

Assignment -4

Assignment Date	28 Oct 2022
Team ID	PNT2022TMID16916
Student Name	Mohamed Abdul Rahuman P
Student Roll Number	92172019102078
Project Name	SmartFarmer-IoT Enabled Smart Farming Application

Question:

Write a Code and Connections in wokwi for **ultrasonic sensor**. Whenever distance is less than 100 cms send “**alert**” to ibm cloud and display in device recent events

Code:

```
#include <WiFi.h> //library for wifi
#include <PubSubClient.h> //library for MQTT
WiFiClient wifiClient;
String data3;
#define ORG "g05aq3"
#define DEVICE_TYPE "selva"
#define DEVICE_ID "selva_assignment_4"
#define TOKEN "qwertyuio"
#define speed 0.034 #define led 14 char server[] = ORG
".messaging.internetofthings.ibmcloud.com"; char publishTopic[]
= "iot-2/evt/selva/fmt/json"; char topic[] = "iot-
2/cmd/status/fmt/String"; char authMethod[] = "use-token-
auth"; char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
```

```
PubSubClient client(server, 1883, wifiClient);
```

```
const int trigpin=19; const  
int echopin=18; String  
command;  
String data="";  
long duration;  
float dist;
```

```
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()  
{
```

```

    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();
    }
}
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");
    Serial.print("Sending payload: ");
    Serial.println(payload);    if
(client.publish(publishTopic, (char*) payload.c_str()))
{
    Serial.println("Publish OK");
}
}
if(dist>100){
    String payload = "{\"Distance\":\"";
    payload += dist;    payload += "}";

    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");  
  }  
}  
  
}
```

Output:

1. When distance greater than 100 cm

Wokwi IoT Platform interface showing a simulation of an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The code in the sketch.ino file is as follows:

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "g05aq3"
6 #define DEVICE_TYPE "selva"
7 #define DEVICE_ID "selva_assignment_4"
8 #define TOKEN "qwertyuio"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/selva/fmt/json";
13 char topic[] = "iot-2/cmd/status/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
19
20
21 const int trigpin=19;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
28
29
30
```

The simulation shows the ESP32 connected to the HC-SR04 sensor. The output log displays the following messages:

```
Publish OK
Sending payload: {"Distance":160.97}
Publish OK
Sending payload: {"Distance":160.97}
Publish OK
```

The interface also shows a timer at 00:26.081 and a battery level at 89%.

2. When distance less than 100 cm

The screenshot displays the Wokwi IoT simulation environment. The left pane shows the sketch code for an ESP32 connected to an Ultrasonic Distance Sensor. The code includes MQTT client setup and logic to send an alert when the distance is less than 100 cm.

```
1 #include <WiFi.h>//library for wifi
2 #include <PubSubClient.h>//library for MQTT
3 WiFiClient wifiClient;
4 String data3;
5 #define ORG "g05aq3"
6 #define DEVICE_TYPE "selva"
7 #define DEVICE_ID "selva_assignment_4"
8 #define TOKEN "qwertyuio"
9 #define speed 0.034
10 #define led 14
11 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
12 char publishTopic[] = "iot-2/evt/selva/fmt/json";
13 char topic[] = "iot-2/cmd/status/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
19
20
21 const int trigpin=19;
22 const int echopin=18;
23 String command;
24 String data="";
25
26 long duration;
27 float dist;
```

The right pane shows the simulation interface. A slider for the Ultrasonic Distance Sensor is set to 87cm. The console output shows the device sending an alert payload:

```
Publish OK
Sending payload: {"Alert Distance":86.96}
Publish OK
Sending payload: {"Alert Distance":86.96}
Publish OK
```