

Assignment -4

Assignment Date	07-11-2022
Student Name	Yugesh M
Student Roll Number	921319104233

Question-1:

WRITE THE CODE AND CONNECTION IN WOKWI FOR THE ULTRASONIC SENSOR, WHENEVER THE DISTANCE IS LESS THAN 100CMS SEND AN ALERT TO THE IBM CLOUD AND DISPLAY THE RECNT EVENTS.

Solution:

WOKWI PROJECT ID/LINK: <https://wokwi.com/projects/3462716581241923>

Code:

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
#define ORG "6jw3v9"
#define DEVICE_TYPE "ESP32"
#define DEVICE_ID "100100C40A24"
#define TOKEN "#####"
String data3;
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);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  wificonnect();
  mqttconnect();
}
```

```

void loop()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration * SOUND_SPEED/2;
    Serial.print("Distance in cm: ");
    Serial.println(distance);
    if(distance<100)
    {
        Serial.println("ALERT!!");
        delay(1000);
        PublishData(distance);
        delay(1000);
        if (!client.loop()) {
            mqttconnect();
        }
    }
    delay(1000);
}

void PublishData(float dist) {
    mqttconnect();
    String payload = "{\"distance\": ";
    payload += dist;
    payload += ", \"alert...\": \"\" \"Distance less than 100 cms\" \"\"";
    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 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="";
}

```

Output:

WOKWI

SAVE

SHARE

Docs

SIGN IN

sketch.ino

diagram.json

libraries.txt

Library Manager

Simulation

▶

+

⋮

```

77 }
78 void wificonnect()
79 {
80   Serial.println();
81   Serial.print("connecting to ");
82   WiFi.begin("Wokwi-GUEST", "", 0);
83   while (WiFi.status() != WL_CONNECTED) {
84     delay(500);
85     Serial.print(".");
86   }
87   Serial.println("");
88   Serial.println("WiFi connected");
89   Serial.println("IP address: ");
90   Serial.println(WiFi.localIP());
91 }
92 void initManagedDevice() {
93   if (client.subscribe(subscribetopic)) {
94     Serial.println(subscribetopic);
95     Serial.println("subscribe to cmd OK");
96   } else {
97     Serial.println("subscribe to cmd FAILED");
98   }
99 }
100 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
101 {
102   Serial.print("callback invoked for topic: ");
103   Serial.println(subscribetopic);
104   for (int i = 0; i < payloadLength; i++)
105   {
106     data3 += (char)payload[i];
107   }
108   Serial.println("data: " + data3);
109   data3="";
110 }

```

Browse

Action

Device Types

Interfaces

Add Device

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By
13546	Disconnected	Raspberry	Device	Oct 31, 2022 3:19 PM		www.testing01@gmail.com

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	["distance":74]	json	a few seconds ago
Distance	["distance":84]	json	a few seconds ago
Distance	["distance":112]	json	a few seconds ago
Distance	["distance":152]	json	a few seconds ago
Distance	["distance":168]	json	a few seconds ago

1 Simulation running

Activate Windows
Go to Settings to activate Windows.