

#### Assignment -4

Assignment Date	8 November 2022
Student Name	NAGARAJAN P
Student Roll Number	513119106057

#### QUESTION:

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

#### PROGRAM:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT

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

#define ORG "zlkemz" //IBM ORGANITION ID
#define DEVICE_TYPE "Nagarajan" //Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "12345678" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "1234567890 " //Token String data3; float dist;
//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
char publishTopic[] = "iot-2/evt/event_1/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); //calling the predefined
client id by passing parameter like server id, port and wificredential
int LED = 14;
int trig = 13;
int echo = 18;
void setup()
{
  Serial.begin(115200);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(LED, OUTPUT);
}
```

```

delay(10); wificonnect();
mqttconnect();
} void loop()// Recursive
Function
{  digitalWrite(trig,LOW);
digitalWrite(trig,HIGH);
delayMicroseconds(10);
digitalWrite(trig,LOW);  float
dur = pulseIn(echo,HIGH);  float
dist = (dur * 0.0343)/2;
    Serial.print ("Distancein cm");
    Serial.println(dist);

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

/*.....retrieving to
Cloud.....*/
void PublishData(float dist)
{
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSON to update the data to ibm cloud
    */
    String object;
    if (dist <100)
    {
        digitalWrite(LED,HIGH);
        Serial.println("object is near");
        object = "Near";
    }
    else {
        digitalWrite(LED,LOW);
        Serial.println("no object found");
        object = "No";
    }

    String payload = "{\"distance\":";
    payload += dist; payload += ","
    "\"object\":"; payload += object;
    payload += "\"}";

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

```

```

    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish ok");// if it successfully 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 definition for
wificonnect
{
    Serial.println();
    Serial.print("Connecting to ");

    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];
    }
    // Serial.println("data: "+ data3);
// if(data3=="Near")

```

```
//  {  
// Serial.println(data3);  
// digitalWrite(LED,HIGH);  
  
//  }  
  
//  else  
//  {  
// Serial.println(data3);  
// digitalWrite(LED,LOW);  
//  }  
data3="";  
  
}
```

sketch.ino diagram.json libraries.txt Library Manager

Simulation

00:11.563 99%

```

1 #include <WIFI.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4
5 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
6
7 //-----credentials of IBM Accounts-----
8
9 #define ORG "z1kemz" //IBM ORGANIZATION ID
10 #define DEVICE_TYPE "Nagarajan" //Device type mentioned in ibm watson IOT Platform
11 #define DEVICE_ID "12345678" //Device ID mentioned in ibm watson IOT Platform
12 #define TOKEN "1234567890" //Token
13 String data3;
14 float dist;
15 //----- Customise the above values -----
16 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
17 char publishTopic[] = "iot-2/evt/event_1/fmt/json"; // topic name and type of event performed
18 char subscribetopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command type AND CO
19 char authMethod[] = "use-token-auth"; // authentication method
20 char token[] = TOKEN;
21 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
22
23 //-----
24 WiFiClient wificlient; // creating the instance for wificlient
25 PubSubClient client(server,1883, callback ,wificlient); //calling the predefined client
26
27 int LED = 14;
28 int trig = 13;
29 int echo = 18;
30 void setup()
31 {
32   Serial.begin(115200);
33   pinMode(trig,OUTPUT);
34   pinMode(echo,INPUT);

```

no object found

Sending payload: {"distance":403.49,"object":"No"}

Publish ok

Distance in cm 403.49

no object found

Sending payload: {"distance":403.49,"object":"No"}

Publish ok

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

Event	Value	Format	Last Received
event_1	{"distance":403.49,"object":"No"}	json	a few seconds ago
event_1	{"distance":403.49,"object":"No"}	json	a few seconds ago
event_1	{"distance":403.49,"object":"No"}	json	a few seconds ago
event_1	{"distance":403.49,"object":"No"}	json	a few seconds ago
event_1	{"distance":403.49,"object":"No"}	json	a few seconds ago