

Assignment Date	25 October 2022
Student Name	S.Nandhagopal
Student Roll Number	811519106090
Maximum Marks	2 Marks

ASSIGNMENT-4

Problem Statement:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100 cms send "alert" to IBM cloud and display in device recent events.

Source Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
#define ORG "p4s6t5"
#define DEVICE_TYPE "Ultrasonic_Sensor_ESP32"
#define DEVICE_ID "1923"
#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 12
#define TRIG_PIN 13
#define led 2
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); // Clear the trigger
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH); // Sets the trigger pin to HIGH state for 10
microseconds
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
int duration=pulseIn(ECHO_PIN, HIGH);
//Serial.println(duration);
//duration = pulseIn(ECHO_PIN, HIGH);
return duration*0.017;
//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(distance);
}else{
PublishData1(distance);
}
//PublishData(distance);
delay(1000);
if(!client.loop()){
mqttconnect();
}
//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.println("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.println("publish ok");
  } else{
    Serial.println("publish failed");
  }
}

void mqttconnect(){
  if(!client.connected()){
    Serial.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(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);
  if(data3=="lighton"){
    Serial.println(data3);
    digitalWrite(led,HIGH);
  }else{
    Serial.println(data3);
    digitalWrite(led,LOW);
  }
  data3="";
}

```

Wokwi Link:

<https://wokwi.com/projects/346509966509605459>

Normal and Alert case:

The screenshot displays the Wokwi web interface for a project simulation. The browser address bar shows the URL: wokwi.com/projects/346509966509605459. The interface includes a top navigation bar with 'WOKWI', 'SAVE', 'SHARE', and a heart icon. Below this, there's a 'Library Manager' tab and a 'Simulation' tab. The 'Simulation' tab is active, showing a 3D model of an ESP32 board with a red LED and a yellow push button connected. The left pane shows the sketch code for an ESP32-based ultrasonic sensor project.

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* subscribetopic,byte* payload, unsigned int payloadLength);
4 #define ORG "p4s6t5"
5 #define DEVICE_TYPE "Ultrasonic_Sensor_ESP32"
6 #define DEVICE_ID "1923"
7 #define TOKEN "12345678"
8 String data3;
9 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
10 char publishTopic[] = "iot-2/evt/distance/fmt/json";
11 char subscribeTopic[] = "iot-2/cmd/test/fmt/String";
12 char authMethod[] = "use-token-auth";
13 char token[] = TOKEN;
14 char clientID[] = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;
15 WiFiClient wificlient;
16 PubSubClient client(server,1883,callback,wificlient);
17 #define ECHO_PIN 12
18 #define TRIG_PIN 13
19 #define led 2
20 void setup() {
21   // put your setup code here, to run once:
22   Serial.begin(115200);
23   pinMode(led, OUTPUT);
24   pinMode(TRIG_PIN, OUTPUT);
25   pinMode(ECHO_PIN, INPUT);
26   wificlient.connect();
27   mqttconnect();
28 }
29 float readDistanceCM() {
30   digitalWrite(TRIG_PIN, LOW); // Clear the trigger
31   delayMicroseconds(2);
32   digitalWrite(TRIG_PIN, HIGH); // Sets the trigger pin to HIGH state for 10 microseconds
33   delayMicroseconds(10);
34   digitalWrite(TRIG_PIN, LOW);
35   int duration=pulseIn(ECHO_PIN, HIGH);

```

←

→

↺

🏠

wokwi.com/projects/346509966509605459

🔇

📄

☆

⚙️

🖥️

👤

⋮

WOKWI

SAVE

SHARE

❤️

Docs👤

sketch.ino

diagram.json

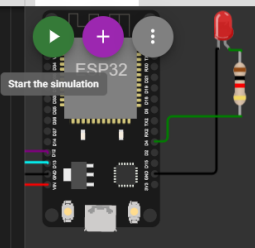
libraries.txt

Library Manager

Simulation

```
36 //Serial.println(duration);
37 //duration = pulseIn(ECHO_PIN, HIGH);
38 return duration*0.017;
39 //Serial.println(duration);
40 }
41 void loop() {
42   float distance = readDistanceCM();
43   //Serial.println(distance);
44   bool isNearby = distance < 100;
45   digitalWrite(led, isNearby);
46   Serial.print("Measured distance: ");
47   Serial.println(distance);
48   if(distance<100){
49     PublishData2(distance);
50   }else{
51     PublishData1(distance);
52   }
53   //PublishData(distance);
54   delay(1000);
55   if(!client.loop()){
56     mqttconnect();
57   }
58   //delay(2000);
59 }
60 void PublishData1(float dist){
61   mqttconnect();
62   String payload= "{\"distance\":";
63   payload += dist;
64   payload+="}";
65   Serial.print("Sending payload:");
66   Serial.println(payload);
67   if(client.publish(publishTopic,(char*)payload.c_str())){
68     Serial.println("publish ok");
69   } else{
70     Serial.println("publish failed");
```

