

Assignment - 4

Assignment Date	29 October 2022
Team ID	PNT2022TMID52559
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.

Upload document with wokwi share link and images of IBM cloud

link - <https://wokwi.com/projects/347025032386445906>

The screenshot displays the Wokwi web interface for a project titled "wokwi.com/projects/347025032386445906". The interface is divided into two main sections: a code editor on the left and a simulation window on the right.

Code Editor (sketch.ino):

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4
5 #define ORG "3e40vw"
6 #define DEVICE_TYPE "iot_device"
7 #define DEVICE_ID "1234"
8 #define TOKEN "123456789"
9 #define speed 0.034
10
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Data/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
18 WiFiClient wifiClient;
19 PubSubClient client(server, 1883, wifiClient);
20
21 void publishData();
22 const int trigpin=18;
23 const int echopin=5;
24 String command;
25 String data="";
26 long duration;
27 int dist;
28
29 void setup()
30 {
31   // Serial.begin(115200);
32 }
```

Simulation Window:

The simulation window shows a hardware setup with an ESP32 microcontroller and an HC-SR04 ultrasonic sensor. The sensor is connected to the ESP32 as follows:

- VCC (red wire) to ESP32 Pin 5
- GND (black wire) to ESP32 Pin GND
- Trig (green wire) to ESP32 Pin 18
- Echo (blue wire) to ESP32 Pin 5

The simulation window also includes a "Simulation" button and a "Docs" button.

Code :

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#define ORG " 3e40vw"
#define DEVICE_TYPE "iot_device"
#define DEVICE_ID "1234"
#define TOKEN "123456789"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";char
publishTopic[] = "iot-2/evt/Data/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;
WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);void
publishData();
const int trigpin=18;
const int echopin=5;
String command;
String data="";
long duration;
int dist;
void setup()
{
  Serial.begin(115200);
```

```

    pinMode(trigpin, OUTPUT);
    pinMode(echopin, INPUT);
    wifiConnect();
    mqttConnect();
}

void loop() {
    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(1000);
        }
        initManagedDevice();
        Serial.println();
    }
}

```

```

    }
}

void initManagedDevice() { if
    (client.subscribe(topic)) {
        Serial.println(client.subscribe(topic));
        Serial.println("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;
    Serial.println(dist);
    if(dist<100){ DynamicJsonDocument
        doc(1024);String payload;
        doc["AlertDistance:"]=dist;
        serializeJson(doc, payload);
        delay(3000);
        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");

    }

}

}

```

IBM Watson IoT Platform

tvсандеep2002@gmail.com
ID: 3e40ww

Browse Action Device Types Interfaces

Add Device +

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID Device Simulator

Device ID	Status	Device Type	Class ID	Date Added
1234	Disconnected	iot_device	Device	Oct 31, 2022 3:21 PM

Items per page 50 | 1-1 of 1 item 1 of 1 page

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	{"AlertDistance":44}	json	a few seconds ago
Data	{"AlertDistance":44}	json	a few seconds ago