

ASSIGNMENT-4

CODE:

```
#include <WiFi.h>

#include <PubSubClient.h>

void callback(char* subscribetopic,byte* payload, unsigned int
payloadLength);

#define ORG "1eozpn"

#define DEVICE_TYPE "esp"

#define DEVICE_ID "123"

#define TOKEN "12345678"

String data3;


char server[]= ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[]="iot-2/evt/distance/fmt/json";
char subscribeTopic[]="iot-2/cmd/test/fmt/String";
char authMethod[]="use-token-auth";
char token[]=TOKEN;
char clientID[]="d:"ORG":"DEVICE_TYPE":"DEVICE_ID;


WiFiClient wifiClient;

PubSubClient client(server,1883,callback,wifiClient);


#define ECHO_PIN 14

#define TRIG_PIN 12
```

```
#define led 27
```

```
void setup() {
```

```
  // put your setup code here, to run once:
```

```
  Serial.begin(115200);
```

```
  pinMode(led, OUTPUT);
```

```
  pinMode(TRIG_PIN, OUTPUT);
```

```
  pinMode(ECHO_PIN, INPUT);
```

```
  wificonnect();
```

```
  mqttconnect();
```

```
}
```

```
float readDistanceCM()
```

```
{ digitalWrite(TRIG_PIN,
```

```
LOW);delayMicroseconds(2);
```

```
digitalWrite(TRIG_PIN, HIGH);
```

```
delayMicroseconds(10);
```

```
digitalWrite(TRIG_PIN, LOW);
```

```
int duration=random(1,200);
```

```
//Serial.println(duration);
```

```
//duration = pulseIn(ECHO_PIN, HIGH);
```

```
return duration ;
```

```
//Serial.println(duration);
```

```
}
```

```

void loop() {
    float distance = readDistanceCM();
    //Serial.println(distance);

    bool isNearby = distance < 100;
    digitalWrite(led, isNearby);

    Serial.print("Measured distance: ");
    Serial.println(distance);
    if(distance<100){ PublishData2(dist
ance);

    }else{ PublishData1(dista
nce);

    }
    //PublishData(distance);
    delay(1000);
    if(!client.loop()){ mqttco
nnect();
    }

    //delay(2000);
}

void PublishData1(float dist){

```

```
mqttconnect();  
String payload= "{\\"distance\\":";  
payload += dist;  
payload+="}";
```

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

```
if(client.publish(publishTopic,(char*)payload.c_str())){Serial.p  
    rintln("publish ok");  
} else{  
    Serial.println("publish failed");  
}  
}
```

```
void PublishData2(float  
    dist){mqttconnect();  
    String payload= "{\\"ALERT\\":";  
    payload += dist;  
    payload+="}";
```

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

```
if(client.publish(publishTopic,(char*)payload.c_str())){Serial.p  
    rintln("publish ok");
```

```

    } else{
        Serial.println("publish failed");
    }
}

void
mqttconnect(){ if(!client.connected()){ Se
rial.print("Reconnecting to ");
Serial.println(server);
while(!client.connect(clientID, authMethod,
token)){Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}

```

```

void
wificonnect(){ Serial.println();
Serial.print("Connecting to");

WiFi.begin("Wokwi-GUEST","",6);
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(subscribe  
Topic)){ Serial.println((subscribeTopic));  
Serial.println("subscribe to cmd ok");  
}else{  
    Serial.println("subscribe to cmd failed");  
}  
}
```

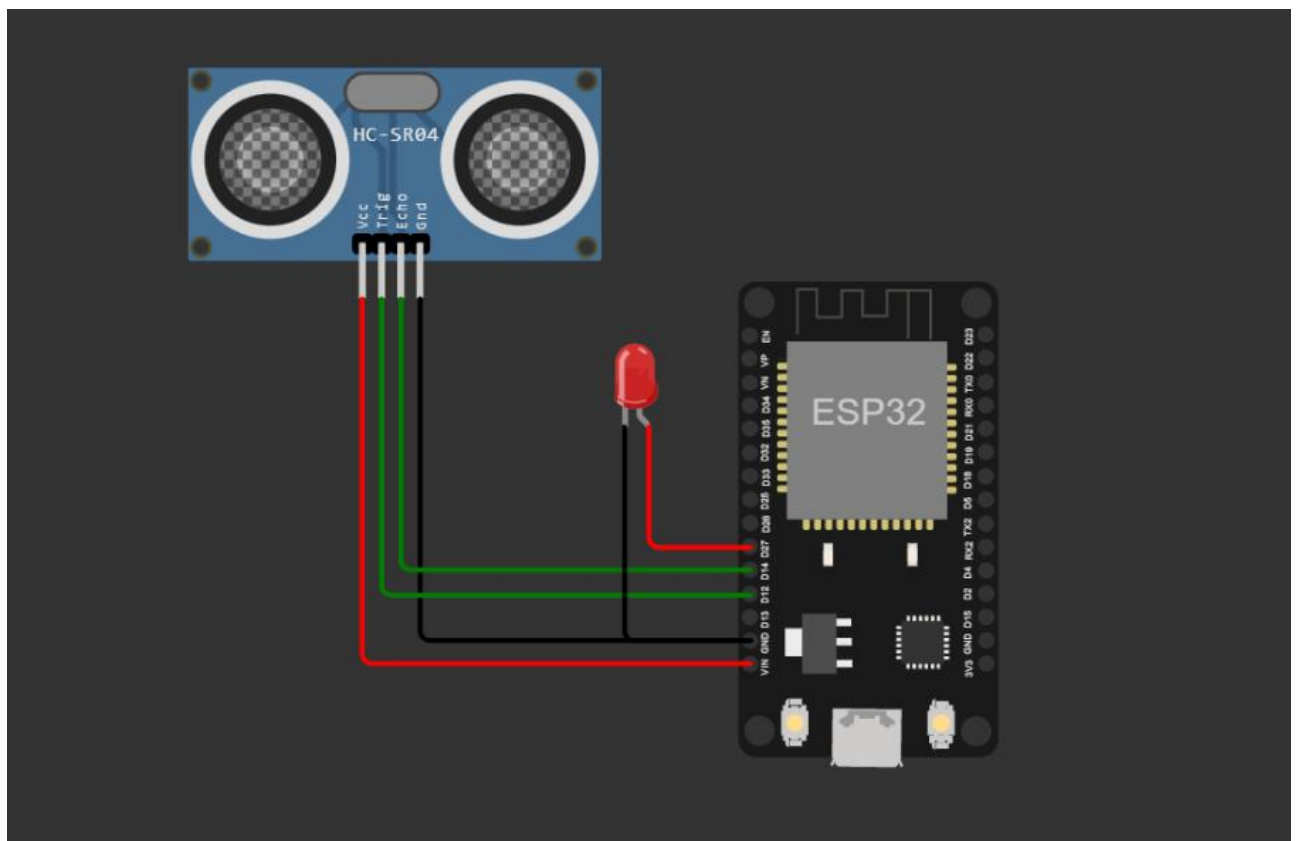
```
void callback(char* subscribeTopic, byte* payload, unsigned int  
payloadLength){  
    Serial.print("callback invoked for topic:");  
    Serial.println(subscribeTopic);  
    for(int i=0; i<payloadLength;  
        i++){data3 += (char)payload[i];  
    }  
    Serial.println("data:"+ data3);  
    if(data3=="lighton"){
```

```

Serial.println(data3);
digitalWrite(led,HIGH);
}else{ Serial.println(dat
a3);
digitalWrite(led,LOW);
}
data3="";
}

