NAME : SNEHA BAI K

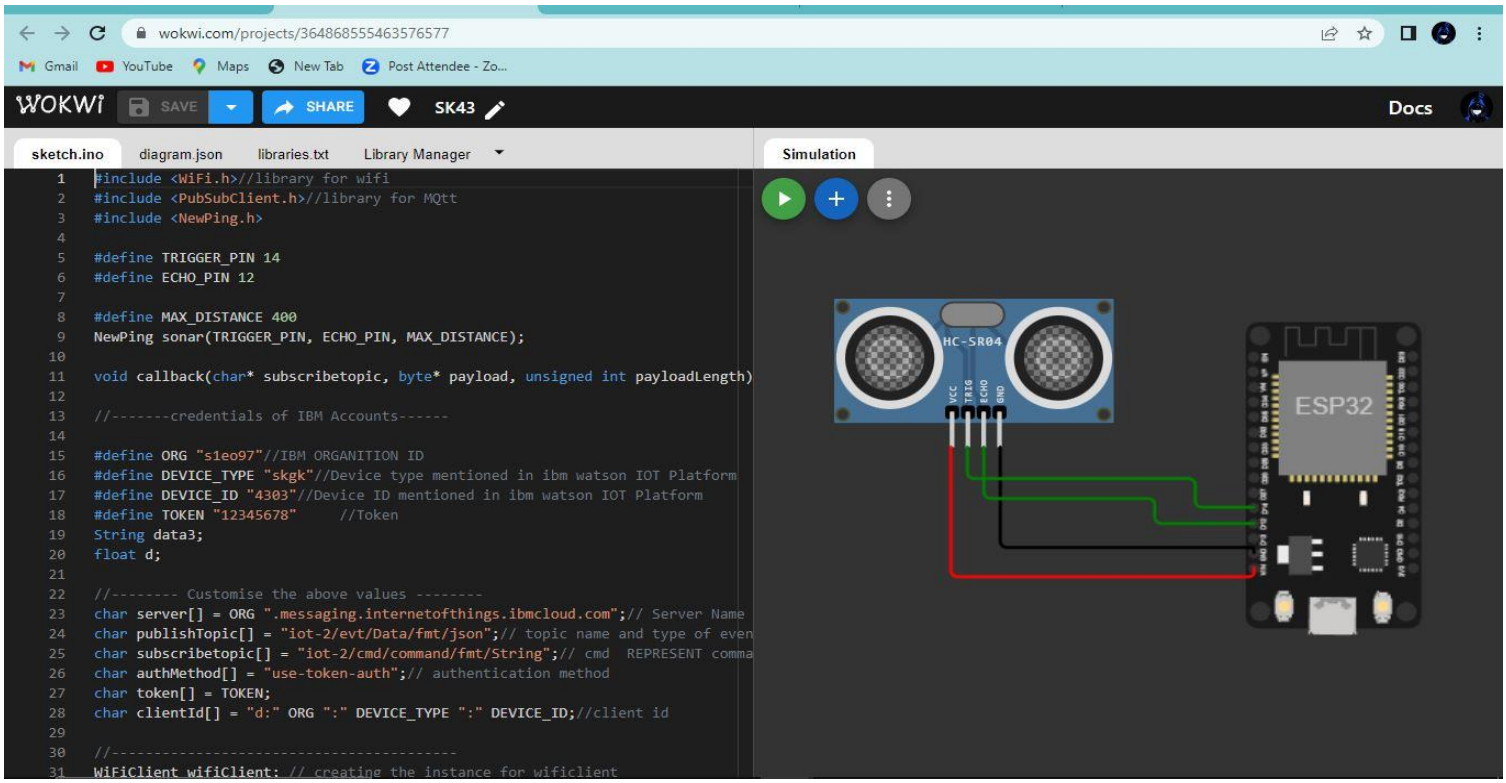
NAAN MUDHALVAN ID : au820420106043

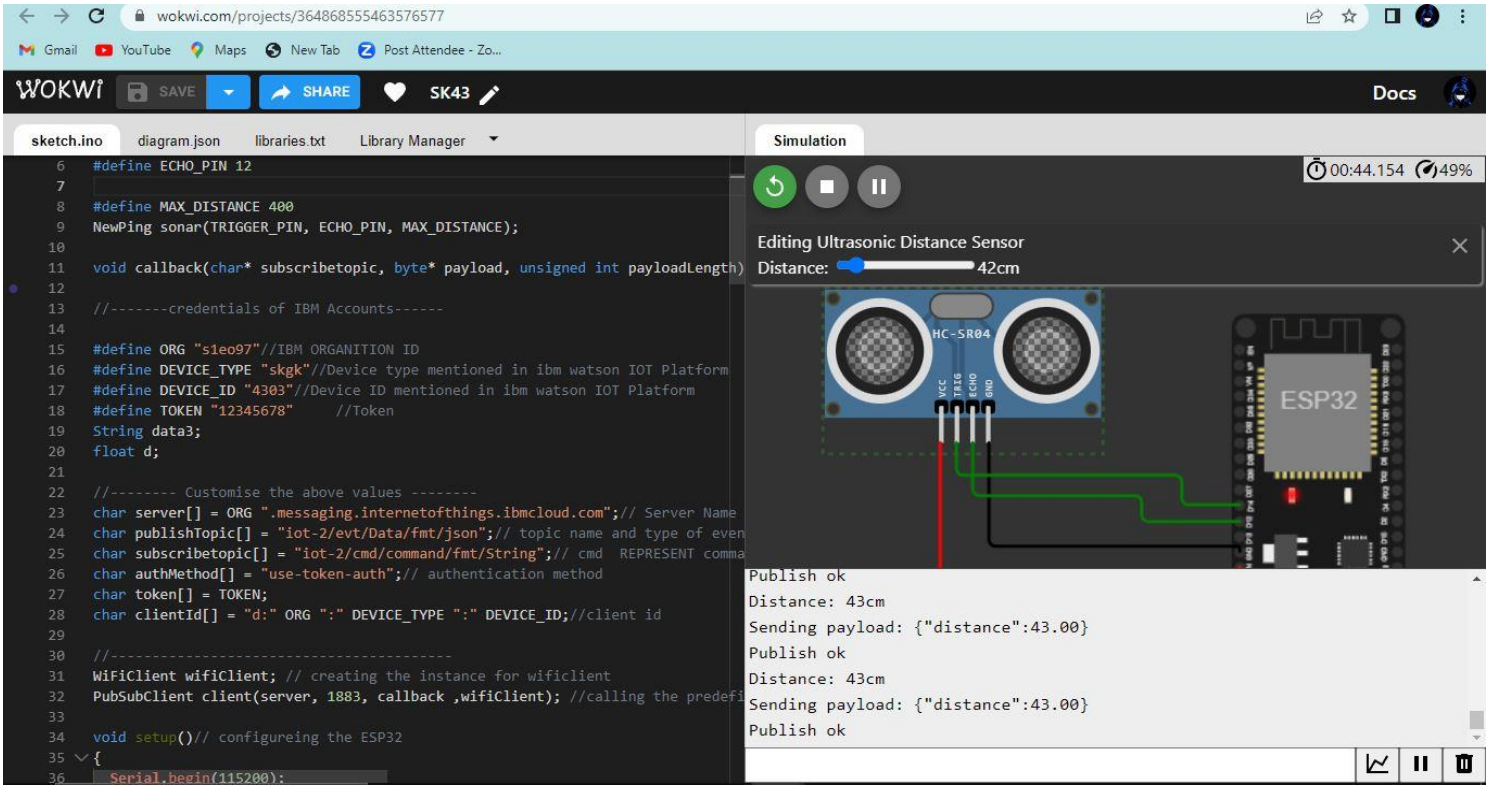
ASSIGNMENT NO : 3

ASSIGNMENT 3 : USE ULTRASONIC SENSOR AND DETECT THE DISTANCE FROM THE OBJECT

ASSIGNMENT LINK :

