

Assignment -4

Assignment Date	28 October 2022
Student Name	Sandhiya.V
Student Roll Number	2019504575
Maximum Marks	2 Marks

Question :

Write Code and Connections in Wokwi for Ultrasonic Sensor. Whenever Distance is less than 100 cm 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 "ckdbr5"
#define DEVICE_TYPE "123"
#define DEVICE_ID "252725"
#define TOKEN "27252527"
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 2
#define TRIG_PIN 4
#define led 5

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);
    delayMicroseconds(2);
```

```

digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
int duration=random(1,200);
//Serial.println(duration);
//duration = pulseIn(ECHO_PIN, HIGH);
return duration ;
//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="";
}

```

Reference :

<https://wokwi.com/projects/347132230257082962>

Output:

- 1) Distance less than 100cm – LED Bulb Glows and ‘Alert’ Message is Displayed along with Distance

WOKWI

sketch.ino

```

127 }
128
129 void initManagedDevice(){
130   if(client.subscribe(subscribeTopic)){
131     Serial.println((subscribeTopic));
132     Serial.println("subscribe to cmd ok");
133   }else{
134     Serial.println("subscribe to cmd failed");
135   }
136 }
137
138 void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength)
139 {
140   Serial.print("callback invoked for topic:");
141   Serial.println(subscribeTopic);
142   for(int i=0; i<payloadLength; i++){
143     data3 += (char)payload[i];
144   }
145   Serial.println("data:" + data3);
146   if(data3=="lighton"){
147     Serial.println(data3);
148     digitalWrite(LED, HIGH);
149   }else{
150     Serial.println(data3);
151     digitalWrite(LED, LOW);
152   }
153   data3="";
154 }
155 
```

Simulation

01:42.098 100%

publish ok
Measured distance: 147.00
Sending payload:{"distance":147.00}
publish ok
Measured distance: 65.00
Sending payload:{"ALERT":65.00}
publish ok

Type here to search

2) Distance more than 100cm – LED Bulb OFF and Distance is Displayed

WOKWI

sketch.ino

```

127 }
128
129 void initManagedDevice(){
130   if(client.subscribe(subscribeTopic)){
131     Serial.println((subscribeTopic));
132     Serial.println("subscribe to cmd ok");
133   }else{
134     Serial.println("subscribe to cmd failed");
135   }
136 }
137
138 void callback(char* subscribeTopic, byte* payload, unsigned int payloadLength)
139 {
140   Serial.print("callback invoked for topic:");
141   Serial.println(subscribeTopic);
142   for(int i=0; i<payloadLength; i++){
143     data3 += (char)payload[i];
144   }
145   Serial.println("data:" + data3);
146   if(data3=="lighton"){
147     Serial.println(data3);
148     digitalWrite(LED, HIGH);
149   }else{
150     Serial.println(data3);
151     digitalWrite(LED, LOW);
152   }
153   data3="";
154 }
155 
```

Simulation

01:54.081 100%

publish ok
Measured distance: 68.00
Sending payload:{"ALERT":68.00}
publish ok
Measured distance: 184.00
Sending payload:{"distance":184.00}
publish ok

Type here to search

Cloud – Storage :

The screenshot shows the IBM Watson IoT Platform dashboard for a device named 'ckdb5'. The top navigation bar includes links for 'Browse', 'Action', 'Device Types', and 'Interfaces', along with a 'Add Device' button. The left sidebar features various icons for device management. The main content area is titled 'Recent Events' and displays a table of received data. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The data shows five entries related to 'distance' values, some of which trigger alerts. At the bottom, there's a search bar, a taskbar with pinned applications, and a system tray showing weather (25°C Cloudy), date (01-11-2022), and time (19:20).

Event	Value	Format	Last Received
distance	{"distance":160}	json	a few seconds ago
distance	{"distance":158}	json	a few seconds ago
distance	{"distance":117}	json	a few seconds ago
distance	{"ALERT":92}	json	a minute ago
distance	{"ALERT":63}	json	a minute ago