

SRI ESHWAR COLLEGE OF ENGINEERING

Coimbatore

ASSIGNMENT IV

TEAM MATES :

PRAVEEN [LEAD]

THRINADH.G

LEANDER

NITHIN

KUMAR

Solution:

WOWKI LINK: <https://wokwi.com/projects/346235465961046612>

```
#include <WiFi.h>
```

```
#include
```

```
<PubSubClient.h>
```

```
#define TRIGGER 2
```

```
#define ECHO 15
```

```
#define sound_speed
```

```
0.034 int distance;
```

```
void callback(char* subscribetopic, byte* payload,
```

```
unsigned int payloadLength);  
//-----credentials of IBM Accounts-----  
#define ORG "wp72r7"  
#define DEVICE_TYPE "iot-  
device-1" #define DEVICE_ID  
"123456789"  
#define TOKEN "987654321"  
String data3;  
//----- Customise the above valueschar server[] = ORG  
".messaging.internetofthings.ibmcloud.com";  
char publishTopic[] = "iot-  
2/evt/Data/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); void setup()
{
Serial.begin(115200);
pinMode(TRIGGER,
OUTPUT);
pinMode(ECHO, INPUT);
delay(10);
Serial.printl
n();
wificonnect(
);
mqttconnec
t();
}
void loop()
{
digitalWrite(TRIGGER, HIGH);
delayMicroseconds(10);
digitalWrite(TRIGGER, LOW);
int
duration=pulseIn(ECHO,HIGH);
distance=(duration*sound_speed
)/2; Serial.print("Distance:");
Serial.print(distance);
Serial.println("cms");
if(distance<100){
}

```

```

PublishData(distance);
}
delay(1000)
;          if
(!client.loop()
)          {
mqttconnect
();
}
}
/.....retrieving to Cloud ...../
void
PublishData(int d) {
mqttconnect();
String payload =
"{\"message\":alert}";
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);
}
initManagedDevic
e();
Serial.println();
} }
void wificonnect()
{
Serial.println();
Serial.print("Connecting to
"); WiFi.begin("Wokwi-
QUEST", "", 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(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++)
{ data3 += (char)payload[i];

}
}

```

```
Serial.println("data: "+
data3); data3="";
}
```

CIRCUIT DIAGRAM:

The screenshot shows the WOKWI IoT simulator interface. On the left, the sketch code is displayed, which includes the necessary libraries and configuration for an ESP32 to connect to an IBM Cloud IoT Gateway. The code defines the device type as 'iot-device-1' and sets up the MQTT client. On the right, the simulation output shows the device connecting to the cloud, receiving an IP address, and sending an alert message.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #define TRIGGER 2
4 #define ECHO 15
5 #define sound_speed 0.034
6 int distance;
7
8 void callback(char* subscribtopic, byte* payload, unsigned int payloadlength);
9
10 //-----credentials of IBM Accounts-----
11
12 #define ORG "wp72r7"
13 #define DEVICE_TYPE "iot-device-1"
14 #define DEVICE_ID "123456789"
15 #define TOKEN "987654321"
16 String data3;
17
18 //----- Customise the above values -----
19 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
20 char publishTopic[] = "iot-2/evt/Data/fmt/json";
21 char subscribtopic[] = "iot-2/cmd/test/fmt/String";
22 char authMethod[] = "use-token-auth";
23 char token[] = TOKEN;
24 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
25
26 //-----
27
28 WiFiClient wificlient;
29 PubSubClient client(server, 1883, callback, wificlient);
30 void setup()
31 {
32   Serial.begin(115200);
33   pinMode(TRIGGER, OUTPUT);
34   pinMode(ECHO, INPUT);
35   pinMode(sound_speed, INPUT);
36 }
```

Connecting to ...
 Wifi connected
 IP address:
 10.10.0.2
 Reconnecting client to wp72r7.messaging.internetofthings.ibmcloud.com
 iot-2/cmd/test/fmt/String
 subscribe to cmd OK

399 cms.
 399 cms.
 399 cms.
 212 cms.
 30 cms.
 Sending payload: {"message":"alert"}
 Publish ok
 30 cms.

IBM CLOUD RECENT EVENTS:

The screenshot shows the IBM Watson IoT Platform 'Browse Devices' page. The 'Recent Events' tab is selected, displaying a table of events. The table has columns for Event, Value, Format, and Last Received. The events listed are all of type 'Data' with a value of '{"message":"alert"}' and a format of 'json'.

Event	Value	Format	Last Received
Data	{"message":"alert"}	json	a few seconds ago
Data	{"message":"alert"}	json	a few seconds ago
Data	{"message":"alert"}	json	a few seconds ago
Data	{"message":"alert"}	json	a few seconds ago
Data	{"message":"alert"}	json	a few seconds ago