

Distance Detection Using 6 November 2022

Ultrasonic Sensor Assignment

Date

Student Name Naveen Kumar T

Student Roll Number 710019106302

Maximum Marks 2 Marks

ASSIGNMENT 4

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 " qz2oyu "//IBM ORGANITION ID

#define DEVICE_TYPE " Naveenkumar "//Device type mentioned in ibm watson IOT Platform

#define DEVICE_ID "1)//Device ID mentioned in ibm watson IOT Platform

#define TOKEN " Naveen@12" //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 ");

    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=="Near")
    // {
    // Serial.println(data3);
}

```

```
// digitalWrite(LED,HIGH);
```

```
// }
```

```
// else
```

```
// {
```

```
// Serial.println(data3);
```

```
// digitalWrite(LED,LOW);
```

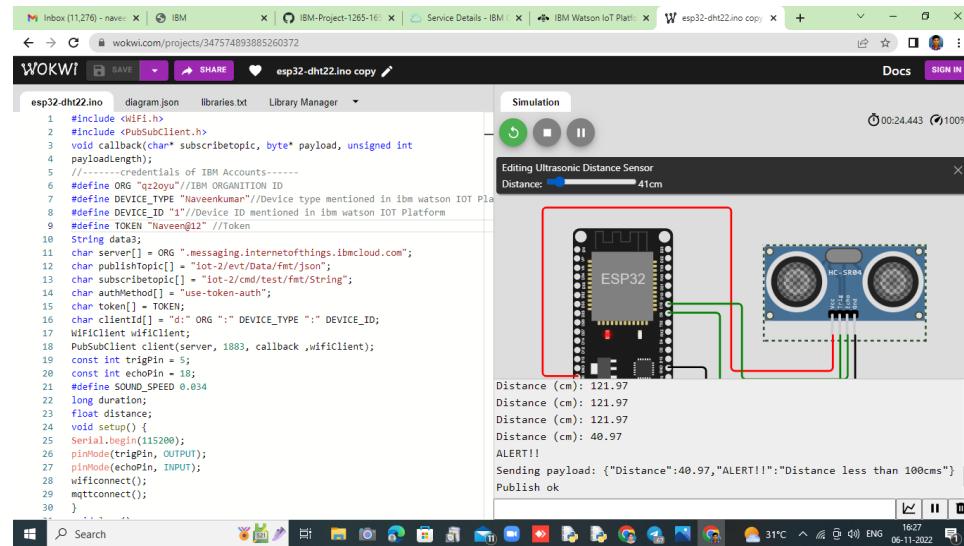
```
// }
```

```
data3="";
```

```
}
```

WOKWI LINK: <https://wokwi.com/projects/347574893885260372>

WOKWI CIRCUIT DIAGRAM AND WOKWI OUTPUT:



IBM WATSON OUTPUT:

The screenshot shows the IBM Watson IoT Platform interface. At the top, there are several tabs: 'Inbox (11,276) - naveen', 'IBM', 'IBM-Project-1265-16', 'Service Details - IBM', 'IBM Watson IoT Platf...', and 'esp32-dht22.ino copy'. The main window is titled 'IBM Watson IoT Platform' and shows a single device named 'Naveenkumar' connected at 4:21 PM on Nov 6, 2022. The device ID is 710019106302@smartinternz.com and the ID is qz2oyu. The interface includes a sidebar with various icons and tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. The 'Recent Events' tab is selected, displaying a table of recent data transmissions. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The data entries are as follows:

Event	Value	Format	Last Received
data	[{"randomNumber":57}]	json	a few seconds ago
data	[{"randomNumber":98}]	json	a few seconds ago
data	[{"randomNumber":62}]	json	a few seconds ago
Data	[{"Distance":40.97,"ALERT!!":"Distance less than ...}]	json	a few seconds ago
Data	[{"Distance":40.97,"ALERT!!":"Distance less than ...}]	json	a few seconds ago

A message box at the bottom left says '1 Simulation running'. The taskbar at the bottom shows various application icons and the system clock indicating 16:28 on 06-11-2022.