

ASSIGNMENT – 4

Name : Manimaran R

Roll no: 711319EC060

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 <PubSubClient.h>
#include <ArduinoJson.h>

#define ORG "he8juu"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "12"
#define TOKEN "12345678"
#define SPEED 0.034

WiFiClient wifiClient;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/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 distance;
```

```

void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  wifiConnect();
  mqttConnect();

}

void loop() {
  publishData();
  delay(500);
  if (!client.loop()) {
    mqttConnect();
  }
}

void wifiConnect() {
  Serial.print("Connecting to WiFi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("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(1000);
    }
    ManagedDevice();
    Serial.println();
  }
}

void ManagedDevice() {
  if (client.subscribe(topic)) {

```

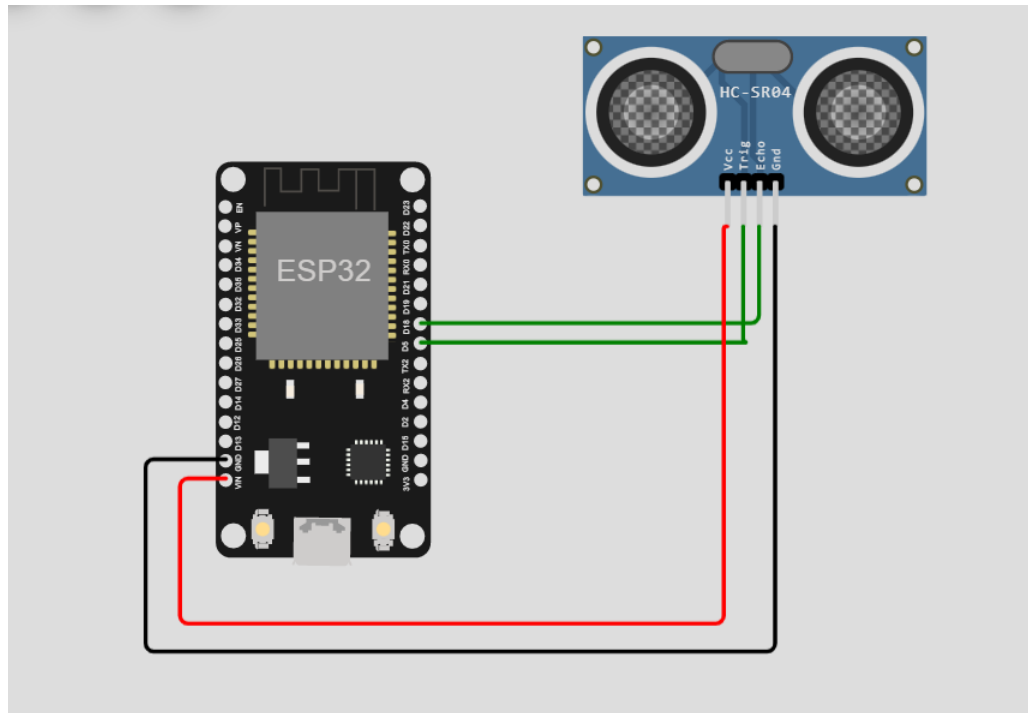
```
    Serial.println(client.subscribe(topic));
    Serial.println("subscribe to cmd OK");
}
else {
    Serial.println("subscribe to cmd FAILED");
}
}
```

```
void publishData() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration * SPEED/2;

    if(distance<100){
        DynamicJsonDocument document(1024);
        String load;
        Serial.println("ALERT!!!");
        document["Distance : "]=distance;
        serializeJson(document,load);
        delay(2000);
        Serial.println("Sending payload: ");
        Serial.println(load);
        if (client.publish(publishTopic, (char*) load.c_str())) {
            Serial.println("Publish OK");
        }
        else {
            Serial.println("Publish FAILED");
        }
    }
    else{

        Serial.print("Distance : ");
        Serial.println(distance);
        Serial.println("LOOKS GOOD!!!");
    }
    delay(1000);
}
```

CIRCUIT:



RESULT:

```
Distance : 294.97
LOOKS GOOD!!
Publish OK
ALERT!!!
Distance: 26.96
Publish OK
-ALERT!!!
Distance: 26.96
Publish OK
ALERT!!!
Distance: 26.96
Publish OK
ALERT!!!
Distance: 26.96
Publish OK
```