Start the simulation



←

→

↺

🏠

wokwi.com/projects/346509966509605459

🔇

📄

☆

⚙️

🖥️

👤

⋮

WOKWI

SAVE

SHARE

❤️

Docs👤

sketch.ino

diagram.json

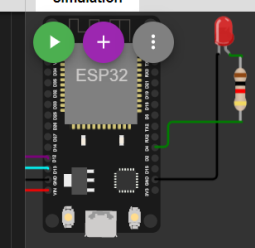
libraries.txt

Library Manager

Simulation

```
71 }
72 }
73 void PublishData2(float dist){
74   mqttconnect();
75   String payload= "{\"ALERT\":";
76   payload += dist;
77   payload+="}";
78   Serial.print("Sending payload:");
79   Serial.println(payload);
80   if(client.publish(publishTopic,(char*)payload.c_str())){
81     Serial.println("publish ok");
82   } else{
83     Serial.println("publish failed");
84   }
85 }
86 void mqttconnect(){
87   if(!client.connected()){
88     Serial.print("Reconnecting to ");
89     Serial.println(server);
90     while(!client.connect(clientID, authMethod, token)){
91       Serial.print(".");
92       delay(500);
93     }
94     initManagedDevice();
95     Serial.println();
96   }
97 }
98 void wificonnect(){
99   Serial.println();
100   Serial.print("connecting to");
101   WiFi.begin("wokwi-GUEST","",6);
102   while(WiFi.status()!=WL_CONNECTED){
103     delay(500);
104     Serial.print(".");
105   }
```

Start the simulation



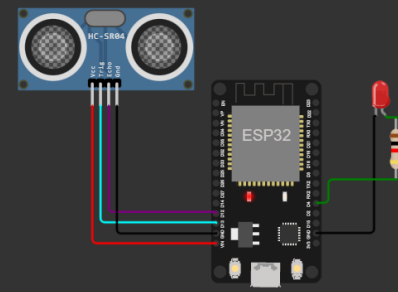
WOKWI

Sketch.ino | diagram.json | libraries.txt | Library Manager

```
106 Serial.println("");
107 Serial.println("WIFI CONNECTED");
108 Serial.println("IP address:");
109 Serial.println(WiFi.localIP());
110 }
111 void initManagedDevice(){
112   if(client.subscribe(subscribeTopic)){
113     Serial.println(subscribeTopic);
114     Serial.println("subscribe to cmd ok");
115   }else{
116     Serial.println("subscribe to cmd failed");
117   }
118 }
119 void callback(char* subscribeTopic, byte* payload, unsigned int
payloadLength){
120   Serial.print("callback invoked for topic:");
121   Serial.println(subscribeTopic);
122   for(int i=0; i<payloadLength; i++){
123     data3 += (char)payload[i];
124   }
125   Serial.println("data:"+ data3);
126   if(data3=="lighton"){
127     Serial.println(data3);
128     digitalWrite(led,HIGH);
129   }else{
130     Serial.println(data3);
131     digitalWrite(led,LOW);
132   }
133   data3="";
134 }
135 }
136 }
```

Simulation

Restart the simulation



publish ok
Measured distance: 399.92
Sending payload:{"distance":399.92}
publish ok
Measured distance: 399.94
Sending payload:{"distance":399.94}
publish ok

IBM Cloud Storage

Browse | Action | Device Types | Interfaces

Add Device

DISTANCEDETECT | Disconnected | ULTRASON | Device | Oct 20, 2022 9:46 AM

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
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago
Data	{"distance":141.21,"object":"No"}	json	a few seconds ago
Data	{"distance":141.18,"object":"No"}	json	a few seconds ago
Data	{"distance":141.2,"object":"No"}	json	a few seconds ago

Items per page 50 | 1-2 of 2 items

1 of 1 page