

Assignment - 4

Assignment Date	11 November 2022
Student Name	Priyadharshini K
Student Roll Number	412519104101

Question-1:

Write code and connections in wokwi for the ultrasonic sensor.

Sketch.ino

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "7hikdc"
#define DEVICE_TYPE "Nodemcu"
#define DEVICE_ID "12345"
#define TOKEN "UU1-JfbzMFv!05!W0a"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/priyadharshini/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);
void publishData();

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("Warning crosses 110cm -- it automatically of the loop");
            digitalWrite(led,HIGH);
        }
    }

    if(dist>101 && dist<111){
        String payload = "{\"Normal Distance\":\"";
        payload += dist;
        payload += "\"}";

        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);
    }
}

```

```

    }

}

    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++){
        dist += (char)payload[i];
    }
    Serial.println("data:" + data3);
    if(data3=="lighton"){
        Serial.println(data3);
        digitalWrite(led,HIGH);
    }
    data3="";
}

```

Diagram.json

```

{
    "version": 1,
    "author": "PRIYADHARSHINI K",
    "editor": "wokwi",
    "parts": [
        { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 60.67, "left":
-63.33, "attrs": {} },
        {
            "type": "wokwi-led",
            "id": "led1",
            "top": 53.21,
            "left": 279.26,
            "attrs": { "color": "red" }
        },
    ],
}

```

```

    "type": "wokwi-hc-sr04",
    "id": "ultrasonic1",
    "top": -33.96,
    "left": 96.97,
    "attrs": { "distance": "90" }
  },
  {
    "type": "wokwi-resistor",
    "id": "r1",
    "top": 253.89,
    "left": 220.39,
    "attrs": { "value": "100" }
  }
],
"connections": [
  [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
  [ "esp:RX0", "$serialMonitor:TX", "", [ ] ],
  [ "ultrasonic1:TRIG", "esp:D5", "yellow", [ "v0" ] ],
  [ "ultrasonic1:ECHO", "esp:D18", "magenta", [ "v0" ] ],
  [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v0" ] ],
  [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v0" ] ],
  [ "esp:D12", "r1:2", "gold", [ "h156.9", "v62.96" ] ],
  [ "led1:C", "esp:GND.2", "black", [ "v0" ] ],
  [ "r1:1", "led1:A", "purple", [ "v28.12", "h94" ] ],
  [ "esp:D12", "esp:D14", "green", [ "h0" ] ]
]
}

```

Simulation:

WOKWI

SAVE SHARE

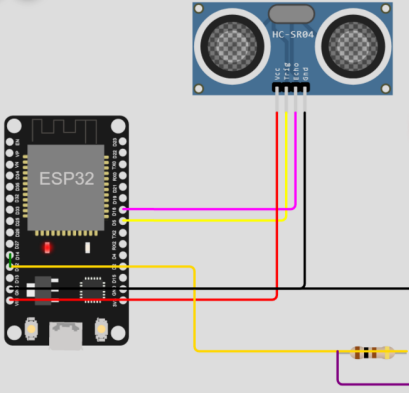
Docs

sketch.ino

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "7hikdc"
6 #define DEVICE_TYPE "NodeMCU"
7 #define DEVICE_ID "12345"
8 #define TOKEN "UUL-3fb2HFv10S!W0a"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/priyadharshini/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 void publishData();
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);
34   pinMode(led, OUTPUT);
35   pinMode(trigpin, OUTPUT);
36 }
```

Simulation

01:15.463 96%



Sending payload: {"Alert Distance":89.95}
Warning crosses 110cm -- it automatically of the loop

