

# ASSIGNMENT-3:

## DISTANCE DETECTOR USING WOKWI AND UPLOAD DATAS IN IBM PLATFORM

Done by  
B. Gowri Shankar  
III-Year  
Department of ECE  
AVS Engineering College

---

### ASSIGNMENT THEME:

To Built Wokwi product, use ultrasonic sensor and detect the distance from the object. whenever distance is less than 100cm upload the value to the ibm cloud.in recent device events as upload to data from wokwi.

**For ex:** Distance is 20cm. Upload the 20 value to the ibm cloud in recent events in the iot platform device.

### PROJECT DESCRIPTION:

For Building a product in wokwi, to detect the distance of an object. Here, we use Ultrasonic sensor in such a way the distance is calculated by the expression as

```
//get duration
duration=pulseIn(echoPin,HIGH);
```

```
//calculate distance
distance=(duration/2)*0.0343;
```

Whenever the distance of an object is less than 100cm, the LED will turned ON and upload the values in a ibm cloud platform and we will observe it in recent events of the same platform. Whereas, whenever the distance is above 100cm,

There is no issue will occur. So we ignore this values for uploading on ibm platform.

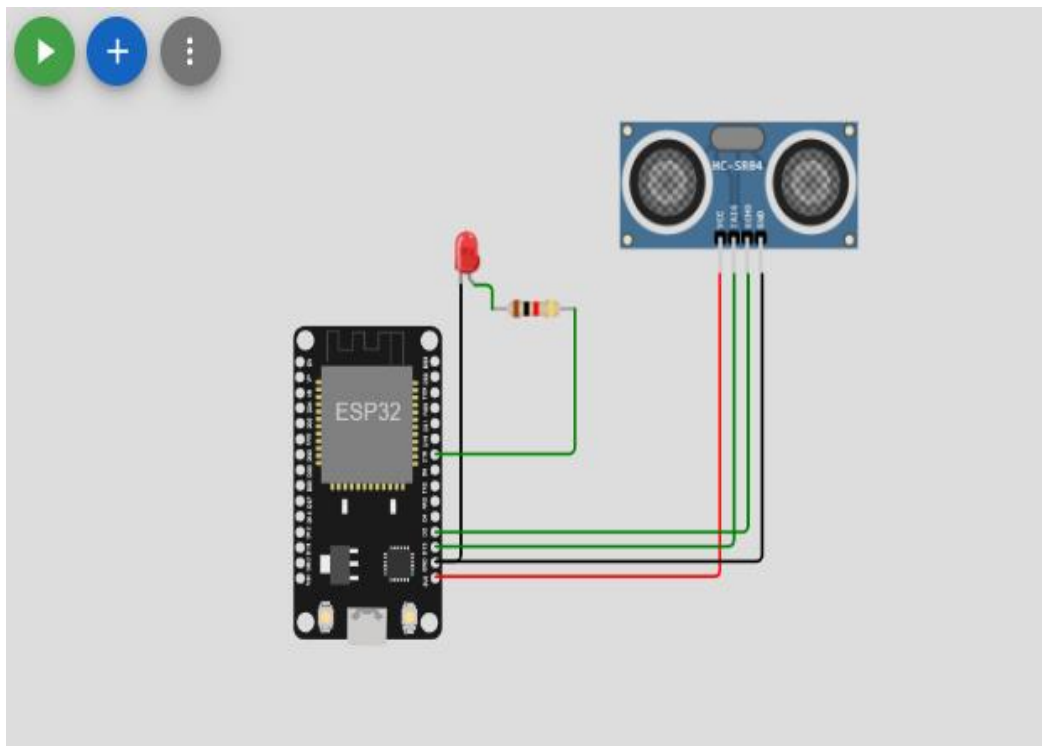
### **APPARATUS REQUIRED:**

The following are the apparatus required for building this assignment. Such as

- Wokwi Stimulator
- ESP32 (A microcontroller)
- LED
- Ultrasonic Sensor
- Connecting wires

### **CONNECTION:**

The Circuit Connections which I made for this assignment are shown in figure below.



