

## ASSIGNMENT- 4

Assignment Date	27 OCT 2022
Student Name	Durga Sri R
Team ID	TNT2022TMID46736
Project Title	IoT Based Smart Crop Protection System for Agriculture
Maximum Marks	2 Marks

### 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. Upload document with wokwi share link and images of IBM cloud.

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "prahw0"
#define DEVICE_TYPE "DurgaSri"
#define DEVICE_ID "Durgasri123"
#define TOKEN "123456789"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/DurgaSri/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=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("Warning crosses 110cm -- it automaticaly of the loop");
            digitalWrite(led, HIGH);
        }
    }

    if(dist>101 && dist<111){
        String payload = "{\"Normal Distance\":\"";
        payload += dist;
        payload += "\"}";

        Serial.print("\n");
        Serial.print("Sending payload: ");
        Serial.println(payload);

    }

}

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++){
        dist += (char)payload[i];
    }
}

```

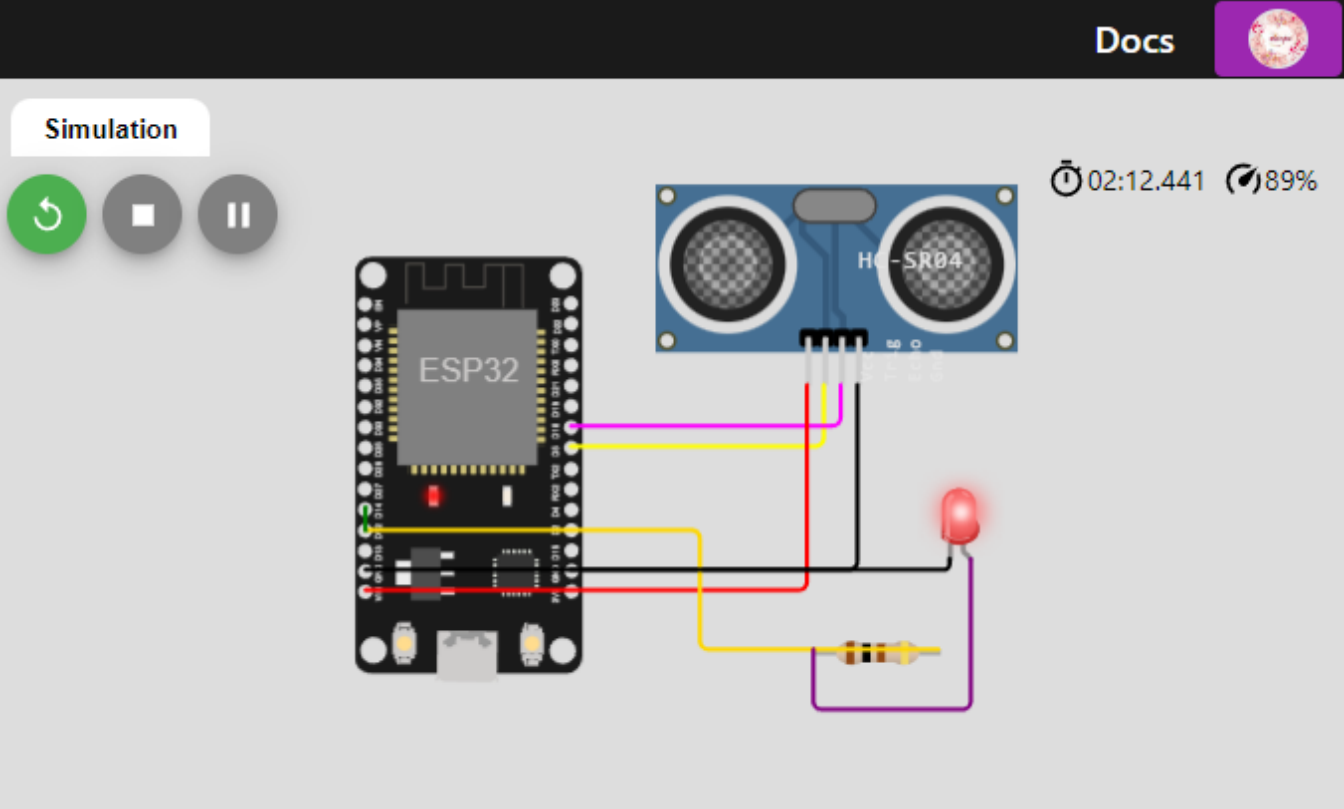
```

Serial.println("data:" + data3);

if(data3=="lighton"){
  Serial.println(data3);
  digitalWrite(led,HIGH);
}
data3="";
}

```

## OUTPUT:



The screenshot shows a Tinkercad simulation environment. At the top, there's a 'Docs' button and a user profile icon. Below that, a 'Simulation' tab is active, with buttons for 'Run' (a green circle with a refresh icon), 'Stop' (a grey square), and 'Pause' (a grey circle with two vertical bars). To the right of these buttons, a timer shows '02:12.441' and a progress indicator shows '89%'. The main workspace contains an ESP32 microcontroller board, an HC-SR04 ultrasonic sensor, and an LED. Wires connect the sensor's VCC to the ESP32's 5V pin, GND to GND, and Trig to D4. The LED's anode is connected to D4, its cathode to GND, and a resistor is connected between D4 and the cathode. The console at the bottom displays the following output:

```

Sending payload: {"Alert Distance":89.95}
Warning crosses 110cm -- it automatically of the loop

