

## ASSIGNMENT-4

### DISTANCE DETECTION USING ULTRASONIC SENSOR

Date	28 October 2022
Team ID	PNT2022TMID17665
Name	Ram Balaji S
Student Roll Number	713319EC089
Maximum Marks	2 Marks

#### Question:

Write code and connections in wokwi for ultrasonic sensor. Whenever distance is less than 100 centimeters it should send "alert" to IBM cloud and display in device recent events

#### Code:

```
#include <WiFi.h>
#include
<PubSubClient.h>
#include <ArduinoJson.h>
WiFiClient wifiClient;
#define ORG "sdh5ix"
#define DEVICE_TYPE "nodemcu"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/status1/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);
void publishData();

const int trigpin=5;
```

```

const int echopin=19;
String command;
String data="";
String name="Alert";
String icon="";
long duration;
int dist;
void setup()
{
  Serial.begin(115200);
  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(".");
      Serial.print("*");
      delay(1000);
    }
    initManagedDevice();
    Serial.println();
  }
}

void initManagedDevice() {

```

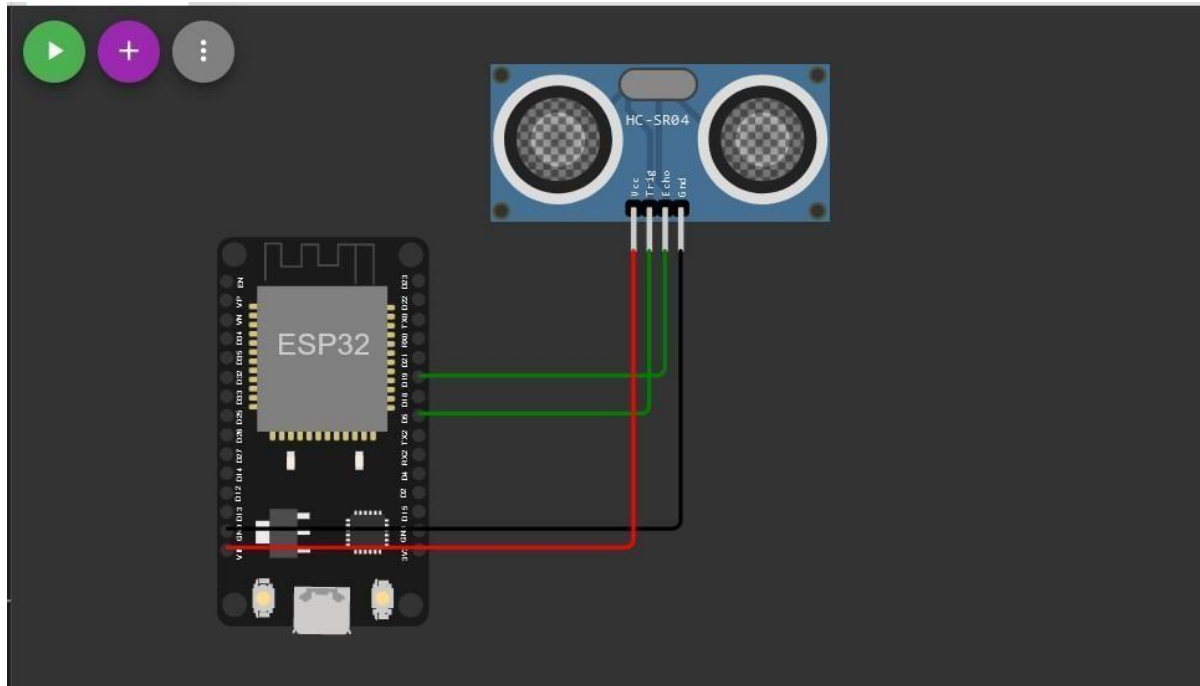
```

if (client.subscribe(topic)) {
    Serial.println(client.subscribe(topic));
    Serial.println("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){
        dist=100-dist;
        icon="Not-Crashed";
    }
    else{
        dist=0;
        icon="Crashed";
    }
    DynamicJsonDocument doc(1024);
    String payload;
    doc["Name"]=name;
    doc["Impact"]=icon;
    doc["Distance"]=dist;
    serializeJson(doc, payload);
    delay(3000);
    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");
    }
}

