

## ASSIGNMENT-3

**Team ID:** NM2023TMID14262

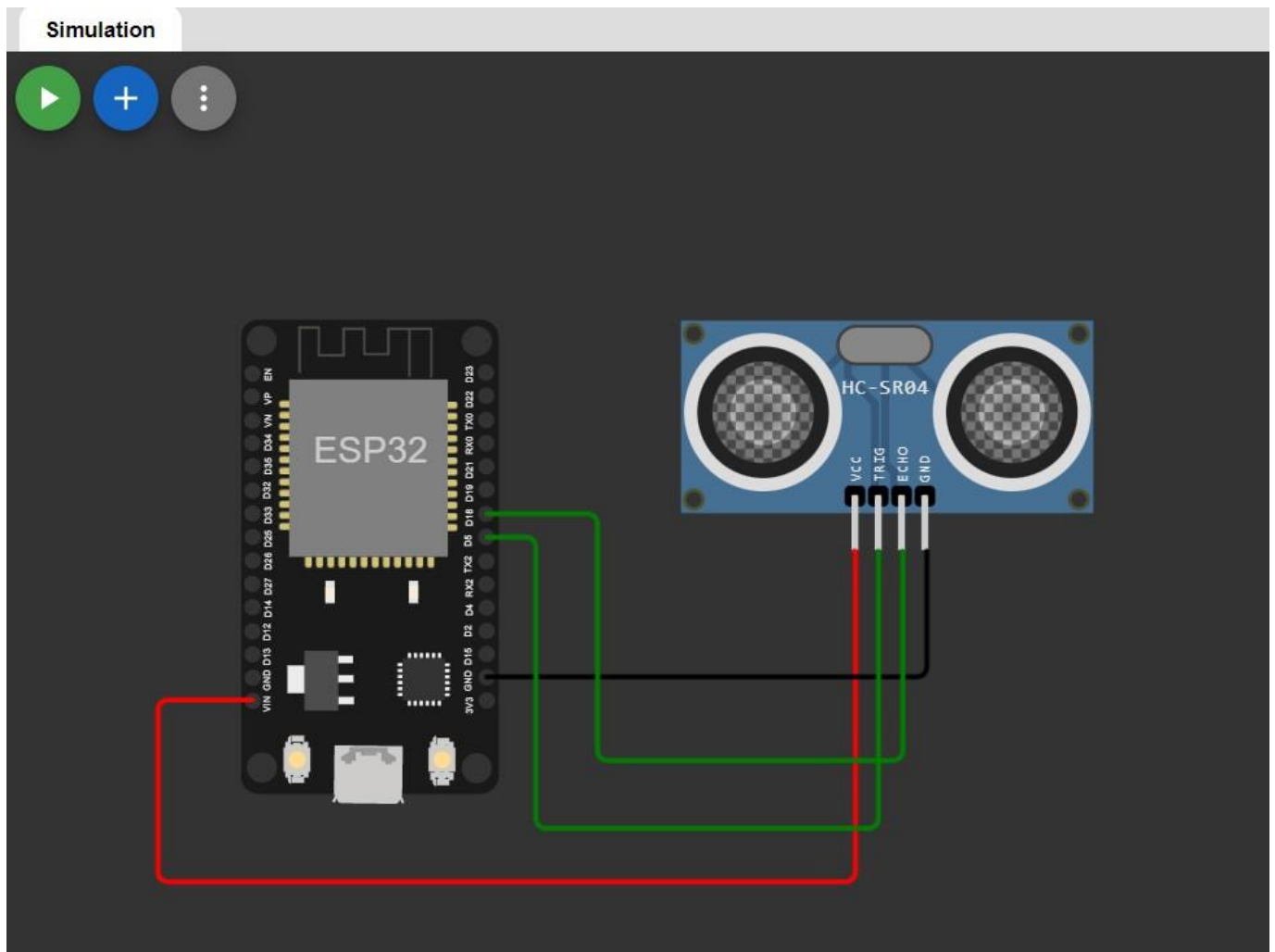
**Member Name:** Yuvaraj S K

**Domain:** IoT

**(Build WOKWI product, use ultrasonic sensor and detect the distance from the object. Whenever distance is less than 100cms upload the value to the IBM cloud in recent device events upload the data from WOKWI)**

**WOKWI LINK:** <https://wokwi.com/projects/364706331579436033>

### CIRCUIT DIAGRAM:



## CODE:

```
//-----  
//--- Assignment 3 ---  
//---Team Mem 01 : Yuvaraj S K ---  
//---Team ID: NM2023TMID14262 ---  
//-----  
  
#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 "rbmk0u"//IBM ORGANITION ID  
#define DEVICE_TYPE "abcd"//Device type mentioned in ibm watson IOT Platform  
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform  
#define TOKEN "12345678" //Token  
String data3;  
  
//----- Customise the above values -----  
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name  
char publishTopic[] = "iot-2/evt/Data/fmt/json";  
char subscribetopic[] = "iot-2/cmd/test/fmt/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);  
  
const int trigPin = 5;  
const int echoPin = 18;  
#define SOUND_SPEED 0.034  
long duration;  
float distance;  
  
void setup() // configuring the ESP32  
{
```

```

Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}

```

```

void loop() // Recursive Function

```

```

{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * SOUND_SPEED/2;
  Serial.print("Distance (cm): ");
  Serial.println(distance);

```

```

  if(distance<100)

```

```

  {
    Serial.println("ALERT!!");
    delay(1000);
    PublishData(distance);
    delay(1000);

```

```

    if (!client.loop())

```

```

    {
      mqttconnect();
    }

```

```

  }
  delay(1000);
}

```

```

/*.....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 payload = "{\"Distance\":\":";
```

```
payload += dist;  
payload += "\",\"ALERT!\":\"\"\"Distance less than 100cms\"\"\"";  
payload += "}";
```

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

```
if (client.publish(publishTopic, (char*) payload.c_str()))  
{  
    Serial.println("Publish ok");  
} 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 defination 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);  
    data3="";  
}
```

# IBM CLOUD SCREENSHOTS:

- 1234

### Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"Distance":42.98,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":28.95,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":87.97,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":87.97,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":73.98,"ALERT!":"Distance less than ...	json	a few seconds ago

IBM Watson IoT Platform

?

yuvarajsk567@gmail.com  
ID: rbmk0u

← Back

Device Drilldown - 1234

Connection Information

Recent Events

State

Device Information

Metadata

Diagnostics

Connection Logs

Device Actions

### Recent Events

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"Distance":42.98,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":28.95,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":87.97,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":87.97,"ALERT!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":73.98,"ALERT!":"Distance less than ...	json	a few seconds ago