

IBM ASSIGNMENT -04

Assignment Date	06 November 2022
Student Name	SIVAHARI V
Student Roll Number	710019106042
Team ID	PNT2022TMID42278

**Write code and connections in wokwi for ultrasonic sensor.
Whenever distance is less than 100 cms send "alert" to ibm cloud
and display in device recent events.**

CODE:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT

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

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

#define ORG "wvvr1d" //IBM ORGANITION ID
#define DEVICE_TYPE "Beacon" //Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "3013" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "Spidey_7" //Token
String data3;
float dist;

//----- 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/test/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
```

```

int LED = 4;
int trig = 5;
int echo = 18;
void setup()
{
  Serial.begin(115200);
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT);
  pinMode(LED, OUTPUT);
  delay(10);
  wificonnect();
  mqttconnect();
}
void loop()// Recursive Function
{

  digitalWrite(trig,LOW);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW);
  float dur = pulseIn(echo,HIGH);
  float dist = (dur * 0.0343)/2;
  Serial.print ("Distancein cm");
  Serial.println(dist);

  PublishData(dist);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}

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

void PublishData(float dist) {
  mqttconnect();//function call for connecting to ibm
  /*
    creating the String in in form JSon to update the data to ibm cloud
  */
  String object;
  if (dist <100)
  {
    digitalWrite(LED,HIGH);

```

```

    Serial.println("object is near");
    object = "Near";
}
else
{
    digitalWrite(LED, LOW);
    Serial.println("no object found");
    object = "No";
}

String payload = "{\"distance\": ";
payload += dist;
payload += ", \"object\": \"";
payload += object;
payload += "\"}";

Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it successfully 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 definition 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];
    }

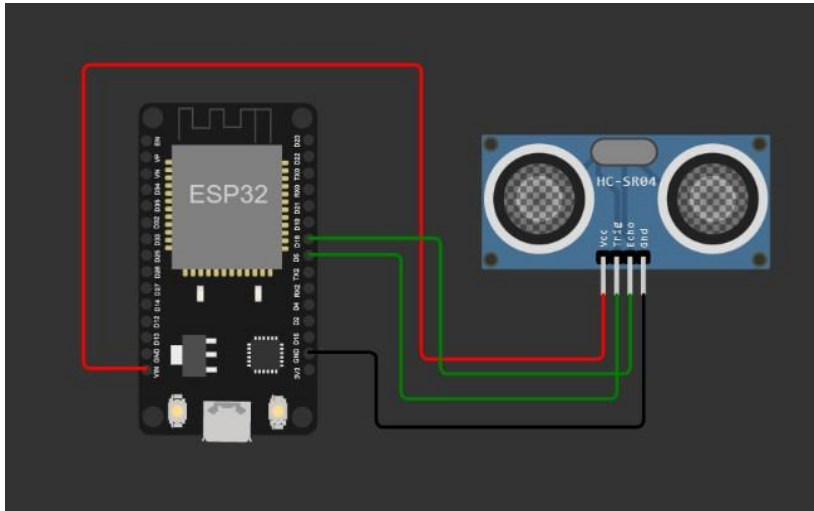
    // Serial.println("data: "+ data3);
    // if(data3=="Near")
    // {
    // Serial.println(data3);
    // digitalWrite(LED,HIGH);

    // }

    // else
    // {
    // Serial.println(data3);
    // digitalWrite(LED,LOW);
    // }
    data3="";
}

```

Schematic Diagram:



WOKWI OUTPUT:

Wokwi - Wokwi Arduino IDE

wokwi.com/projects/347641906393514580

WOKWI

sketch.ino diagram.json libraries.bt Library Manager