```

## DIAGRAM:



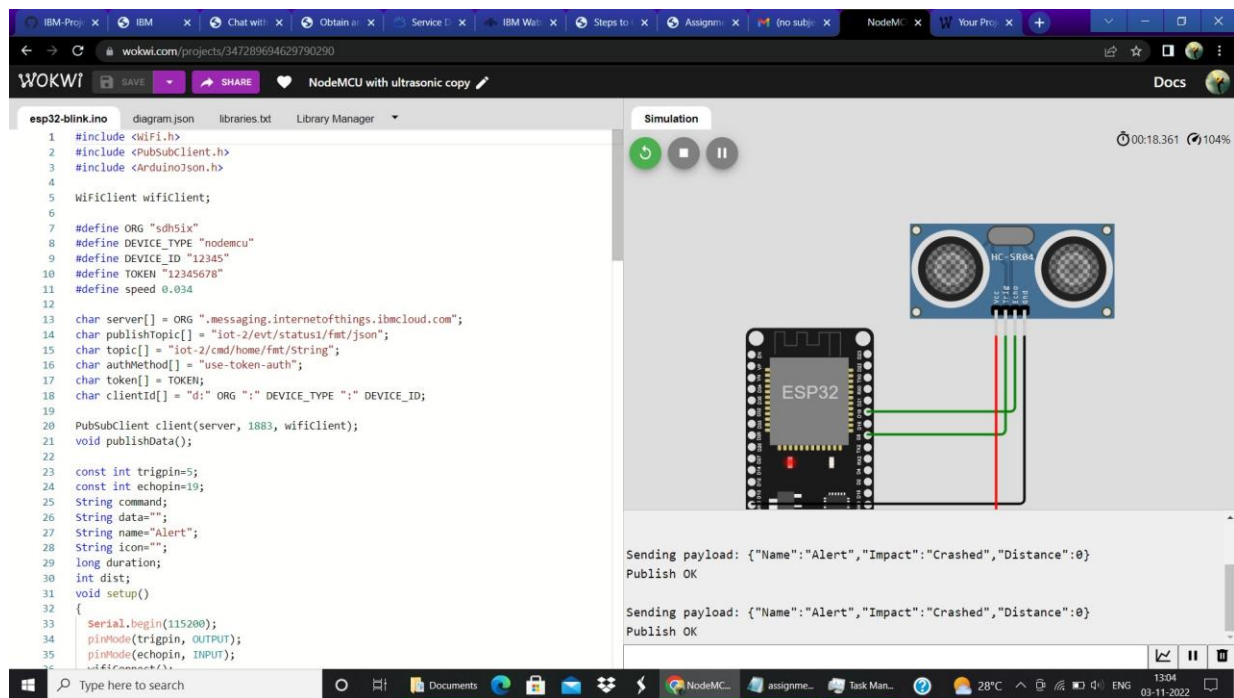
## OUTPUT:

The screenshot shows the Wokwi web IDE interface. On the left, the code for the ESP32 is displayed in the `esp32-blink.ino` file. The code includes the necessary libraries, defines the device type and ID, and sets up the connection to the IBM Cloud IoT Platform. The sensor's distance is read and published as a JSON payload.

