

Assignment 4:

Title:

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100cms 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.

Roll No: CITC1904077

Name : GOKUL R

Wowki Link: <https://wokwi.com/projects/348308707259449940>

Source Code:

```
#include <WiFi.h>//library for wifi
#include <WiFiClient.h>
#include <PubSubClient.h>//library for MQTT
#include <ArduinoJson.h>
// creating the instance by passing pin and typr of dht connected
float distance;
#define sound_speed 0.034
int trigpin=18;  int
echopin=19;  int led=5;
int LED=9;  long
duration;
  String message;// creating the instance by passing pin and typr of
dht connected

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

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

#define ORG "4k7d81"//IBM ORGANITION ID
#define DEVICE_TYPE "IBM"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "63697477"//Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "2(zMmQG_bjWs3d+-7b"      //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

```
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); //calling
```

the predefined client id by passing parameter like server id,portand

wificredential

```
void setup()// configureing the ESP32
```

```
{
```

```
    Serial.begin(115200);
```

```
pinMode(trigpin,OUTPUT);
```

```
pinMode(echopin,INPUT);
```

```
pinMode(led,OUTPUT);
```

```
delay(10);    Serial.println();
```

```
wificonnect();
```

```
mqttconnect();
```

```
}
```

```
void loop()// Recursive Function
```

```
{
```

```
digitalWrite(trigpin,LOW);
```

```
digitalWrite(trigpin,HIGH);
```

```
delay(1000);
```

```
digitalWrite(trigpin,LOW);  duration=pulseIn(echopin,HIGH);
```

```
distance=duration*sound_speed/2;
```

```
Serial.println("distance"+String(distance)+"cm");
```

```
if(distance<100)
```

```
{
```

```
    message="Alert";
```

```
digitalWrite(led,HIGH);
```

```
    } else {
```

```
        message="No problem";
```

```
digitalWrite(led,LOW);
```

```
    }
```

```
delay(1000);
```

```

PublishData(dista
nce,message);
    // if (!client.loop()) {
    //     mqttconnect();
    // }
}

/*.....retrieving to
Cloud.....*/

void PublishData(float d, String a) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSON to update the data to ibm
cloud
    */
    DynamicJsonDocument doc(1024);
    String payload;    doc["Distance:
"]=d;    doc["Message: "]=a;
    serializeJson(doc, 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);
    digitalWrite(LED,HIGH);
    }
    else
    {
    Serial.println(data3);
    digitalWrite(LED,LOW);
}
}

```

```

    } data3="";
}

```

Output:

WOKWI SAVE SHARE ULTRASONIC SENSOR copy Docs SIGN IN

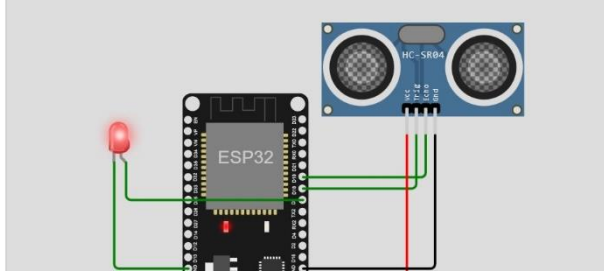
sketch.ino diagram.json libraries.txt Library Manager

```

1 #include <WiFi.h> //library for wifi
2 #include <WiFiClient.h>
3 #include <PubSubClient.h> //library for MQTT
4 // creating the instance by passing pin and type of dht connected
5 float distance;
6 #define sound_speed 0.034
7 int trigpin=18;
8 int echopin=19;
9 int led=5;
10 int LED=9;
11 long duration;
12 String message; // creating the instance by passing pin and type of dht connected
13
14 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
15
16 //-----credentials of IBM Accounts-----
17
18 #define ORG "4k7d81" //IBM ORGANIZATION ID
19 #define DEVICE_TYPE "IBM" //Device type mentioned in ibm watson IOT Platform
20 #define DEVICE_ID "63697477" //Device ID mentioned in ibm watson IOT Platform
21 #define TOKEN "2(zHmQ6_bjW53d+-7b" //Token
22 String data3;
23 float h, t;
24
25
26 //----- Customise the above values -----
27 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
28 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform
29 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND
30 char authMethod[] = "use-token-auth"; // authentication method
31 char token[] = TOKEN;
32 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
33
34

```

Simulation 01:12.915 99%



```

distance44.98cm
Sending payload: {"distance":44.98},{"message":Alert}
Publish ok
distance44.95cm
Sending payload: {"distance":44.95},{"message":Alert}
Publish ok
distance44.98cm

```

WOKWI SAVE SHARE ULTRASONIC SENSOR copy Docs SIGN UP

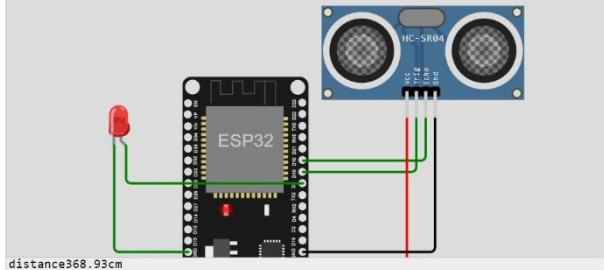
sketch.ino diagram.json libraries.txt Library Manager

```

1 #include <WiFi.h> //library for wifi
2 #include <WiFiClient.h>
3 #include <PubSubClient.h> //library for MQTT
4 // creating the instance by passing pin and type of dht connected
5 float distance;
6 #define sound_speed 0.034
7 int trigpin=18;
8 int echopin=19;
9 int led=5;
10 int LED=9;
11 long duration;
12 String message; // creating the instance by passing pin and type of dht connected
13
14 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
15
16 //-----credentials of IBM Accounts-----
17
18 #define ORG "4k7d81" //IBM ORGANIZATION ID
19 #define DEVICE_TYPE "IBM" //Device type mentioned in ibm watson IOT Platform
20 #define DEVICE_ID "63697477" //Device ID mentioned in ibm watson IOT Platform
21 #define TOKEN "2(zHmQ6_bjW53d+-7b" //Token
22 String data3;
23 float h, t;
24
25
26 //----- Customise the above values -----
27 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
28 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event perform
29 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND
30 char authMethod[] = "use-token-auth"; // authentication method
31 char token[] = TOKEN;
32 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
33
34

```

Simulation 02:07.714 99%



```

distance368.93cm
Sending payload: {"distance":368.93},{"message":No problem}
Publish ok
distance368.93cm
Sending payload: {"distance":368.93},{"message":No problem}
Publish ok
distance368.93cm

```

IBM Watson IoT Platform

?

dhaswanthdeenu@gmail.com

ID: 4k7d8l

Search

Browse

Action

Device Types

Interfaces

Add Device

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
63697477	Connected	IBM	Device	Nov 14, 2022 6:46 PM	

Identity

Device Information

Recent Events

State

Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"Distance": "368.9339905","Message": "No pro..."}	json	a few seconds ago
Data	{"Distance": "368.9339905","Message": "No pro..."}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 of 1 page