Type here to search

WOKWI

SAVE SHARE

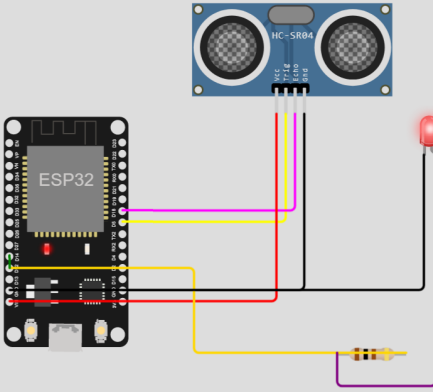
Docs

sketch.ino

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wificlient;
4 String data3;
5 #define ORG "7hikdc"
6 #define DEVICE_TYPE "NodeMCU"
7 #define DEVICE_ID "12345"
8 #define TOKEN "UUL-3fb2HFv10S!W0a"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/priyadharshini/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 void publishData();
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);
34   pinMode(led, OUTPUT);
35   pinMode(trigpin, OUTPUT);
36 }
```

Simulation

01:05.839 99%



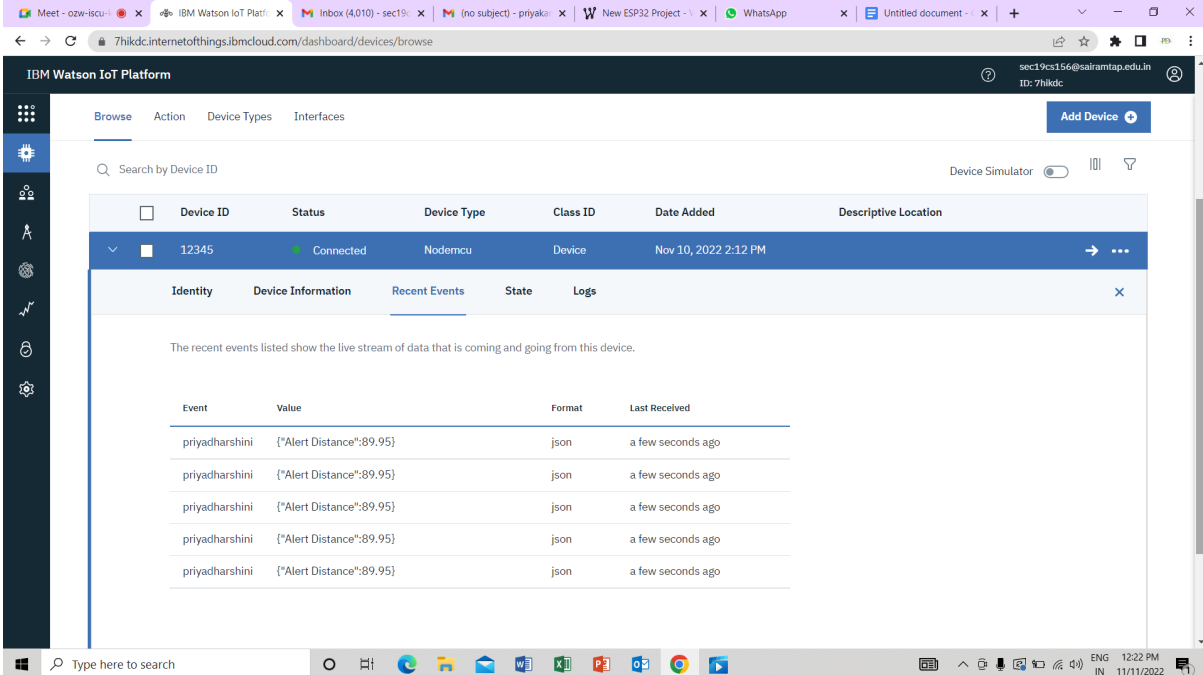
Sending payload: {"Alert Distance":89.98}
Warning crosses 110cm -- it automatically of the loop

Type here to search

Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events.

IOT platform:

Recent events



The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes '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, with ID '12345', is in a 'Connected' state and is a 'Nodemcu' type. Below the device list, the 'Recent Events' tab is selected, showing a stream of data. The events are listed in a table with columns for 'Event', 'Value', 'Format', and 'Last Received'. The events are all from the device 'priyadharshini' and represent an 'Alert Distance' of '89.95' in 'json' format, received 'a few seconds ago'.

Event	Value	Format	Last Received
priyadharshini	{"Alert Distance":89.95}	json	a few seconds ago
priyadharshini	{"Alert Distance":89.95}	json	a few seconds ago
priyadharshini	{"Alert Distance":89.95}	json	a few seconds ago
priyadharshini	{"Alert Distance":89.95}	json	a few seconds ago
priyadharshini	{"Alert Distance":89.95}	json	a few seconds ago

WOKWI link:

<https://wokwi.com/projects/348012693934834260>