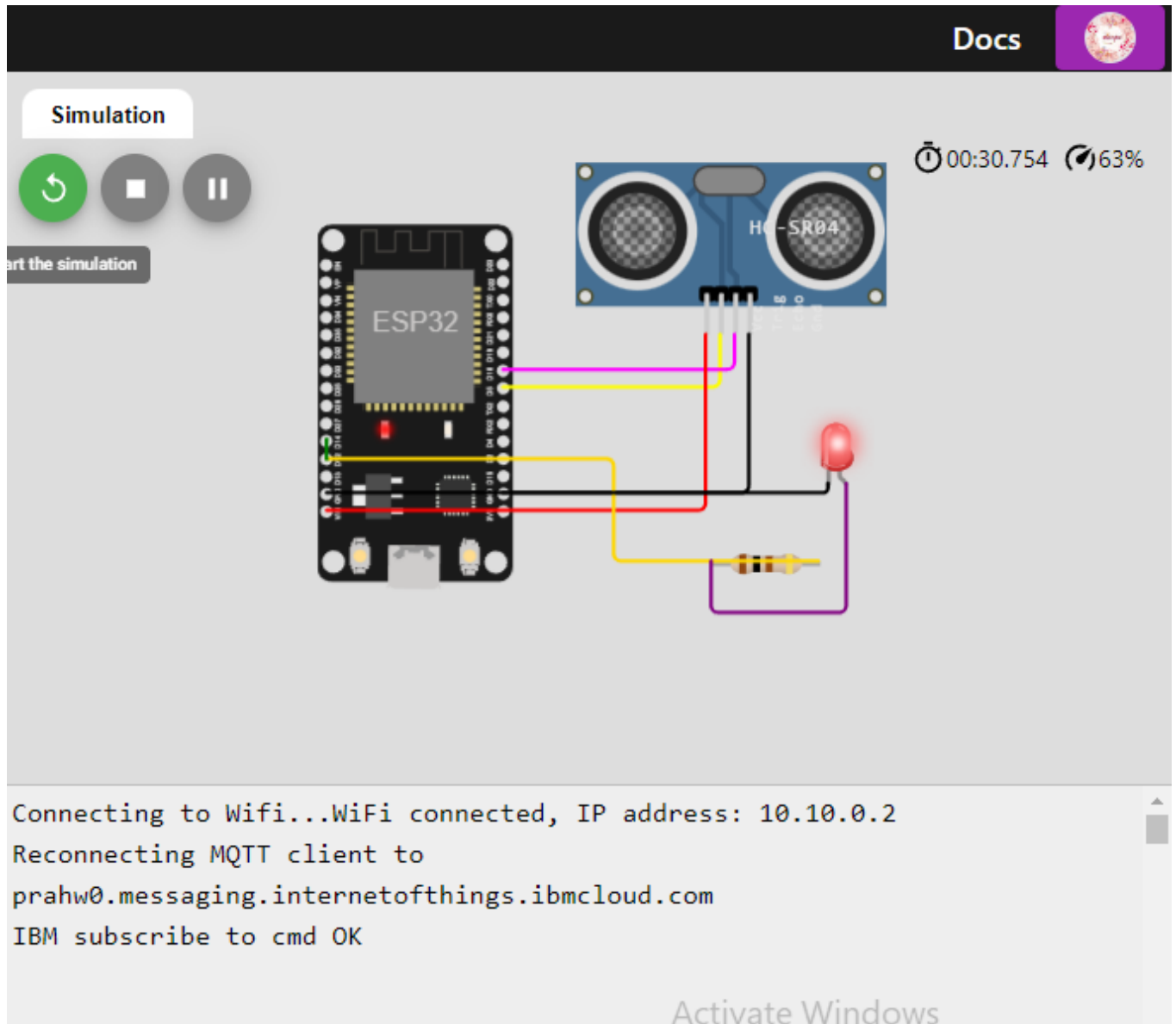
Sending payload: {"Alert Distance":89.95}
Warning crosses 110cm -- it automatically of the loop

Sending payload: {"Alert Distance":89.95}
Warning crosses 110cm -- it automatically of the loop

```

At the bottom right of the console, there is a watermark that says 'Activate Windows Go to Settings to activate Windows.' and a set of icons for 'Save', 'Run', and 'Delete'.

When the distance is less than 100cm alertmessage will appear in the IBM cloud.



While Distance is greater than 100cm there is no alert message in the IBM cloud.

🔍 Search by Device ID

<input type="checkbox"/>	Device ID	Status	Device Type
▼ <input type="checkbox"/>	Durgasri123	● Connected	DurgaSri
Identity		Device Information	Recent Events
State			

Device ID	Durgasri123
Device Type	DurgaSri
Date Added	2 Nov 2022 06:30
Added By	durgaram1912002@gmail.com
Connection Status	<b>Connected</b> Connection Time: 2 Nov 2022 06:59 Client Address: 145.40.94.93 Insecure

## IBM CLOUD OUTPUT:

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added
▼ <input type="checkbox"/>	Durgasri123	● Connected	DurgaSri	Device	2 Nov 2022 06:30
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	
DurgaSri		{"Alert Distance":89.96}	json	a few seconds ago	
DurgaSri		{"Alert Distance":89.74}	json	a few seconds ago	

## WOKWI URL:

<https://wokwi.com/projects/347233575147405908>