Assignment -4
Distance Detection Using Ultrasonic Sensor

Assignment Date	15 October 2022
Student Name	Mr. JETTI NITHIN REDDY
Student Roll Number	111519104054
Maximum Marks	2 Marks

Question-1:

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

WOKWI LINK

https://wokwi.com/projects/346781721885147732/

Solution:

```
#include <WiFi.h>

#include <PubSubClient.h>

WiFiClient wifiClient;

String data3;

#define ORG "7rpixh"

#define DEVICE_TYPE "ESP32_Controller"

#define DEVICE_ID "BME280_Sensor"

#define TOKEN "y@RMBgQ53@SjAYkNFG"

#define speed 0.034
```

```cpp
#define led 14


char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Srinivas/fmt/json";


char topic[] =
"iot-2/cmd/home/fmt/String"; char
authMethod[] = "use-token-auth"; char
token[] = TOKEN;


char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);

void publishData();



const int trigpin=5;


const int echopin=18;


String command;


String data="";


long duration;


float dist;
```

```cpp
void setup()

{

  Serial.begin(115200);

  pinMode(led, OUTPUT);

  pinMode(trigpin, OUTPUT);

  pinMode(echopin, INPUT);

  wifiConnect();

  mqttConnect();

}

void loop() {

  bool isNearby = dist < 100;

  digitalWrite(led, isNearby);

  publishData();

  delay(500);

  if (!client.loop()) {

    mqttConnect();

  }

}

void wifiConnect() {
```

```arduino
  Serial.print("Connecting to "); Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);

  while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

  }


  Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP()); }


void mqttConnect() {

  if (!client.connected()) {

    Serial.print("Reconnecting MQTT client to ");

Serial.println(server);

    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");

      delay(500);

    }

    initManagedDevice();

    Serial.println();

  }

}
```

```cpp
void initManagedDevice() {

  if (client.subscribe(topic)) {

    // Serial.println(client.subscribe(topic));
    Serial.println("IBM subscribe to cmd OK");

  } else {

    Serial.println("subscribe to cmd FAILED");

  }

}

void publishData()

{

  digitalWrite(trigpin,LOW);

  digitalWrite(trigpin,HIGH);

  delayMicroseconds(10);

  digitalWrite(trigpin,LOW);

  duration=pulseIn(echopin,HIGH);

  dist=duration*speed/2;

  if(dist<100){

    String payload = "{\"Alert Distance\":";

    payload += dist;

    payload += "}";

    Serial.print("\n");
```

```arduino
    Serial.print("Sending payload: ");

    Serial.println(payload);

      if(client.publish(publishTopic, (char*) payload.c_str())) {
      Serial.println("Warning crosses 110cm -- it automaticaly of the
loop");

      digitalWrite(led,HIGH);

    }

  }

    if(dist>101 && dist<111){

    String payload = "{\"Normal Distance\":";

    payload += dist;

    payload += "}";

    Serial.print("\n");

    Serial.print("Sending payload: ");

    Serial.println(payload);

    }

  }


  void callback(char* subscribeTopic, byte* payload, unsigned
int payloadLength){
```

```cpp
Serial.print("callback invoked for topic:");
Serial.println(subscribeTopic);

for(int i=0; i<payloadLength; i++){

  dist += (char)payload[i];

}

Serial.println("data:"+ data3);

if(data3=="lighton"){

  Serial.println(data3);

  digitalWrite(led,HIGH);

}

data3="";

}
```



```cpp
1  #include <WiFi.h>
2  #include <PubSubClient.h>
3  WiFiClient wifiClient;
4  String data3;
5  #define ORG "7rpixh"
6  #define DEVICE_TYPE "ESP32_Controller"
7  #define DEVICE_ID "BME280_Sensor"
8  #define TOKEN "y@RMBgQ53@SjAYkNFG"
9  #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/Srinivas/fmt/json";
13 char topic[] = "iot-2/cmd/home/fmt/String";
14 char authMethod[] = "use-token-auth";
15 char token[] = TOKEN;
16 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
17 PubSubClient client(server, 1883, wifiClient);
18 void publishData();
19
20
21 const int trigpin=5;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
31 void setup()
32 {
33   Serial.begin(115200);
34   pinMode(led, OUTPUT);
35   pinMode(trigpin, OUTPUT);
36   pinMode(echopin, INPUT);
```

Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert Distance":89.95}
Warning crosses 110cm -- it automaticaly of the loop

Sending payload: {"Alert Distance":89.95}
Warning crosses 110cm -- it automaticaly of the loop