

Assignment -04

Assignment	06 NOVEMBER 2022
Student Name	HARIHARAN C
Student Roll Number	513419106013
Maximum 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.

Upload document with wokwi share link and images of IBM cloud.

SOLUTION:

Wokwi link: <https://wokwi.com/projects/342753883265696339>

The screenshot shows the Wokwi IDE interface. On the left, the code for `esp32-dht22.ino` is displayed. The code includes libraries for WiFi and PubSubClient, defines IBM Cloud credentials, and sets up an MQTT client. It also configures an ultrasonic sensor (HC-SR04) connected to an ESP32. The sensor's VCC is connected to the ESP32's 5V pin, GND to GND, Trig to pin 5, and Echo to pin 18. The code sends an alert message to the IBM Cloud when the distance is less than 100 cm.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "mybs2s"//IBM ORGANITION ID
6 #define DEVICE_TYPE "Hari04"//Device type mentioned in ibm watson I
7 #define DEVICE_ID "02468"//Device ID mentioned in ibm watson IOT PL
8 #define TOKEN "kVaQU*-_3DU7cLYD4-" //Token
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/HARI_assignment4/fmt/json";
13 char topic[] = "iot-2/cmd/home/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wifiClient);
18 const int trigpin=5;
19 const int echopin=18;
20 String command;
21 String data="";
22 long duration;
23 float dist;
24 void setup()
25 {
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(trigpin, OUTPUT);
29   pinMode(echopin, INPUT);
```

The simulation window shows the hardware setup: an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The console output shows the following sequence of events:

- Connecting to Wifi...WiFi connected, IP address: 10.10.0.2
- Reconnecting MQTT client to mybs2s.messaging.internetofthings.ibmcloud.com
- IBM subscribe to cmd OK
- Sending payload: {"Alert Distance":58.04}

CONNECTING TO IBM WASTON CLOUD;

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for Browse, Action, Device Types, and Interfaces. A search bar is present with the text 'Search by Device ID'. The main content area displays a table of devices. The first device listed is '02468' with status 'Connected', device type 'Hari04', and class ID 'Device'. Below the table, a detailed view for the selected device is shown, including fields for Device ID, Device Type, Date Added, Added By, and Connection Status. The connection status is 'Connected' with a connection time of 'Nov 11, 2022 3:08 PM' and a client address of '50.31.197.64 Insecure'. A 'Device Simulator' toggle is visible in the top right corner, set to 'Off'. A notification at the bottom right states '0 Simulations running'.

Device ID	Status	Device Type	Class ID	Date Added
02468	Connected	Hari04	Device	Nov 11, 2022 12:18 PM

Device Information

Device ID	02468
Device Type	Hari04
Date Added	Nov 11, 2022 12:18 PM
Added By	hariharan0467@gmail.com
Connection Status	Connected Connection Time: Nov 11, 2022 3:08 PM Client Address: 50.31.197.64 Insecure

Display in the device recent events:

The screenshot shows the IBM Watson IoT Platform dashboard with the 'Recent Events' tab selected for the device '02468'. The tab displays a list of recent events, each with an event name, value, format, and last received time. The events are related to 'HARI_assign...' and show values for 'Alert Distance' and 'Distance'. A notification at the bottom right states '0 Simulations running'.

Event	Value	Format	Last Received
HARI_assign...	{"Alert Distance":57.95}	json	a few seconds ago
HARI_assign...	{"Alert Distance":58.04}	json	a few seconds ago
HARI_assign...	{"Distance":131.95}	json	a minute ago
HARI_assign...	{"Distance":131.97}	json	a minute ago
HARI_assign...	{"Distance":131.95}	json	a minute ago

CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "mybs2s"//IBM ORGANITION ID
#define DEVICE_TYPE "Hari04"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "02468"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "kVaQU*@_3DU7clYD4-" //Token
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/HARI_assignment4/fmt/json";
char topic[] = "iot-2/cmd/home/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");  
}  
}  
}
```