## CODING:

```
//ASSIGNMENT-3:
//To Built Wokwi product, use ultrasonic sensor and detect the
distance from the object. whenever distance is less than 100cm upload
the value to the ibm cloud.in recent device events as upload to data
from wokwi.

#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQTT
#define triggerPin 15 // what pin we're connected to
#define echoPin 2
#define LED 18

double duration, distance;

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

//-----credentials of IBM Accounts-----

#define ORG "4566ei">//IBM ORGANITION ID
#define DEVICE_TYPE "abcdef">//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "123456">//Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "12345678" //Token
String data3;

//----- Customise the above values -----
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";//
Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type
of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";//
cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----
WiFiClient wifiClient; // creating the instance for wificlient
```

```
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand
wificredential
```

```
void setup()// configuring the ESP32
```

```
{
  Serial.begin(9600);
  pinMode(trigerPin,OUTPUT);
  pinMode(echoPin,INPUT);
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}
```

```
void loop()// Recursive Function
```

```
{
  digitalWrite(trigerPin,LOW);
  delayMicroseconds(2);
  digitalWrite(trigerPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigerPin,LOW);
  delayMicroseconds(2);

  //get duration
  duration=pulseIn(echoPin,HIGH);

  //calculate distance
  distance=(duration/2)*0.0343;

  //consider minimum distance as 100cm
  if(distance<100){
    digitalWrite(LED,HIGH);
    delayMicroseconds(30000);
    Serial.println("The distance is less than 100cm");
    digitalWrite(LED,LOW);
    delayMicroseconds(30000);
  }
  else{
    digitalWrite(LED,LOW);
    Serial.println("The distance is greater than 100cm");
    Serial.println("-----");
  }
}
```

```

    }

    PublishData(distance);
    delay(1000);
    if (!client.loop()) {
        mqttconnect();
    }
}

/*.....retrieving to
Cloud.....*/

void PublishData(float distance) {
    mqttconnect();//function call for connecting to ibm
    // creating the String in in form Json to update the data to ibm
    cloud

    if(distance<100){

        String payload = "{\"distance\":";
        payload += distance;
        payload += "}";

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

        if (client.publish(publishTopic, (char*) payload.c_str())) {
            Serial.println("Publish ok");// if it sucessfully upload data on
            the cloud then it will print publish ok in Serial monitor or else it
            will print publish failed
        } else {
            Serial.println("Publish failed");
        }

        Serial.println("-----");
    }
}

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() //function defination for wificonnect
{
    Serial.println();
    Serial.print("Connecting to ");

    WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to
    establish the connection
    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++) {
        //Serial.print((char)payload[i]);
        data3 += (char)payload[i];
    }
    Serial.println("data: "+ data3);
}

```

```

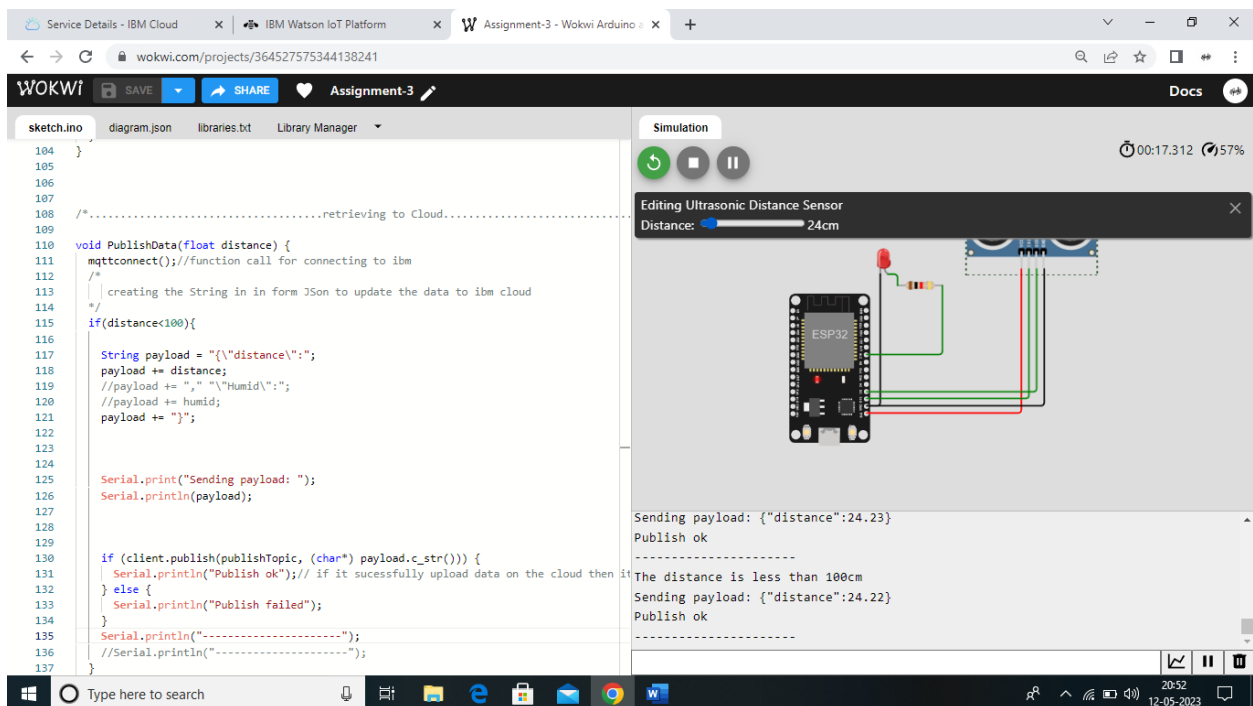
    if(data3=="lighton")
    {
Serial.println(data3);
digitalWrite(LED,HIGH);
    }
    else
    {
Serial.println(data3);
digitalWrite(LED,LOW);
    }
    data3="";
}

