

## Industry-specific intelligent fire management system :

NAME: BALLAM PRAVEEN KUMAR

ROLL.NO: 711119106004

TEAM.ID: PNT2022TMID42644

### 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..

### Solution:

```
#include <WiFi.h>
#include <PubSubClient.h>

#define ORG "0bm892"
#define DEVICE_TYPE "ESP32_Controller"
#define DEVICE_ID "Sensor"
#define TOKEN "1234567890"
#define trigpin 5
#define echopin 18
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
long duration;
float dist;
void setup()
{
    Serial.begin(9900);

    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(500);
        }
        Serial.println();
    }
}

void publishData()
{
    digitalWrite(trigpin, LOW);
    digitalWrite(trigpin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin, LOW);
    duration=pulseIn(echopin, HIGH);
    dist=(duration*0.034) /2;
    if(dist<100)
    {
        String payload = "{\"Distance\":\"";
        payload += dist;
        payload += ",";
        payload += "\"Status\":\"";
        payload += "\"Alert\"}";
        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 += ",";
  payload += "\"Status\":";
  payload += "\"Normal\"}";
  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");
  }
}
}
}

```

## Execution :

The screenshot displays the Wokwi IDE interface for an ESP32 project. The left pane shows the `sketch.ino` file with the following code:

```

1 #include <WiFi.h>
2 #include <PubSubClient.h>
3
4 #define ORG "0bm892"
5 #define DEVICE_TYPE "ESP32_Controller"
6 #define DEVICE_ID "Sensor"
7 #define TOKEN "1234567890"
8 #define trigpin 5
9 #define echopin 18
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/data/fmt/json";
12 char authMethod[] = "use-token-auth";
13 char token[] = TOKEN;
14 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
15 WiFiClient wifiClient;
16 PubSubClient client(server, 1883, wifiClient);
17 long duration;
18 float dist;
19 void setup()
20 {
21   Serial.begin(9900);
22
23   pinMode(trigpin, OUTPUT);
24   pinMode(echopin, INPUT);
25   wifiConnect();
26   mqttConnect();
27 }
28
29 void loop() {

```

The right pane shows a simulation of the ESP32 board with an HC-SR04 ultrasonic sensor. The console output displays the following messages:

```

Sending payload: {"Distance":34.97,"Status":"Alert"}
Publish OK

Sending payload: {"Distance":125.95,"Status":"Normal"}
Publish OK

Sending payload: {"Distance":125.95,"Status":"Normal"}
Publish OK

Sending payload: {"Distance":1.99,"Status":"Alert"}
Publish OK

```

The bottom status bar indicates the temperature is 28°C, the weather is Cloudy, and the time is 18:15 on 30-10-2022.

## IBM CLOUD OUTPUT :

The screenshot displays the IBM Watson IoT Platform interface. The browser address bar shows the URL: <https://0bm892.internetofthings.ibmcloud.com/dashboard/devices/browse>. The user is logged in as 'praveenkumarraj111@gmail.com' with ID '0bm892'.

The dashboard has a sidebar with navigation icons and a top navigation bar with tabs: 'Browse', 'Action', 'Device Types', and 'Interfaces'. An 'Add Device' button is located in the top right corner.

The main content area shows details for a device named 'ESP32\_Controller', which is a 'Sensor' and is 'Connected'. The device was last seen on 'Oct 30, 2022 4:42 PM'. Below this, there are tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is selected.

A message states: 'The recent events listed show the live stream of data that is coming and going from this device.'

Event	Value	Format	Last Received
data	{"Distance":1.99,"Status":"Alert"}	json	a few seconds ago
data	{"Distance":125.95,"Status":"Normal"}	json	a few seconds ago
data	{"Distance":125.95,"Status":"Normal"}	json	a few seconds ago
data	{"Distance":34.97,"Status":"Alert"}	json	a few seconds ago
data	{"Distance":34.97,"Status":"Alert"}	json	a few seconds ago

At the bottom right, a box indicates '0 Simulations running'.