttconnect();
    }
  }
  delay(100);
}
void PublishData(float dist) {
  mqttconnect();
  String payload = "{\"Distance\":";
  payload += dist;
  payload += ", \"ALERT!!\":";
  payload += "\"Distance less than 100cms\"";
}

```

```

payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");
} 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(100);
        }
        initManagedDevice();
        Serial.println();
    }
}

void wificonnect()
{
    Serial.println();
    Serial.print("Connecting to ");
    WiFi.begin("Wokwi-GUEST", "", 6);
    while (WiFi.status() != WL_CONNECTED) {
        delay(100);
        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);  
  data3="";  
}
```

Resource list - IBM Cl... Node-RED: node-red sketch.ino - Wokwi Ar... Service Details - IBM C... IBM Watson IoT Platf... Inbox (11) - rameshki... wokwi.com/projects/346508314441417298

WOKWI SAVE SHARE Docs

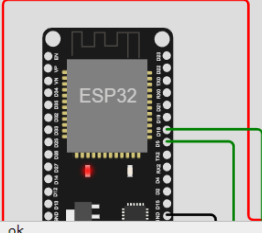
sketch.ino diagram.json libraries.txt Library Manager

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribetopic, byte* payload, unsigned int
4 payloadLength);
5 //-----credentials of IBM Accounts-----
6 #define ORG "bb0w3d"//IBM ORGANITION ID
7 #define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
8 #define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
9 #define TOKEN "s8@L6@ahNqyZP0!2ET" //Token
10 String data3;
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;
29   mqttconnect();
30 }

```

Simulation 00:50.898 25%



Publish ok
Distance (cm): 399.94
ALERT!!
Sending payload:
{ "Distance": 399.94, "ALERT!!": "Distance less than 100cms" }
Publish ok

Assignment 4 (1).pdf Assignment 4.pdf Show all

Resource list - IBM Cl... Node-RED: node-red sketch.ino - Wokwi Ar... Service Details - IBM C... IBM Watson IoT Platf... Inbox (11) - rameshki... wokwi.com/projects/346508314441417298

WOKWI SAVE SHARE Docs

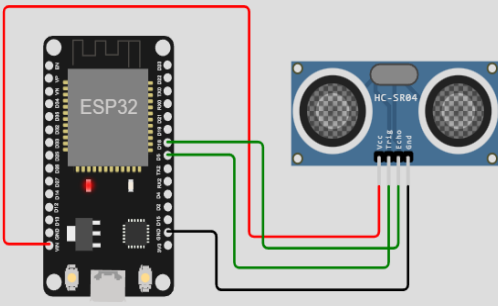
sketch.ino diagram.json libraries.txt Library Manager

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribetopic, byte* payload, uns
4 payloadLength);
5 //-----credentials of IBM Accounts-----
6 #define ORG "bb0w3d"//IBM ORGANITION ID
7 #define DEVICE_TYPE "ESP32"//Device type mentioned in i
8 #define DEVICE_ID "1234"//Device ID mentioned in ibm wa
9 #define TOKEN "s8@L6@ahNqyZP0!2ET" //Token
10 String data3;
11 char server[] = ORG ".messaging.internetofthings.ibmcl
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_I
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;
29   mqttconnect();
30 }

```

Simulation 00:02.598 59%



Connecting to

Assignment 4 (1).pdf Assignment 4.pdf Show all

Wokwi link:

<https://wokwi.com/projects/346508314441417298>

Wokwi output:

The screenshot displays the Wokwi web interface for a project simulation. The top navigation bar includes tabs for 'Resource list - IBM Cloud', 'Node-RED: node-red-kdjpl...', 'sketch.ino - Wokwi Arduino', 'Service Details - IBM Cloud', and 'IBM Watson IoT Platform'. The main interface is divided into two primary sections: a code editor on the left and a simulation window on the right.

Code Editor (sketch.ino):

```
68 }
69 void mqttconnect() {
70   if (!client.connected()) {
71     Serial.print("Reconnecting client to ");
72     Serial.println(server);
73     while (!client.connect(clientId, authMethod, token)) {
74       Serial.print(".");
75       delay(100);
76     }
77     initManagedDevice();
78     Serial.println();
79   }
80 }
81 void wificonnect()
82 {
83   Serial.println();
84   Serial.print("Connecting to ");
85   WiFi.begin("Wokwi-GUEST", "", 6);
86   while (WiFi.status() != WL_CONNECTED) {
87     delay(100);
88     Serial.print(".");
89   }
90   Serial.println("");
91   Serial.println("WiFi connected");
92   Serial.println("IP address: ");
93   Serial.println(WiFi.localIP());
94 }
95 void initManagedDevice() {
96   if (client.subscribe(subscribetopic)) {
97     Serial.println("Subscribed to ");
```

Simulation Window:

The simulation window shows a visual representation of the hardware. An ESP32 microcontroller is connected to an HC-SR04 ultrasonic sensor via jumper wires. The sensor is connected to the ESP32's VCC, GND, and Trig/Echo pins. The simulation is running, as indicated by the 'Simulation' tab and the 'Publish ok' status.

Simulation Output:

```
Publish ok
Distance (cm): 399.94
ALERT!!
Sending payload: {"Distance":399.94,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 399.94
ALERT!!
```

The bottom of the interface shows a file explorer with two PDF files: 'Assignment 4 (1).pdf' and 'Assignment 4.pdf'. A 'Show all' button is visible in the bottom right corner.

IBM cloud output:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Resource list - IBM Cloud', 'Node-RED : node-red-kdjpl...', 'Sketch.ino - Wokwi Arduino', 'Service Details - IBM Cloud', and 'IBM Watson IoT Platform'. The URL bar shows 'bb0w3d.internetofthings.ibmcloud.com/dashboard/devices/browse'. The main header identifies the user as '913019106009@smartinternz.com' with ID 'bb0w3d'. The left sidebar contains icons for various platform features. The main content area is titled 'Browse' and includes a search bar 'Search by Device ID' and a 'Device Simulator' toggle. A table lists devices, with the first device (ID 1234) selected. This device is 'Connected', of type 'ESP32', and was added on '18 Nov 2022 11:40'. A dropdown menu for this device shows tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'State' tab is active, displaying 'Showing Raw Data | No Interfaces Available' and a table of data points.

Property	Value	Type	Event	Last Received
Distance	399.92	Number	Data	a few seconds ago
ALERT!!	Distance less than 100cms	String	Data	a few seconds ago

Below the selected device, another device (ID 1234) is listed with status 'Connected', type 'Sensor_River_UP', and added on '14 Nov 2022 20:05'. The bottom of the screen shows a file manager with 'Assignment 4 (1).pdf' and 'Assignment 4.pdf' open.