

NAME:RIBIN RAPPAI

DEPARTMENT:MEDICAL ELECTRONICS

YEAR:3RD YEAR

COLLEGE:SENGUNTHAR COLLEGE OF ENGINEERING

GROUP:GROUP-3

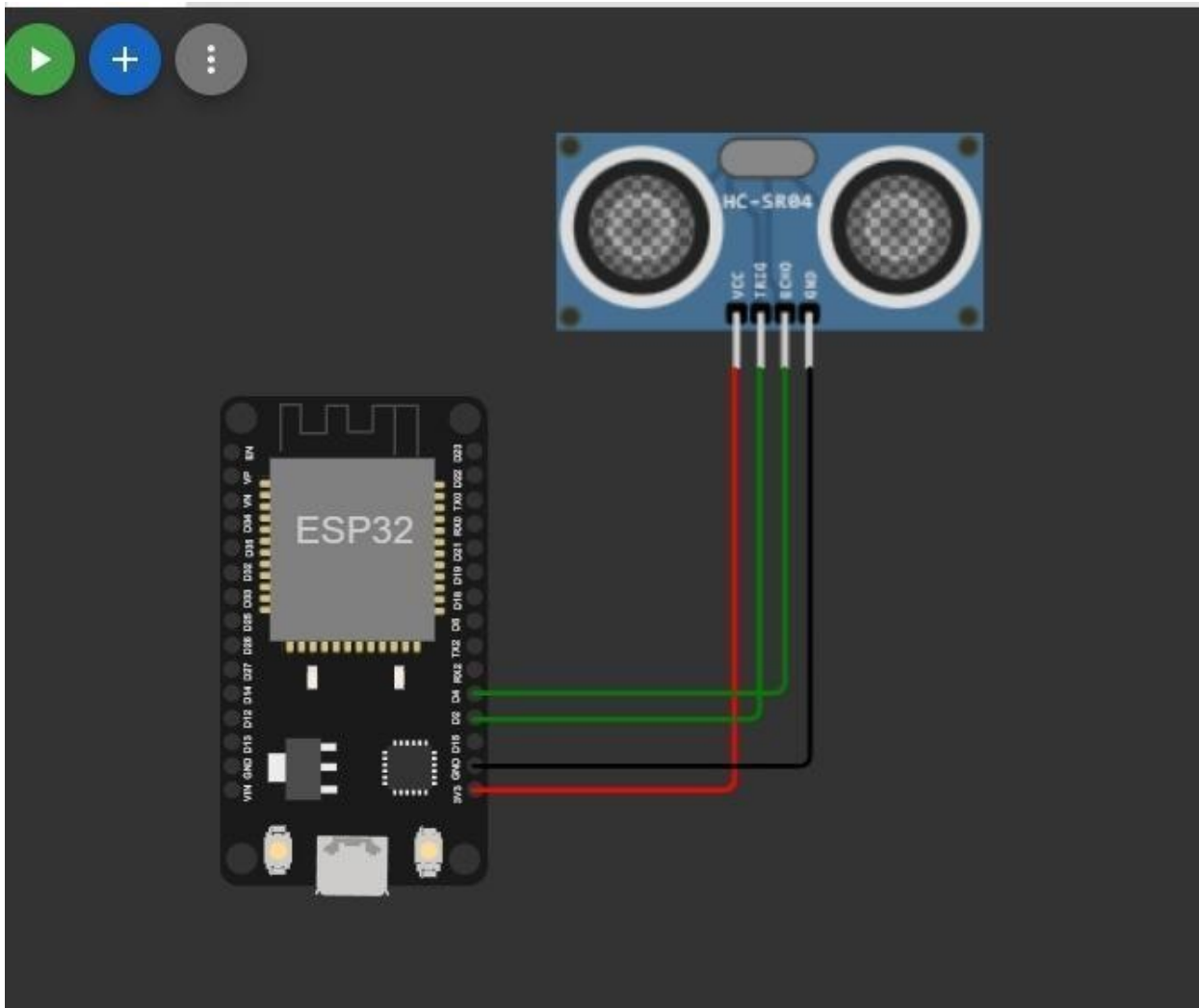
NMTEAMID:NM2023TMID13573

**PROFESSIONAL READINESS
FORINNOVATION,EMPLOYABILITY
ANDENTERPRENEURSHIP**

INTERNETOFTHINGS

ASSIGNMENT-3

Wokwisimulation:



