

ASSIGNMENT – 4

Name: Nivetha.M

Date: 01.11.2022

Register Number: 922119106065

Project Title: Signs with Smart Connectivity for Better Road Safety

Question:

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.

Program:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "8ceru9"
#define DEVICE_TYPE "ESP32"
#define DEVICE_ID "niveassignment"
#define TOKEN "nive2204"
#define speed 0.034
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=13;
const int echopin=12;
String command;
String data="";
long duration;
int dist;
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 "); 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(1000);
    }
    initManagedDevice();
    Serial.println();
  }
}
void initManagedDevice() {
  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);
  digitalWrite(trigpin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin, LOW);
  duration=pulseIn(echopin, HIGH);
  dist=duration*speed/2;
  if(dist<100){
    DynamicJsonDocument doc(1024);
    String payload;
    doc["Distance Alert:"]=dist;
    serializeJson(doc, payload);
    delay(3000);
    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);
    if (client.publish(publishTopic, (char*) payload.c_str())) {
      Serial.println("Publish OK");
    } else {
      Serial.println("Publish FAILED");
    }
  }
}
}

```

Wokwi Link: <https://wokwi.com/projects/347029694617485907>

Output Images:

The screenshot displays the Wokwi online Arduino IDE interface. On the left, the sketch code is visible, which includes the necessary libraries and configurations for an ESP32 to connect to IBM Watson IoT and publish distance data. The code defines the server, topics, and pins, and includes a loop that publishes data every 500ms. On the right, the simulation window shows an ESP32 board connected to an HC-SR04 ultrasonic sensor. The sensor's output is being published as JSON payloads, with the distance value being updated from 13cm to 58cm.

```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4 WiFiClient wificlient;
5 #define ORG "8ceru9"
6 #define DEVICE_TYPE "ESP32"
7 #define DEVICE_ID "niveassignment"
8 #define TOKEN "nive2204"
9 #define speed 0.034
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/Data/fmt/json";
12 char topic[] = "iot-2/cmd/home/fmt/String";
13 char authMethod[] = "use-token-auth";
14 char token[] = TOKEN;
15 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
16 PubSubClient client(server, 1883, wificlient);
17 void publishData();
18 const int trigpin=13;
19 const int echopin=12;
20 String command;
21 String data="";
22 long duration;
23 int dist;
24 void setup()
25 {
26   Serial.begin(115200);
27   pinMode(trigpin, OUTPUT);
28   pinMode(echopin, INPUT);
29   wificlient.connect();
30   mqttConnect();
31 }
32 void loop() {
33   publishData();
34   delay(500);
35   if (!client.loop()) {
36     mqttConnect();
37   }
38 }
```

Simulation window output:

```
Publish OK
Sending payload: {"Distance Alert":58}
Publish OK
Sending payload: {"Distance Alert":13}
Publish OK
```

IBM Cloud:

The screenshot shows the IBM Watson IoT Platform dashboard. The 'Recent Events' tab is selected, displaying a list of events for the device 'niveassignment'. The events are listed in a table with columns for Event, Value, Format, and Last Received. The events show a sequence of distance alerts, with the first event having a value of 13 and the subsequent three having a value of 58.

Event	Value	Format	Last Received
Data	{"Distance Alert":13}	json	a few seconds ago
Data	{"Distance Alert":58}	json	a few seconds ago
Data	{"Distance Alert":58}	json	a few seconds ago
Data	{"Distance Alert":58}	json	a few seconds ago