

Assignment -4

Team ID	PNT2022TMID06004
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

Screenshot of the Wokwi IoT Platform simulation interface showing the code and the simulated hardware.

Code (sketch.ino):

```
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;
```

Simulation:

The simulation shows an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The sensor's VCC pin is connected to the ESP32's 5V pin, and its GND pin is connected to the ESP32's GND pin. The sensor's Trig pin is connected to the ESP32's pin 19, and its Echo pin is connected to the ESP32's pin 18.

Simulation Log:

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

The simulation is running at 00:26.081 with 89% battery level.

IBM RECENT EVENTS:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main content area shows a device named 'selva' with a status of 'Connected'. Below this, the 'Recent Events' tab is selected, displaying a table of live data streams. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The data shows five events, all with a 'Distance' value and a 'json' format, received 'a few seconds ago'. A notification at the bottom right indicates '1 Simulation running'.

IBM Watson IoT Platform

312819106034@smartinternz.com
ID: g05aq3

Browse Action Device Types Interfaces

selva_assignment_4 Connected selva Device Oct 24, 2022 8:13 PM

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
selva	{"Distance":160.97}	json	a few seconds ago
selva	{"Distance":160.97}	json	a few seconds ago
selva	{"Distance":160.96}	json	a few seconds ago
selva	{"Distance":160.97}	json	a few seconds ago
selva	{"Distance":153.97}	json	a few seconds ago

1 Simulation running

2. When distance less than 100 cm

Wokwi Assignment 4 - Wokwi A x IBM Watson IoT Platform x +

wokwi.com/projects/346410390406562387

WOKWI SAVE SHARE

Docs

sketch.ino diagram.json libraries.txt Library Manager

```
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;
```

Simulation

00:33.027 85%

Editing Ultrasonic Distance Sensor

Distance: 87cm

ESP32

Publish OK

Sending payload: {"Alert Distance":86.96}

Publish OK

Sending payload: {"Alert Distance":86.96}

Publish OK

Type here to search

21:12 24-10-2022

IBM RECENT EVENTS:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area shows a modal window with the 'Recent Events' tab selected. This tab displays a table of recent events for a device named 'selva'. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. All events are of type 'selva' and contain the JSON value '{"Alert Distance":86.96}' in 'json' format, received 'a few seconds ago'. A status message at the bottom right of the modal indicates '1 Simulation running'.

Event	Value	Format	Last Received
selva	{"Alert Distance":86.96}	json	a few seconds ago
selva	{"Alert Distance":86.96}	json	a few seconds ago
selva	{"Alert Distance":86.96}	json	a few seconds ago
selva	{"Alert Distance":86.96}	json	a few seconds ago
selva	{"Alert Distance":86.96}	json	a few seconds ago

WOKWI LINK :- <https://wokwi.com/projects/346410390406562387>