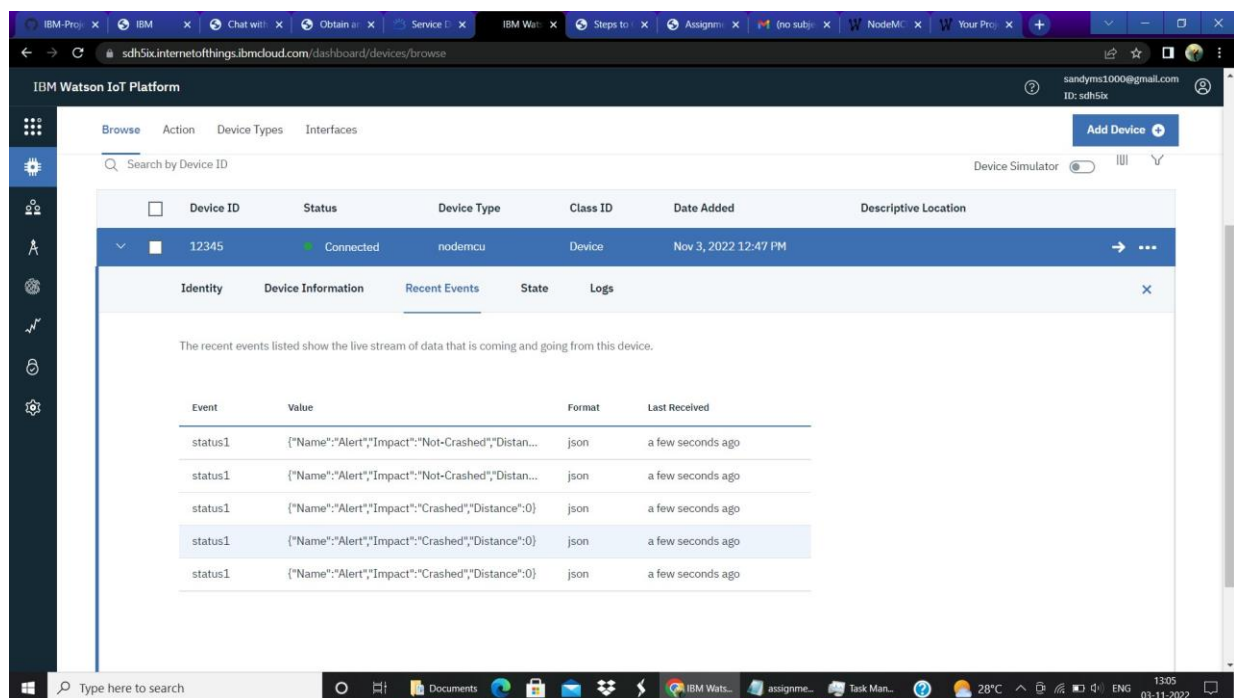
```
1 #include <WiFi.h>
2 #include <PubSubClient.h>
3 #include <ArduinoJson.h>
4
5 WiFiClient wificlient;
6
7 #define ORG "sdhSix"
8 #define DEVICE_TYPE "nodemcu"
9 #define DEVICE_ID "12345"
10 #define TOKEN "12345678"
11 #define speed 0.034
12
13 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
14 char publishTopic[] = "iot-2/evt/status/fmt/json";
15 char topic[] = "iot-2/cmd/home/fmt/string";
16 char authMethod[] = "use-token-auth";
17 char token[] = TOKEN;
18 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
19
20 PubSubClient client(server, 1883, wificlient);
21 void publishData();
22
23 const int trigpin=5;
24 const int echopin=19;
25 String command;
26 String data="";
27 String name="Alert";
28 String icon="";
29 long duration;
30 int dist;
31 void setup()
32 {
33   Serial.begin(115200);
34   pinMode(trigpin, OUTPUT);
35   pinMode(echopin, INPUT);
36   wificlient.begin();
37 }
38
39 void loop()
40 {
41   if (client.connected()) {
42     client.publish(topic, data);
43     publishData();
44   }
45   delay(1000);
46 }
```

The simulation window on the right shows the HC-SR04 sensor's distance reading as 76cm. Below the sensor, the published payload is displayed:

```
Publish OK
Sending payload: {"Name":"Alert","Impact":"Crashed","Distance":0}
Publish OK
Sending payload: {"Name":"Alert","Impact":"Not-Crashed","Distance":25}
Publish OK
```



## Data uploaded to Iot Watson Platform



<https://wokwi.com/projects/347289694629790290>