

#### Assignment -4

Assignment Date	04 November 2022
Student Name	Sriabirami V
Student Roll Number	412519104137

##### 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 "17mq0"
#define DEVICE_TYPE "Sriabirami_ESP32"
#define DEVICE_ID "Sriabirami2022"
#define TOKEN "TZoE2PQ@&b2SM8FpgP"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Sriabirami/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": "Sriabirami Abirami",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -62.59, "left": -
219.46, "attrs": {} },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": -123.46,
      "left": 273.78,
      "attrs": { "color": "red" }
    },
    {
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": -196.94,
      "left": 53.14,
      "attrs": { "distance": "90" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 141.59,
      "left": 176.56,
      "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" ] ]
  ]
}
```

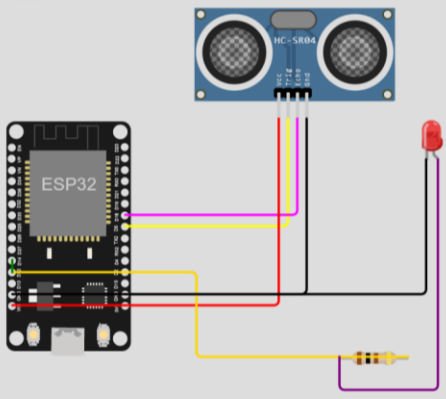
## Simulation:

WOKWI SAVE SHARE sketch.ino Docs S

sketch.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 WiFiClient wifiClient;
4 String data;
5 #define ORG "17mq0"
6 #define DEVICE_TYPE "Sriabirami_ESP32"
7 #define DEVICE_ID "Sriabirami2022"
8 #define TOKEN "T20E2PQ08b2SM8Fpgp"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Sriabirami/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);
```

Simulation

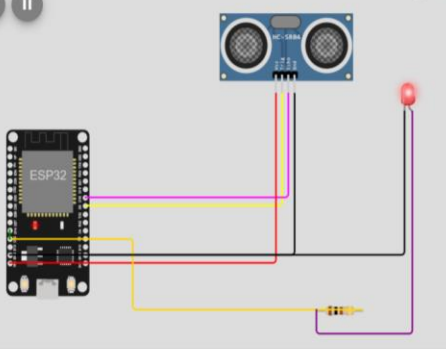


WOKWI SAVE SHARE sketch.ino Docs S

sketch.ino diagram.json libraries.txt Library Manager

```
94 payload += dist;
95 payload += " ";
96
97 Serial.print("\n");
98 Serial.print("Sending payload: ");
99 Serial.println(payload);
100 if(client.publish(publishTopic, (char*) payload.c_str())) {
101   Serial.println("Warning crosses 110cm -- it automatically of the loop");
102   digitalWrite(led,HIGH);
103 }
104
105 if(dist>101 && dist<111){
106   String payload = "{\\Normal Distance\\": ";
107   payload += dist;
108   payload += " ";
109
110   Serial.print("\n");
111   Serial.print("Sending payload: ");
112   Serial.println(payload);
113 }
114
115 }
116
117
118
119
120 void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength){
121   Serial.print("callback invoked for topic:");
122   Serial.println(subscribeTopic);
123   for(int i=0; i<payloadLength; i++){
124     dist += (char)payload[i];
125   }
126   Serial.println("data:" + data);
```

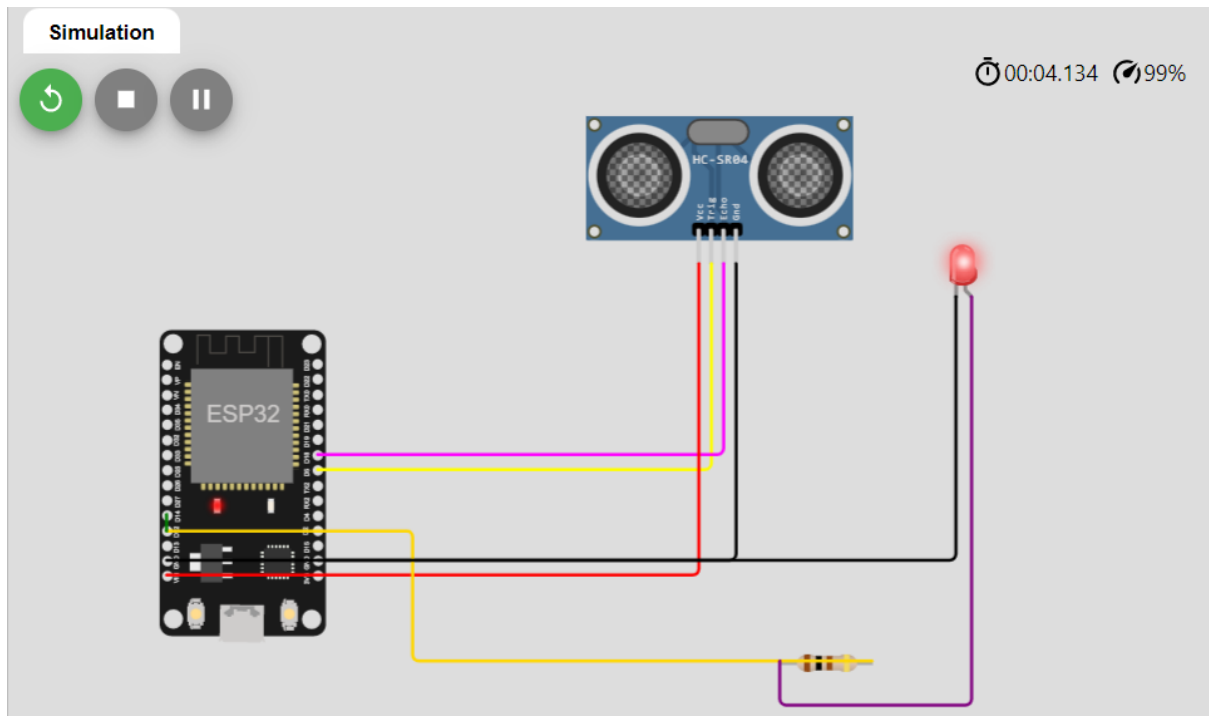
Simulation



00:14.071 99%

Connecting to Wifi..Wifi connected, IP address: 10.10.0.2  
Reconnecting MQTT client to 17mq0.messaging.internetofthings.ibmcloud.com  
IBM subscribe to cmd OK

Sending payload: {\\Alert Distance\\:89.96}  
Warning crosses 110cm -- it automatically of the loop



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