```

Circuit:



Output:

normal(>100cms)

WOKWI

https://wokwi.com/projects/347584000187957843

SAVE SHARE sketch.ino

diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribtopic,byte* payload,unsigned int payloadlength);
4 #define ORG "ieozpn"
5 #define DEVICE_TYPE "esp"
6 #define DEVICE_ID "123"
7 #define TOKEN "12345678"
8 String data3;
9
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/distance/fmt/json";
12 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:"ORG":"DEVICE_TYPE":"DEVICE_ID";
16
17 WiFiClient wifiClient;
18 PubSubClient client(server,1883,callback,wifiClient);
19
20 #define ECHO_PIN 14
21 #define TRIG_PIN 12
22 #define led 27
23
24 void setup() {
25   // put your setup code here, to run once:
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(TRIG_PIN, OUTPUT);
29   pinMode(ECHO_PIN, INPUT);
30   wifiConnect();
31   mqttConnect();
32 }
33 float readDistanceCM() {
34   digitalWrite(TRIG_PIN, LOW);
35   delayMicroseconds(2);
```

Simulation

05:17.071 64%

publish ok
Measured distance: 19.00
Sending payload:{\"ALERT\":19.00}
publish ok
Measured distance: 144.00
Sending payload:{\"distance\":144.00}
publish ok

Alert(<100cms)

WOKWI

https://wokwi.com/projects/347584000187957843

View site information SHARE sketch.ino

diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribtopic,byte* payload,unsigned int payloadlength);
4 #define ORG "ieozpn"
5 #define DEVICE_TYPE "esp"
6 #define DEVICE_ID "123"
7 #define TOKEN "12345678"
8 String data3;
9
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/distance/fmt/json";
12 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:"ORG":"DEVICE_TYPE":"DEVICE_ID";
16
17 WiFiClient wifiClient;
18 PubSubClient client(server,1883,callback,wifiClient);
19
20 #define ECHO_PIN 14
21 #define TRIG_PIN 12
22 #define led 27
23
24 void setup() {
25   // put your setup code here, to run once:
26   Serial.begin(115200);
27   pinMode(led, OUTPUT);
28   pinMode(TRIG_PIN, OUTPUT);
29   pinMode(ECHO_PIN, INPUT);
30   wifiConnect();
31   mqttConnect();
32 }
33 float readDistanceCM() {
34   digitalWrite(TRIG_PIN, LOW);
35   delayMicroseconds(2);
```

Simulation

04:01.372 96%

publish ok
Measured distance: 101.00
Sending payload:{\"distance\":101.00}
publish ok
Measured distance: 3.00
Sending payload:{\"ALERT\":3.00}
publish ok

IBM CLOUD RECEIVED DATA:

←

↺

🔒

https://1eozpn.internetofthings.ibmcloud.com/dashboard/devices/browse

🔍

🌱

🛡️

🔄

🔖

📁

⬇️

👤

IBM Watson IoT Platform

2019504056@student.annauniv.edu
ID: 1eozpn

🔍

👤

⋮

🔧

👤

🔍

⚡

🕒

⚙️

Browse

Action

Device Types

Interfaces

Add Device +

🔍 Search by Device ID

Device Simulator ☒

📄

🔍

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	
▼	123	Disconnected	esp	Device	Nov 6, 2022 6:39 PM		➔ ...

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
distance	{"ALERT":46}	json	a few seconds ago
distance	{"ALERT":2}	json	a few seconds ago
distance	{"distance":184}	json	a few seconds ago
distance	{"distance":199}	json	a few seconds ago
distance	{"distance":143}	json	a few seconds ago

0 Simulations running