

## **ASSIGNMENT-4**

### **DISTANCE DETECTION USING ULTRASONICSENSOR**

Date	12 November 2022
Team ID	PNT2022TMID38788
Project Name	Signs With Smart Connectivity for Better Road Safety
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 devicerecent events

#### **Code:**

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

WiFiClient wifiClient;

#define ORG "9tg03j"
#define DEVICE_TYPE "RaspberryPi"
#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() {

```

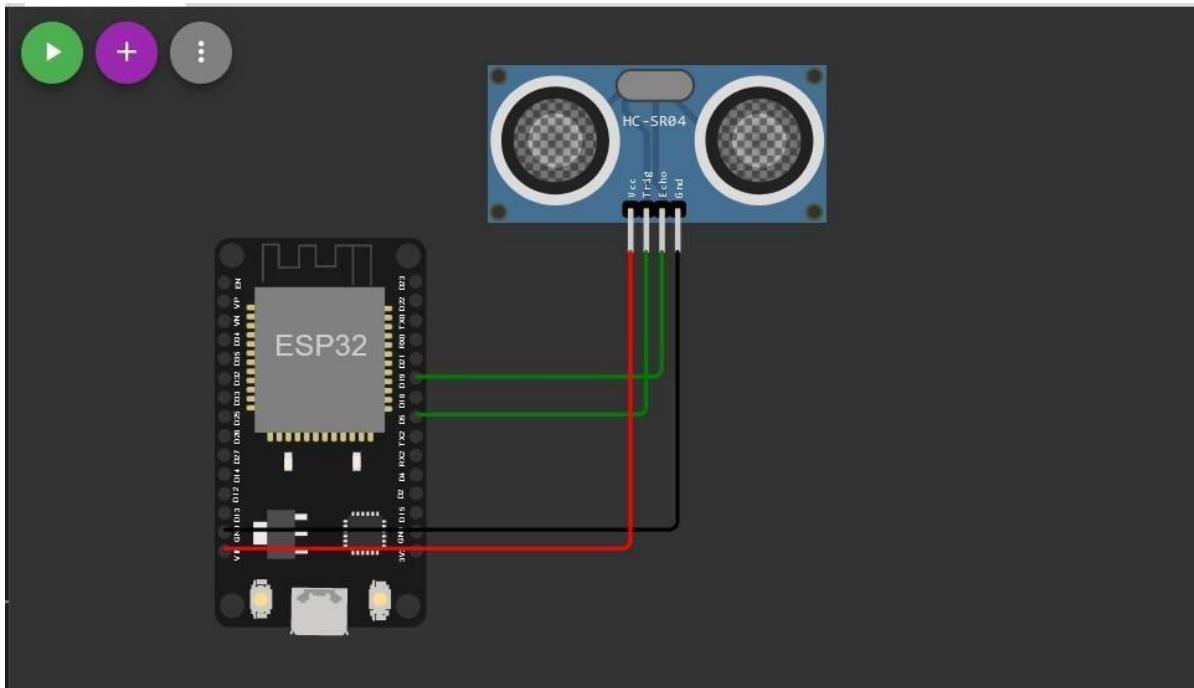
```

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:



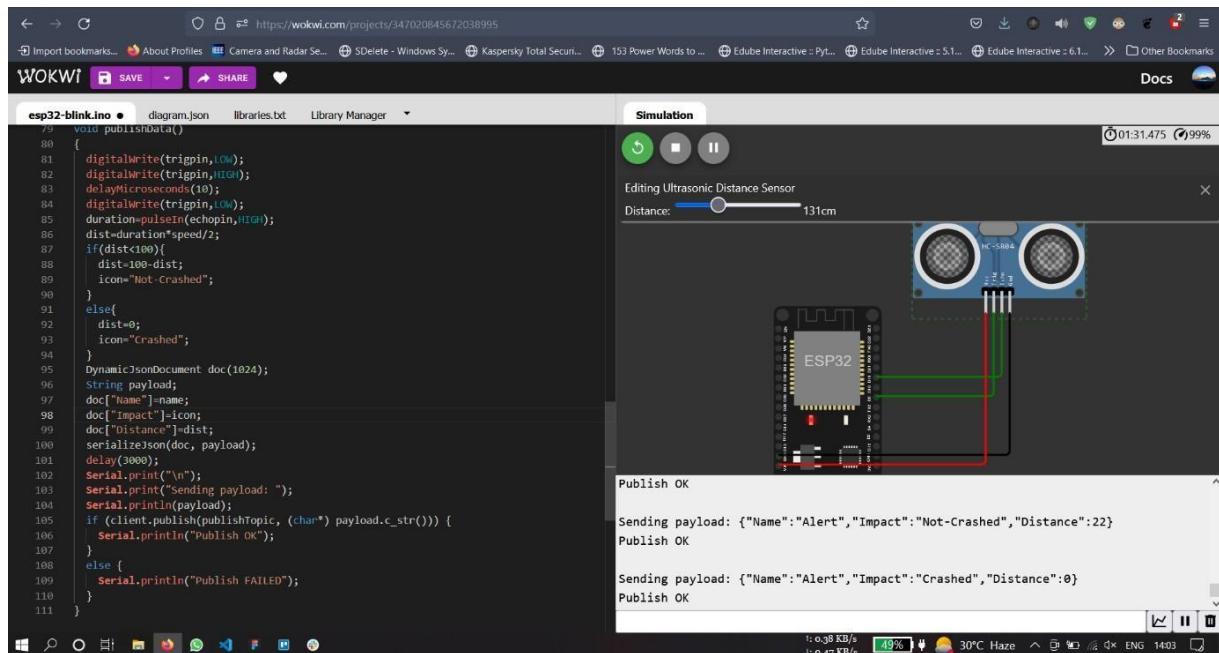
## OUTPUT:

The screenshot shows the Wokwi simulation interface. On the left is the code editor with the file 'esp32-blink.ino' open. The code implements a basic ultrasonic distance measurement loop. It toggles the trig pin (D10), measures the duration of the echo pulse, calculates the distance, and then publishes a JSON payload to a MQTT topic. The right side of the screen shows the simulation environment with the physical components and their connections. A status bar at the bottom provides system information like battery level (49%), temperature (30°C), and signal strength.

```
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");
  }
}
```

Simulation window output:

- Editing Ultrasonic Distance Sensor
- Distance: 79cm
- Publish OK
- Sending payload: {"Name": "Alert", "Impact": "Not-Crashed", "Distance": 22}
- Publish OK
- Sending payload: {"Name": "Alert", "Impact": "Not-Crashed", "Distance": 21}
- Publish OK



## Data uploaded to IoT Watson Platform

The screenshot shows the IBM Watson IoT Platform Device Drilldown page for device ID `098765432`. The left sidebar includes icons for Home, Devices, Events, Metrics, Diagnostics, and Actions. The main content area has a title "Device Drilldown - 098765432".

**Recent Events**

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"Name": "Alert", "Icon": "Trash", "FillPercent": 0}	json	a few seconds ago
event_1	{"Name": "Alert", "Icon": "Trash", "FillPercent": 0}	json	a few seconds ago
event_1	{"Name": "Alert", "Icon": "Trash", "FillPercent": 0}	json	a few seconds ago
event_1	{"Name": "Alert", "Icon": "Trash", "FillPercent": 0}	json	a few seconds ago
event_1	{"Name": "Alert", "Icon": "Trash", "FillPercent": 0}	json	a few seconds ago

A note at the bottom right says "1 Simulation running". The system tray at the bottom shows various icons and the date/time: "11/12/2022 10:38 PM".