

## ASSIGNMENT-4

### DISTANCE DETECTION USING ULTRASONIC SENSOR

Assignment Date	29.10. 2022
Student Name	S.Indhupriya
Student Roll Number	710419106011
Maximum Marks	2 Marks

#### Question-1:

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.

WOWKI LINK:

<https://wokwi.com/projects/346856352591643219>

#### CODE:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "co65hn"
#define DEVICE_TYPE "ManiMD"
#define DEVICE_ID "manimd07"
#define TOKEN "0708012359"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/manimd/fmt/json";
char topic[] = "iot-2/cmd/led/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");
        }
    }
}

```

## OUTPUT:

The screenshot shows the Arduino IDE with the file `esp32-blink.ino` open. The code is a C++ program for an ESP32 microcontroller. It includes the `WiFi` and `PubSubClient` libraries. The code defines constants for the IBM Watson IoT Platform, including the organization ID, device type, device ID, and token. It also defines the server name, topic, and authentication method. The code sets up the WiFi and PubSubClient, and then enters a loop that sends a payload to the server every 5 seconds. The payload is a JSON object containing the distance and the object name. The simulation output shows the following messages:

```
no object found
Sending payload: {"distance":141.21,"object":"No"}
Publish ok
Distancein cm141.21
no object found
Sending payload: {"distance":141.21,"object":"No"}
Publish ok
```

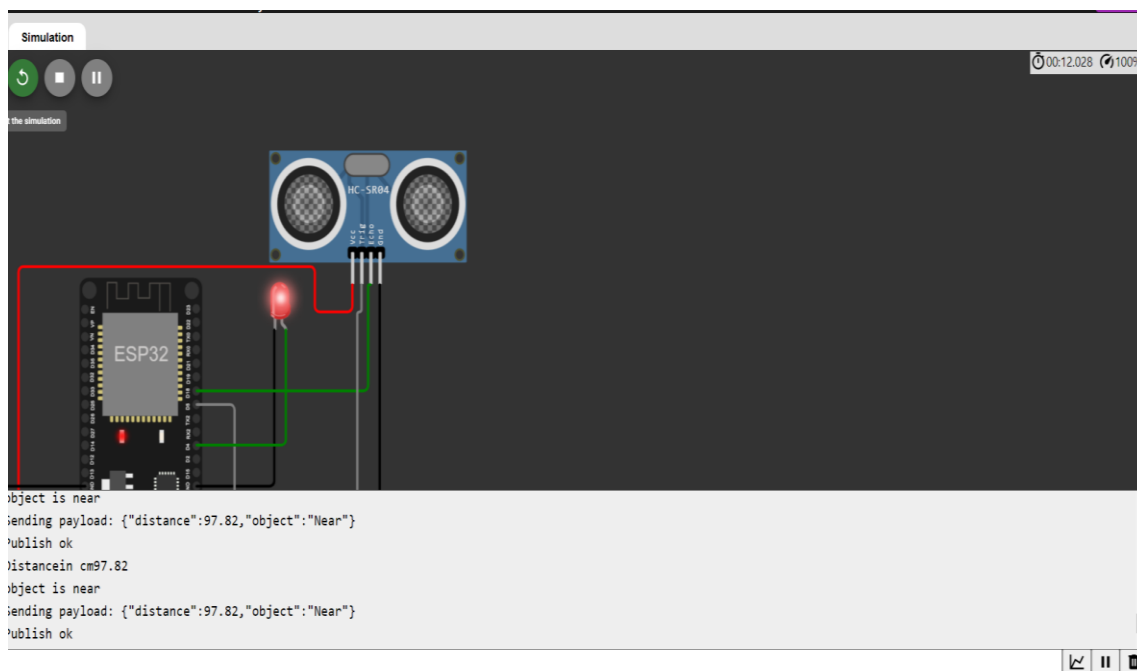
## DATA SEND TO THE IBM CLOUD DEVICE WHEN THE OBJECT IS FAR

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes `Browse`, `Action`, `Device Types`, and `Interfaces`. The main content area displays the details for the `DISTANCEDETECT` device, which is currently `Disconnected`. The `Recent Events` tab is selected, showing a list of events. The events are as follows:

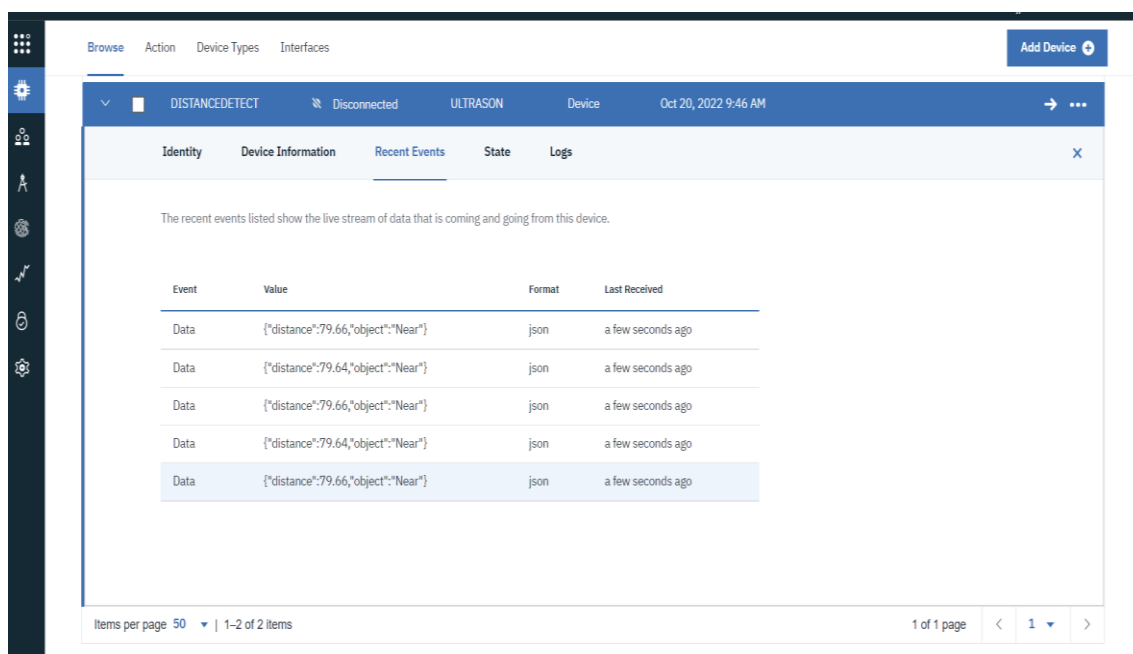
Event	Value	Format	Last Received
Data	<code>{"distance":141.21,"object":"No"}</code>	json	a few seconds ago
Data	<code>{"distance":141.21,"object":"No"}</code>	json	a few seconds ago
Data	<code>{"distance":141.18,"object":"No"}</code>	json	a few seconds ago
Data	<code>{"distance":141.2,"object":"No"}</code>	json	a few seconds ago

The interface also includes a sidebar with various icons and a bottom status bar showing `Items per page 50` and `1-2 of 2 items`.

## when object is near to the ultrasonic sensor



## Data sent to the IBM Cloud Device when the object is near



<https://wokwi.com/projects/346856352591643219>