

Assignment -4

Assignment Date	19 October 2022
Student Name	Kavin P
Student Roll Number	722819104056
Maximum Marks	2 Marks

Question-1:

Write code and connections in wokwi for ultrasonic sensor.

Whenever distance is less than 100cms send "alert" to IBM cloud and display in device recent events.

Code:

```
#include <WiFi.h>

#include <PubSubClient.h>

#include "DHT.h"

#define DHTPIN 4

#define DHTTYPE DHT22

#define TRIGGER 2

#define ECHO 15

#define sound 0.034

DHT dht (DHTPIN, DHTTYPE);

int distance;

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

//-----credentials of IBM Accounts-----

#define ORG "85br02"

#define DEVICE_TYPE "gas_leakage_system"

#define DEVICE_ID "gas_leakage_device"

#define TOKEN "_NAMIjLh@&H&(w6*Ts"

String data3;

//----- Customise the above values -----

char 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);

dht.begin();

delay(10);

Serial.println();

wificonnect();

mqttconnect();

}

void loop()

{

digitalWrite(TRIGGER, HIGH);

delayMicroseconds(10);

digitalWrite(TRIGGER, LOW);

int time=pulseIn(ECHO,HIGH);

distance=(time* sound)/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);
        }
        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);

    data3="";

}

```

Wokwi Platform

WOKWI

SAVE

SHARE

Docs

K

sketch.ino

diagram.json

libraries.txt

Library Manager

Simulation

+

```

89  while (!client.connect(clientId, authMethod, token)) {
90      Serial.print(".");
91      delay(500);
92  }
93
94  initManagedDevice();
95  Serial.println();
96  }
97  }
98  void wificonnect()
99  {
100     Serial.println();
101     Serial.print("connecting to ");
102
103     WiFi.begin("wokwi-GUEST", "", 6);
104     while (WiFi.status() != WL_CONNECTED) {
105         delay(500);
106         Serial.print(".");
107     }
108     Serial.println("");
109     Serial.println("WiFi connected");
110     Serial.println("IP address: ");
111     Serial.println(WiFi.localIP());
112 }
113
114 void initManagedDevice() {
115     if (client.subscribe(subscribetopic)) {
116         Serial.println((subscribetopic));
117         Serial.println("subscribe to cmd OK");
118     } else {
119         Serial.println("subscribe to cmd FAILED");
120     }
121 }
122

```

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

Device Recent Events

IBM Watson IoT Platform

?

kavin.p@sece.ac.in
ID: 85br02



Browse Action Device Types Interfaces

Add Device +

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
--------------------------	-----------	--------	-------------	----------	------------	----------------------

▼	<input type="checkbox"/>	gas_leakage_device	Disconnected	gas_leakage_system	Device	Oct 21, 2022 8:31 PM	→ ...
---	--------------------------	--------------------	--------------	--------------------	--------	----------------------	-------

Identity	Device Information	Recent Events	State	Logs
----------	--------------------	---------------	-------	------

X

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

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