

ASSIGNMENT 4

STUDENT NAME : DINESH A 19BEC202

DATE	25-10-2022
TEAM ID	PNT2022TMID15161
PROJECT NAME	Industry-Specific Intelligent fire management system

Write 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

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

WiFiClient wifiClient;

String data3;

#define ORG "c0mbt9"

#define DEVICE_TYPE "Node"

#define DEVICE_ID "1234"

#define TOKEN "987654321"

#define speed 0.034

#define led 14

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

char publishTopic[] = "iot-
2/evt/shanmugam_assignment4/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);

const int trigpin=5; 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:

i) When distance greater than 100 cm

The screenshot shows the Wokwi IDE interface. On the left, the sketch.ino file contains the following code:

```
104 if(dist>100){
105   String payload = "{\"Distance\":";
106   payload += dist;
107   payload += "}";
108
109   Serial.print("\n");
110   Serial.print("Sending payload: ");
111   Serial.println(payload);
112   if(client.publish(publishTopic, (char*) payload.c_str())) {
113     Serial.println("Publish OK");
114   } else {
115     Serial.println("Publish FAILED");
116   }
117 }
118 }
119 }
120 }
121 }
122 }
```

On the right, the simulation window shows an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. The distance is displayed as 147cm. Below the simulation, the serial monitor shows the following output:

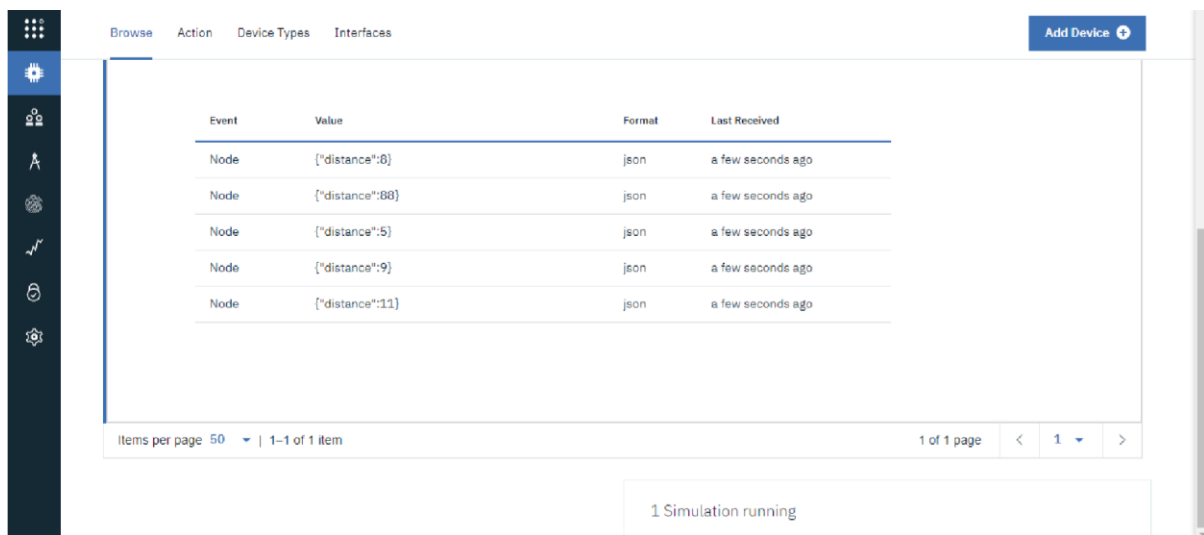
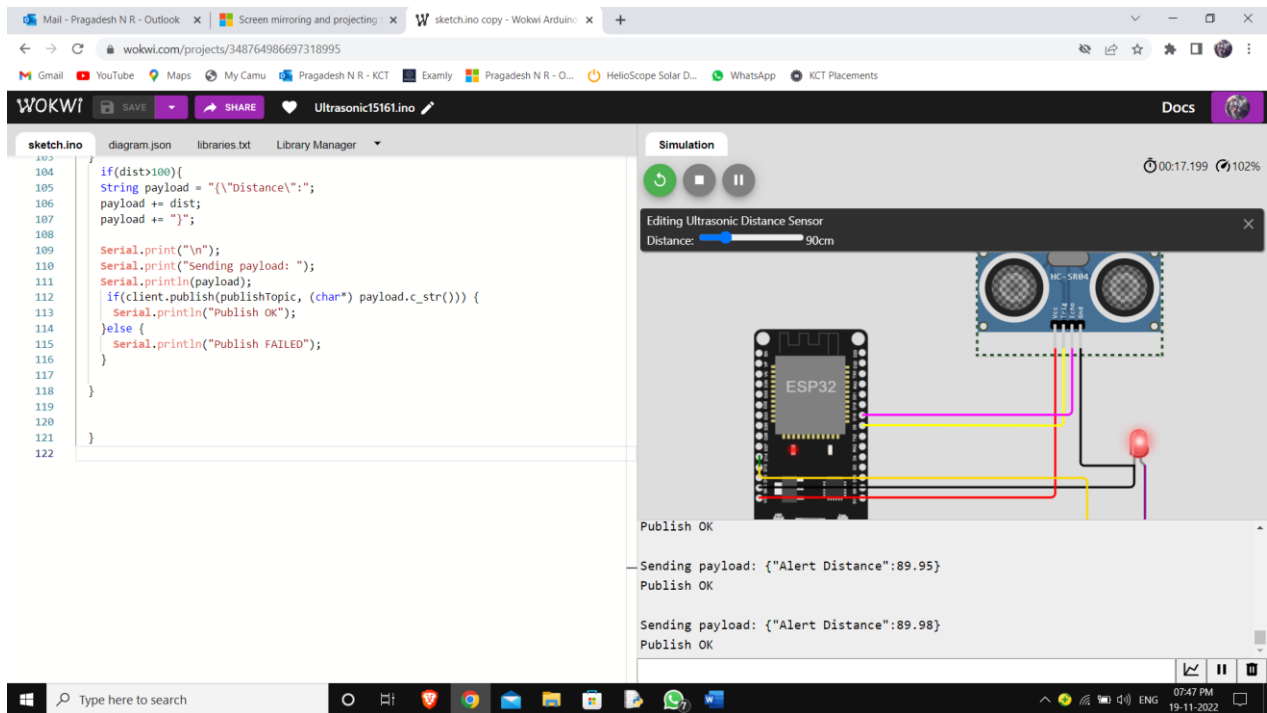
```
Publish OK
Sending payload: {"Distance":146.98}
Publish OK
Sending payload: {"Distance":146.98}
Publish OK
```

The screenshot shows the Wokwi web interface. The 'Recent Events' tab is selected, displaying a table of events. The table has four columns: Event, Value, Format, and Last Received. The events listed are:

Event	Value	Format	Last Received
Node	{"distance":144}	json	a few seconds ago
Node	{"distance":182}	json	a few seconds ago
Node	{"distance":196}	json	a few seconds ago
Node	{"distance":165}	json	a few seconds ago
Node	{"distance":164}	json	a few seconds ago

At the bottom of the interface, a status bar indicates '1 Simulation running'.

ii)When distance less than 100



WOKWI LINK -

<https://wokwi.com/projects/348764986697318995>