Code:

```
#include<WiFi.h>//libraryforwifi
#include<PubSubClient.h>//libraryforMQ
#include"Ultrasonic.h"
Ultrasonicultrasonic(2,4)
;floatdistance;

voidcallback(char*subscribetopic, byte*payload, unsignedintpayloadLength);

//-----credentialsofIBMAccounts-----

#defineORG"sptrws"//IBMORGANITIONID
#defineDEVICE_TYPE"abcd"//DevicetypementionedinibmwatsonIOTPlatform
#define DEVICE_ID "1234" //Device ID mentioned in ibm watson IOT
Platform#defineTOKEN "12345678"//Token
Stringdata3;
//floath,t;
```

```

//-----Customisetheabovevalues -----
charserver[]="ORG".messaging.internetofthings.ibmcloud.com";//ServerName
char publishTopic[]="iot-2/evt/Data/fmt/json";//topic
nameandtypeofeventperformandformatinwhichdatatobesend
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd REPRESENT command
type ANDCOMMANDISTESTOFFORMATSTRING
charauthMethod[]="use-token-
auth";//authenticationmethodchartoken[]=TOKEN;
charclientId[]="d:"ORG":"DEVICE_TYPE ":"DEVICE_ID";//clientid

//_____
WiFiClientwifiClient;//creatingtheinstanceforwificlient
PubSubClientclient(server,1883,callback,wifiClient);//callingthepredefinedclientidbypassingpara
meterlikeserverid,portand wificredential
voidsetup();//configureingtheESP32
{
    Serial.begin(115200);

    delay(10);Seri
al.println();wif
iconnect();mq
ttconnect();
}

voidloop();//RecursiveFunction
{

    distance=ultrasonic.read(CM);

    Serial.print("DistanceinCM:");
    Serial.println(distance);delay(
1000);

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

/.....retrievingtoCloud...../

void PublishData(float distance)
{mqttconnect();//functioncallforconnectingtoibm
/*
    creatingtheStringininformJSontoupdatethe datatoibmcloud
*/
Stringpayload="{\"distance\":";

```

```

payload +=
distance;payload+
= "};

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

if(client.publish(publishTopic,(char*)payload.c_str())){
    Serial.println("Publishok");//ifitsuccessfullyuploaddataonthecloudthenit
willprintpublishokinSerialmonitororelseitwillprintpublishfailed
}else{
    Serial.println("Publishfailed");
}

}

voidmqttconnect(){
    if
    (!client.connected()){Serial.print("R
econnectingclientto");Serial.println
(server);
    while(!client.connect(clientId,authMethod,token)){Serial.
    print(". ");
    delay(500);
    }

    initManagedDevice()
    ;Serial.println();
}

}

voidwificonnect()//functiondefinationforwificonnect
{
    Serial.println();Serial.print("
Connectingto");

WiFi.begin("Wokwi-GUEST","",6);//passingthewificredentialstoestablishtheconnection

while(WiFi.status()!=WL_CONNECTED){de
    lay(500);
    Serial.print(". ");
}
Serial.println("");Serial.println("
WiFi
connected");Serial.println("IP
address:
");Serial.println(WiFi.localIP());
}

voidinitManagedDevice(){
    if (client.subscribe(subscribetopic))
    {Serial.println((subscribetopic));Se

```

```
rial.println("subscribetocmdOK");  
}else{  
  Serial.println("subscribetocmdFAILED");
```

```

}
}

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); data3 = "";

}

```

Output in wokwi:

The screenshot displays the Wokwi IoT simulator interface. On the left, the 'sketch.ino' editor shows the following code:

```

1 #include <Wifi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include "Ultrasonic.h"
4 Ultrasonic ultrasonic(2, 4);
5 float distance;
6
7
8 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
9
10 //-----credentials of IBM Accounts-----
11
12 #define ORG "sptrws" //IBM ORGANIZATION ID
13 #define DEVICE_TYPE "abcd" //Device type mentioned in ibm watson IOT Platform
14 #define DEVICE_ID "1234" //Device ID mentioned in ibm watson IOT Platform
15 #define TOKEN "12345678" //Token
16 String data3;
17 //float h, t;
18
19
20 //----- Customise the above values -----
21 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
22 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
23 char subscribetopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command
24 char authMethod[] = "use-token-auth"; // authentication method
25 char token[] = TOKEN;
26 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
27
28

```

On the right, the 'Simulation' window shows a visual representation of an ESP32 microcontroller connected to an HC-SR04 ultrasonic sensor. Below the simulation, a terminal log displays the following output:

```

Distance in CM: 357.00
Sending payload: {"distance":357.00}
Publish ok
Distance in CM: 357.00
Sending payload: {"distance":357.00}
Publish ok

```

Output in ibmcloud:

IBM Watson IoT Platform

mohanrajs.md20@scew.org
ID: sptwrs

Browse

Action

Device Types

Interfaces

Add Device +

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

Event	Value	Format	Last Received
Data	{"distance":357}	json	a few seconds ago
Data	{"distance":357}	json	a few seconds ago
Data	{"distance":357}	json	a few seconds ago
Data	{"distance":357}	json	a few seconds ago
Data	{"distance":357}	json	a few seconds ago