```

This is the source Code of my assignment for detecting the distance using ultrasonic sensor and uploading the distance values (if <100cm) in an IBM-Cloud IOT Platform.

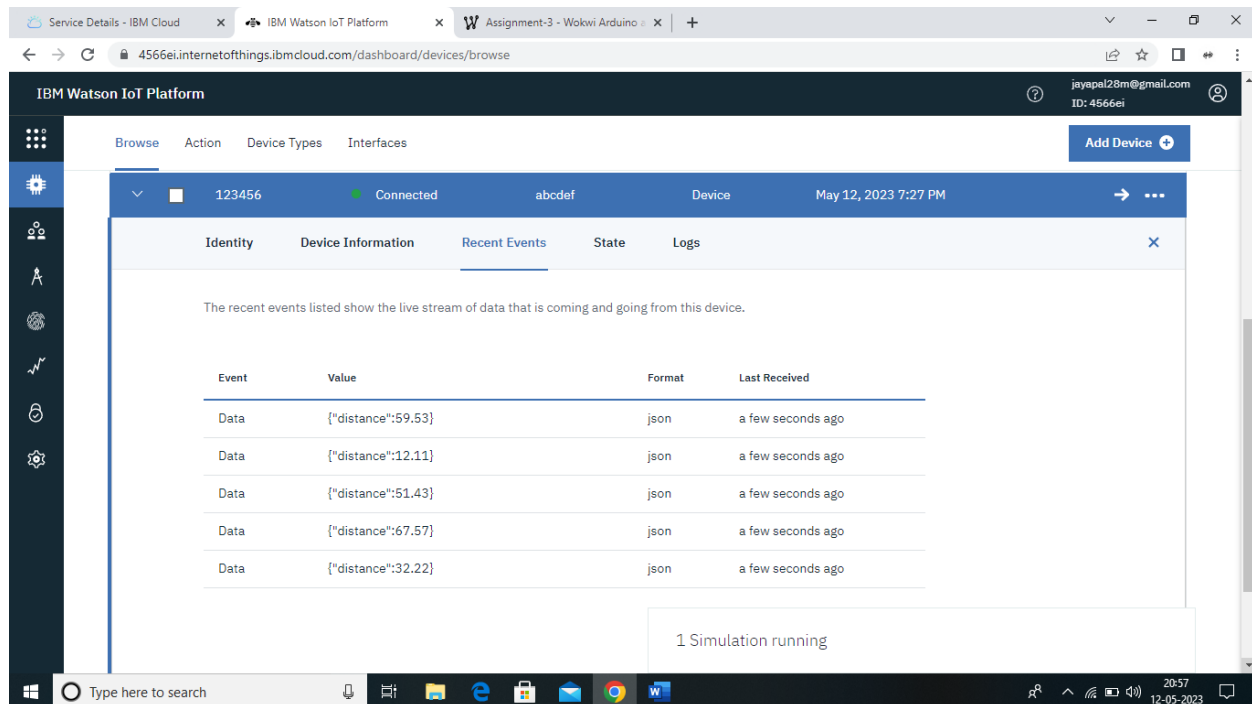
## OUTPUT:

The wokwi output for this assignment is shown below.



## **IBM RECENT EVENTS:**

The recent events for this assignment is noted in an ibm cloud iot platform is shown in figure below.



## **REFERENCE LINK:**

For your reference, I attached the link of my assignment here.

<https://wokwi.com/projects/364527575344138241>

## **RESULT:**

Thus, I have successfully completed my assignment "DISTANCE DETECTION USING WOKWI and UPLOAD DATAS IN RECENT EVENTS ON AN IBM PLATFORM".