https://wokwi.com/projects/364868555463576577





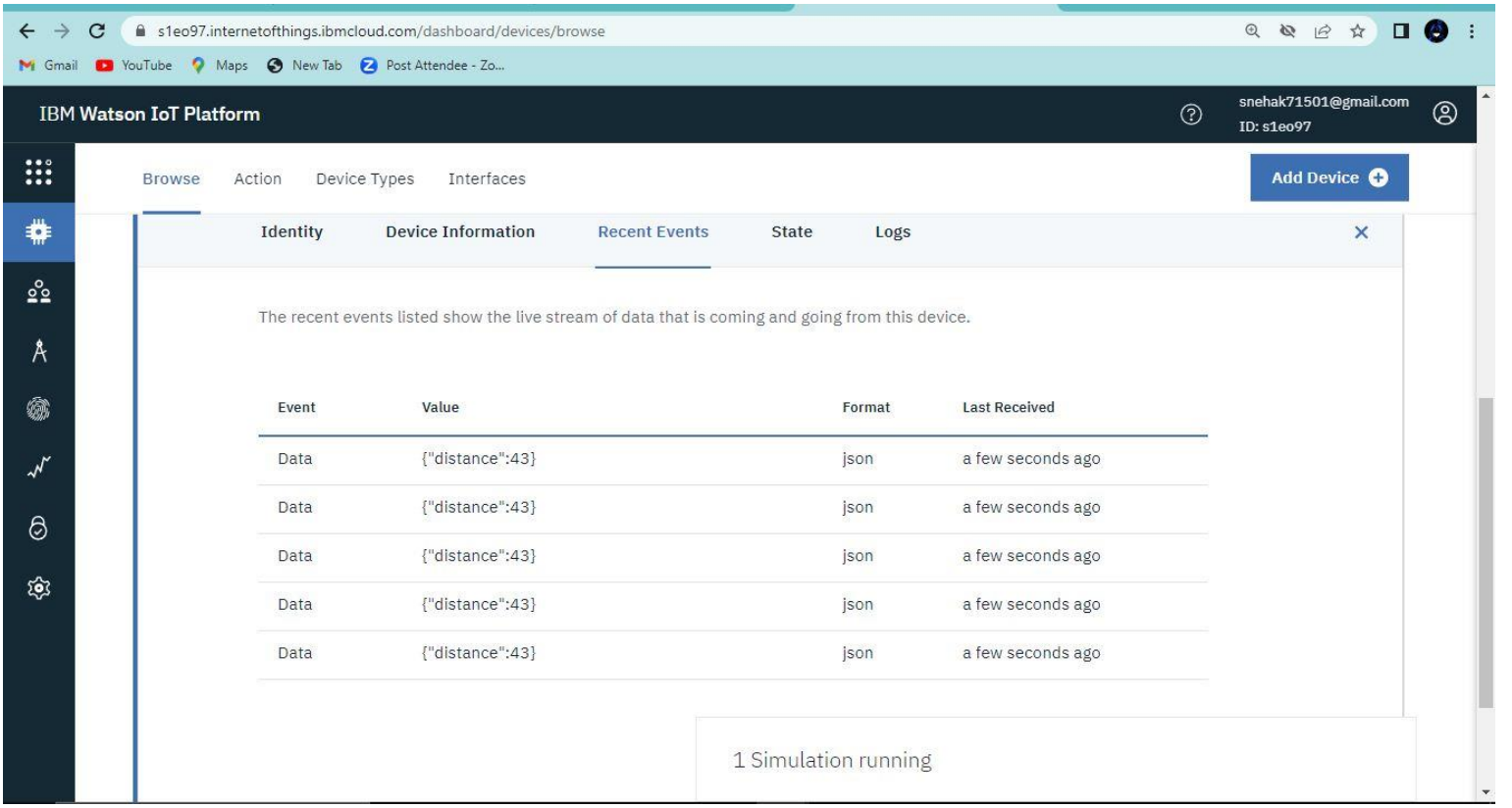
CODING :

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include <NewPing.h>

#define TRIGGER_PIN 14
#define ECHO_PIN 12

#define MAX_DISTANCE 400
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

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

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

#define ORG "s1eo97"//IBM ORGANITION ID
```

```
#define DEVICE_TYPE "skgk"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "4303"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float d;

//-------- 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 com-
mand 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 prede-
fined client id by passing parameter like server id,portand wificredential

void setup()// configureing the ESP32
{
  Serial.begin(115200);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{
    // Send a pulse to the ultrasonic sensor and wait for the echo
  unsigned int distance = sonar.ping_cm();

  // Print the distance to the serial monitor
  Serial.print("Distance: ");
  d = (distance);
  Serial.print(distance);
  Serial.println("cm");

  // Wait a short time before taking the next measurement
  delay(100);

  PublishData(d);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
```

```
  }
}

void PublishData(float d) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"distance\":";
  payload += d;
  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=="lighton")
  {
Serial.println(data3);
  }
  else
  {
Serial.println(data3);
  }
data3="";
}
```



LIBRARY FILE :

1.PubSubClient

2.ArduinoJson

3.NewPing