

Student Name	Abdul Raheem G A
Student Roll Number	2019504501

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 "8kj2a4"//IBM ORGANITION ID
#define DEVICE_TYPE "Abd_bot"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "beach"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token
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/346958589411197523>

Normal and Alert case:

The screenshot displays the Wokwi IDE interface. On the left, the 'sketch.ino' file contains the following C++ code:

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 void callback(char* topic, byte* payload, unsigned int payloadLength);
4 #define ORG "8kj2a4" //IBM ORGANITION ID
5 #define DEVICE_TYPE "Abd_bot" //Device type mentioned in ibm watson IOT Platform
6 #define DEVICE_ID "beach" //Device ID mentioned in ibm watson IOT Platform
7 #define TOKEN "12345678" //Token
8 String data;
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   wifiConnect();
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);
```

The right side of the IDE shows a 'Simulation' window with a visual representation of the ESP32, HC-SR04 sensor, and LED. Below the simulation, a console log displays the following messages:

```
Connecting to...
WIFI CONNECTED
IP address:
10.10.0.2
Reconnecting to 8kj2a4.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/String
```

The bottom status bar of the IDE shows the system clock at 21:46 on 30-10-2022, with a temperature of 29°C and weather condition 'Mostly cloudy'.

WOKWI

SAVE

SHARE

♥

Docs

👤

sketch.ino

diagram.json

libraries.txt

Library Manager

Simulation

⏮

⏪

⏩

⏭

```

35 int duration=pulseIn(ECHO_PIN, HIGH);
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");

```

03:05.149

14%

Connecting to..

WIFI CONNECTED

IP address:

10.10.0.2

Reconnecting to 8kj2a4.messaging.internetofthings.ibmcloud.com

iot-2/cmd/test/fmt/String

Type here to search

29°C Mostly cloudy

21:48

30-10-2022

IBM Cloud Storage

IBM Watson IoT Platform

abdulpeacebird5@gmail.com

ID: 8kj2a4

Browse

Action

Device Types

Interfaces

Add Device

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":47.97}	json	a few seconds ago
distance	{"ALERT":47.97}	json	a few seconds ago
distance	{"ALERT":47.97}	json	a few seconds ago
distance	{"ALERT":47.99}	json	a few seconds ago
distance	{"ALERT":47.97}	json	a few seconds ago

Items per page 50

1-1 of 1 item

1 of 1 page

<

1

>

Type here to search

29°C Mostly cloudy

21:49

30-10-2022