

TEAM ID	PNT2022TMID04647
PROJECT NAME	IoT-Based Smart Crop Protection System for Agriculture

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 ECHO_GPIO 12
```

```
#define TRIGGER_GPIO 13
```

```
#define MAX_DISTANCE_CM 100
```

```
#include "Ultrasonic.h"
```

```
Ultrasonic ultrasonic(13, 12);
```

```
int distance;
```

```
void callback(char* subscribtopic, byte* payload, unsigned int payloadLength);
```

```
#define ORG "wjquinn"
```

```
#define DEVICE_TYPE "ESP32_controller"
```

```
#define DEVICE_ID "ESP32"
```

```
#define TOKEN "1234567890"
```

```
String data3;
```

```
float h, t;
```

```
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
```

```
char publishTopic[] = "iot-2/evt/Data/fmt/json";
```

```
char subscribtopic[] = "iot-2/cmd/command/fmt/String";
```

```
char authMethod[] = "use-token-auth";
```

```
char token[] = TOKEN;
```

```
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
```

```
WiFiClient wifiClient;
```

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

```
void setup()
```

```
{
```

```
  Serial.begin(115200);
```

```
  delay(10);
```

```
  Serial.println();
```

```
  wificonnect();
```

```

    mqttconnect();
}
void loop()
{
    distance = ultrasonic.read(CM);
    if (distance < 100) {
        Serial.print("Distance in CM: ");
        Serial.println(distance);
        PublishData(distance);
        delay(1000);
        if (!client.loop()) {
            mqttconnect();
        }
    }
    delay(1000);
}
void PublishData(float temp) {
    mqttconnect();
    String payload = "{\"Alert Distance\":\"";
    payload += temp;
    payload += "\"}";
    Serial.print("Sending payload: ");
    Serial.println(payload);
    if (client.publish(publishTopic, (char*) payload.c_str())) {
        Serial.println("Publish ok");
    }
    else {
        Serial.println("Publish failed");
    }
}

void mqttconnect() {
    if (!client.connected()) {
        Serial.print("Reconnecting client to ");
        Serial.println(server);
        while (!client.connect(clientId, authMethod, token)) {
            Serial.print(".");
            delay(500);
        }
        initManagedDevice();
        Serial.println();
    }
}
void wificonnect()
{
    Serial.println();
    Serial.print("Connecting to ");

```

```

WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
void initManagedDevice() {
    if (client.subscribe(subscribetopic)) {
        Serial.println((subscribetopic));
        Serial.println("subscribe to cmd OK");
    } else {
        Serial.println("subscribe to cmd FAILED");
    }
}
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
    Serial.print("callback invoked for topic: ");
    Serial.println(subscribetopic);
    for (int i = 0; i < payloadLength; i++) {
        data3 += (char)payload[i];
    }
    Serial.println("data: " + data3);
    if (data3 == "lighton")
    {
        Serial.println(data3);
    }
    else
    {
        Serial.println(data3);
    }
    data3 = "";
}

```

EXECUTION:

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3
4 #define ECHO_GPIO 12
5 #define TRIGGER_GPIO 13
6 #define MAX_DISTANCE_CM 100 // Maximum of 5 meters
7 #include "Ultrasonic.h"
8
9 Ultrasonic ultrasonic(13, 12);
10 int distance;
11
12 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
13
14 //-----credentials of IBM Accounts-----
15
16 #define ORG "wjquinn" //IBM ORGANIZATION ID
17 #define DEVICE_TYPE "ESP32_controller" //Device type mentioned in ibm watson IoT Platform
18 #define DEVICE_ID "ESP32" //Device ID mentioned in ibm watson IoT Platform
19 #define TOKEN "1234567890" //Token
20 String data3;
21 float h, t;
22
23 //----- Customise the above values -----
24
25 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
```

Simulation

01:25:181 99%

Publish ok

Distance in CM: 57

Sending payload: {"Alert Distance":57.00}

Publish ok

Distance in CM: 57

Sending payload: {"Alert Distance":57.00}

Publish ok

OUTPUT:

Browse Action Device Types Interfaces

Add Device +

Sensor Connected ESP32_Controller Device Oct 30, 2022 4:42 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
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