```
1 #include <WiFi.h> // Library for WiFi
2 #include <PubSubClient.h> // Library for MQTT
3
4 void callback(char* topic, byte* payload, unsigned int payloadLength);
5
6 //-----credentials of IBM Accounts-----
7
8 #define ORG "wvwrld" // IBM ORGANIZATION ID
9 #define DEVICE_TYPE "Beacon" // Device type mentioned in IBM Watson IoT Platform
10 #define DEVICE_ID "3013" // Device ID mentioned in IBM Watson IoT Platform
11 #define TOKEN "spidey_7" // Token
12 String data3;
13 float dist;
14
15 //----- Customise the above values -----
16
17 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
18 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event performed
19 char subscribeTopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command type AND
20 char authMethod[] = "use-token-auth"; // authentication method
21 char token[] = TOKEN;
22 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; // client id
23
24 //-----
25
26 WiFiClient wificlient; // creating the instance for wificlient
27 PubSubClient client(server, 1883, callback, wificlient); // calling the predefined client
28
29 int LED = 4;
30 int trig = 5;
31 int echo = 18;
32 void setup()
33 {
34   Serial.begin(115200);
35   pinMode(trig, OUTPUT);
36 }
```

Simulation

Editing Ultrasonic Distance Sensor

Distance: 44cm

object is near

Sending payload: {"distance":44.35,"object":"Near"}

Publish ok

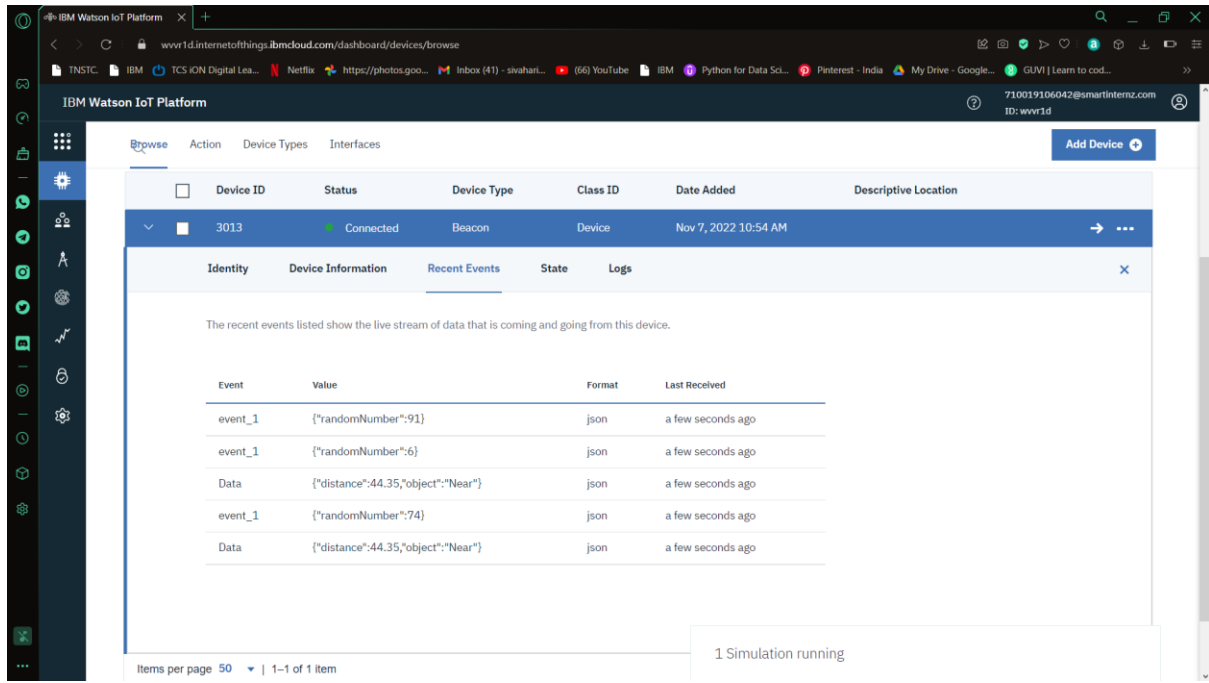
Distancein cm44.35

object is near

Sending payload: {"distance":44.35,"object":"Near"}

Publish ok

IBM CLOUD OUTPUT:



The screenshot displays the IBM Watson IoT Platform dashboard. The main view shows a table of devices. One device, ID 3013, is highlighted and its details are shown in a modal window. The device is 'Connected' and is a 'Beacon' type. The modal window shows the 'Recent Events' tab, which lists a stream of data events. The events include random numbers and distance/object data, all received 'a few seconds ago'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
3013	Connected	Beacon	Device	Nov 7, 2022 10:54 AM	

Event	Value	Format	Last Received
event_1	{"randomNumber":91}	json	a few seconds ago
event_1	{"randomNumber":6}	json	a few seconds ago
Data	{"distance":44.35,"object":"Near"}	json	a few seconds ago
event_1	{"randomNumber":74}	json	a few seconds ago
Data	{"distance":44.35,"object":"Near"}	json	a few seconds ago

1 Simulation running

WOKWI LINK:

<https://wokwi.com/projects/347641906393514580>