

Assignment -4

Assignment Date	02 November 2022
Student Name	Mr. Gokul Erusappan N
Student Roll Number	611219106023
Maximum Marks	2 Marks
Team ID	PNT2022TMID30278

Question :

Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud.

Code:

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
const int trigPin = 5;
const int echoPin = 18;
//define sound speed in cm/uS
#define Speed 0.034
#define cm_to_inch 0.393701
long duration;
float distance;
float distanceInch;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-----credentials of IBM Accounts-----

#define ORG "kdp4os"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "Ultrasonicsensor"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "09876543" //Token
String data3;

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event
perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command
type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback, wifiClient);
```

```

void setup() {
    Serial.begin(115200); // Starts the serial communication
    pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echoPin, INPUT); // Sets the echoPin as an Input
    Serial.println();
    wificonnect();
    mqttconnect();
}

void loop() {
    // Clears the trigPin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on HIGH state for 10 micro seconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);

    // Calculate the distance
    distance = duration * Speed / 2;

    // Convert to inches
    distanceInch = distance * cm_to_inch;

    // Prints the distance in the Serial Monitor
    Serial.print("Distance : ");
    Serial.println(distance);

    PublishData(distance);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

void PublishData(float centimeter) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSon to update the data to ibm cloud
    */
    String payload = "{\"Distance in Centimeter\":\"";
    payload += centimeter;
    payload += "\"}";

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

```

```

    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish ok");// if it sucessfully upload data on the cloud
then it will print publish ok in Serial monitor or else it will print publish
failed
    } 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() //function defination for wificonnect
{
    Serial.println();
    Serial.print("Connecting... ");

    WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
the connection
    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++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
}
}

```

Output:

WOKWI SAVE SHARE Docs

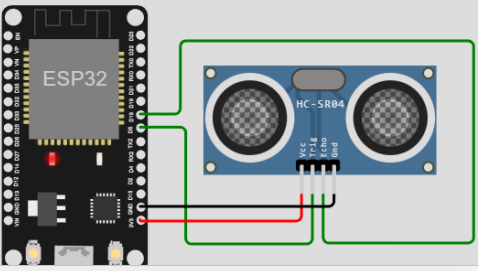
esp32_ultrasonic.ino diagram.json libraries.txt Library Manager

```

1 #include <WiFi.h>
2 #include <WiFiClient.h>
3 #include <PubSubClient.h>
4 const int trigPin = 5;
5 const int echoPin = 18;
6 //define sound speed in cm/uS
7 #define Speed 0.034
8 #define cm_to_inch 0.393701
9 long duration;
10 float distance;
11 float distanceInch;
12
13
14 void callback(char* subscribtopic, byte* payload, unsigned int payloadLength)
15 //-----credentials of IBM Accounts-----
16
17 #define ORG "kdp4os"//IBM ORGANITION ID
18 #define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT Platform
19 #define DEVICE_ID "Ultrasonicsensor"//Device ID mentioned in ibm watson IOT Platform
20 #define TOKEN "09876543" //Token
21 String data3;
22
23
24 //----- Customise the above values -----
25
26 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
27 char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event
28 char subscribtopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command
29 char authMethod[] = "use-token-auth";// authentication method
30 char token[] = TOKEN;
31

```

Simulation 01:17.795 97%



Connecting... ..
 WiFi connected
 IP address:
 10.10.0.2
 Reconnecting client to kdp4os.messaging.internetofthings.ibmcloud.com
 iot-2/cmd/test/fmt/String
 subscribe to cmd OK

WOKWI SAVE SHARE Docs

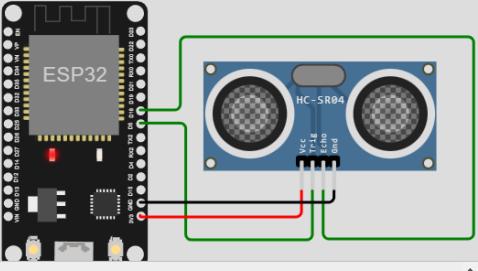
esp32_ultrasonic.ino diagram.json libraries.txt Library Manager

```

30 char token[] = TOKEN;
31 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
32
33 WiFiClient wifiClient; // creating the instance for wificlient
34 PubSubClient client(server, 1883, callback ,wifiClient);
35
36 void setup() {
37   Serial.begin(115200); // Starts the serial communication
38   pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
39   pinMode(echoPin, INPUT); // Sets the echoPin as an Input
40   Serial.println();
41   wifiConnect();
42   mqttconnect();
43 }
44
45
46 void loop() {
47   // Clears the trigPin
48   digitalWrite(trigPin, LOW);
49   delayMicroseconds(2);
50   // Sets the trigPin on HIGH state for 10 micro seconds
51   digitalWrite(trigPin, HIGH);
52   delayMicroseconds(10);
53   digitalWrite(trigPin, LOW);
54
55   // Reads the echoPin, returns the sound wave travel time in microseconds
56   duration = pulseIn(echoPin, HIGH);
57
58   // Calculate the distance
59   distance = duration * Speed/2;

```

Simulation 01:41.969 81%



Distance : 99.99
 Sending payload: {"Distance in Centimeter":99.99}
 Publish ok
 Distance : 99.94
 Sending payload: {"Distance in Centimeter":99.94}
 Publish ok
 Distance : 99.98

IBM Watson IoT Platform

2k19ece023@kio.tac.in

ID: kdp4os

Browse

Action

Device Types

Interfaces

Add Device

Device ID

Status

Device Type

Class ID

Date Added

Ultrasonicsensor

Connected

ESP32

Device

Oct 31, 2022 11:58 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 in Centimeter":99.99}	json	a few seconds ago
Data	{"Distance in Centimeter":100.01}	json	a few seconds ago
Data	{"Distance in Centimeter":99.98}	json	a few seconds ago
Data	{"Distance in Centimeter":99.98}	json	a few seconds ago
Data	{"Distance in Centimeter":99.98}		

0 Simulations running

Wokwi Share Link :

<https://wokwi.com/projects/347016872248476243>