

ASSIGNMENT 4

Distance Detection Using Ultrasonic Sensor Assignment	5 November 2022
Date	
Student Name	B Sudha Priyadharshini
Student Roll Number	710019106045
Maximum Marks	2 Marks

WOKWI CODE:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQTT
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "dks66l"//IBM ORGANITION ID
#define DEVICE_TYPE "Sudha"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "45"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "sudha2002@" //Token
String data3;
float dist;
//----- 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/test/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
int LED = 4;
```

```
int trig = 5;

int echo = 18;

void setup()
{
  Serial.begin(115200);
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT);
  pinMode(LED, OUTPUT);
  delay(10);
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

  digitalWrite(trig,LOW);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW);
  float dur = pulseIn(echo,HIGH);
  float dist = (dur * 0.0343)/2;
  Serial.print ("Distancein cm");
  Serial.println(dist);
  PublishData(dist);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}
```

```

/*.....retrieving to Cloud.....*/
void PublishData(float dist) {
    mqttconnect();//function call for connecting to ibm
    /*
        creating the String in in form JSon to update the data to ibm cloud
    */
    String object;
    if (dist <100)
    {
        digitalWrite(LED,HIGH);
        Serial.println("object is near");
        object = "Near";
    }
    else
    {
        digitalWrite(LED,LOW);
        Serial.println("no object found");
        object = "No";
    }

    String payload = "{\"distance\":";
    payload += dist;
    payload += "," "\"object\":";
    payload += object;
    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");

    }

}

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
{

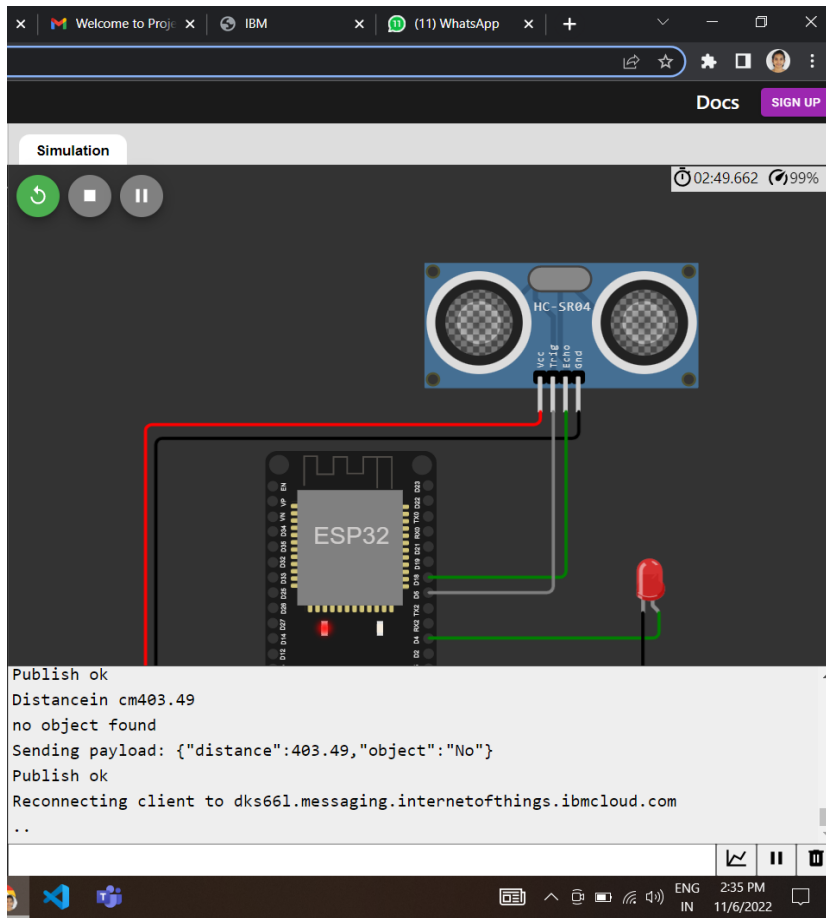
    Serial.println();

    Serial.print("Connecting to ");

```

WOKWI LINK: <https://wokwi.com/projects/347567837551788626>

WOKWI CIRCUIT DIAGRAM AND WOKWI OUTPUT:



The image shows a Wokwi simulation environment. At the top, there's a browser-like interface with tabs for 'Welcome to Proj...', 'IBM', and '(11) WhatsApp'. Below this is a 'Docs' section with a 'SIGN UP' button. The main area is labeled 'Simulation' and contains a circuit diagram. The circuit consists of an ESP32 microcontroller, an HC-SR04 ultrasonic sensor, and a red LED. The sensor is connected to the ESP32 via four wires: VCC (red), GND (black), Trig (green), and Echo (blue). The red LED is connected to the ESP32's GND and a digital output pin. The terminal at the bottom displays the following output:

```
Publish ok
Distancein cm403.49
no object found
Sending payload: {"distance":403.49,"object":"No"}
Publish ok
Reconnecting client to dks661.messaging.internetofthings.ibmcloud.com
..
```

The bottom status bar shows the system clock as 2:35 PM on 11/6/2022, along with language settings (ENG, IN) and network status.

IBM WATSON OUTPUT:

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area shows a table of devices. One device, with ID '45', is selected, and its details are shown in a modal window. The modal window has tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, displaying a list of events. A 'Device Simulator' toggle is visible in the top right corner of the modal window. A status bar at the bottom of the modal window indicates '1 Simulation running'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
45	Disconnected	Sudha	Device	Nov 4, 2022 11:16 AM	

Event	Value	Format	Last Received
event_1	{\"randomNumber\":91}	json	a few seconds ago
event_1	{\"randomNumber\":68}	json	a few seconds ago
Data	{\"distance\":403.49,\"object\":\"No\"}	json	a few seconds ago
event_1	{\"randomNumber\":9}	json	a few seconds ago
Data	{\"distance\":403.49,\"object\":\"No\"}	json	a few seconds ago

1 Simulation running