***Assignment 4: Ultra sonic sensor***

**Wokwi Link :**

https://wokwi.com/projects/346229697892319828

**Program:**

#include <WiFi.h>

#include <WiFiClient.h>

#include <PubSubClient.h>

const char\* ssid = "Wokwi-GUEST";

const char\* password = "";

#define ORG "4fvguz"

#define DEVICE\_TYPE "ESP32"

#define DEVICE\_ID "ultrasonic\_sensor"

#define TOKEN "12345678"

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char pubTopic[] = "iot-2/evt/status1/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883, NULL, wifiClient);

#define ECHO\_PIN 13

#define TRIG\_PIN 12

#define LED\_BUILTIN 5

#define DHT\_PIN 15

void setup()

{

**Serial**.begin(115200);

  pinMode(LED\_BUILTIN, OUTPUT);

  pinMode(TRIG\_PIN, OUTPUT);

  pinMode(ECHO\_PIN, INPUT);

**Serial**.print("Connecting to ");

**Serial**.print(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED)

  {

    delay(500);

**Serial**.print(".");

  }

**Serial**.println("");

**Serial**.print("WiFi connected, IP address: ");

**Serial**.println(WiFi.localIP());

  if (!client.connected())

  {

**Serial**.print("Reconnecting client to ");

**Serial**.println(server);

    while (!client.connect(clientId, authMethod, token))

    {

**Serial**.print(".");

      delay(500);

    }

**Serial**.println("Bluemix connected");

  }

}

float readDistanceCM() {

  digitalWrite(TRIG\_PIN, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIG\_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIG\_PIN, LOW);

  int duration = pulseIn(ECHO\_PIN, HIGH);

  return duration \* 0.034 / 2;

}

void loop() {

  float distance = readDistanceCM();

  bool isNearby = distance < 100;

  digitalWrite(LED\_BUILTIN, isNearby);

**Serial**.print("Measured distance: ");

**Serial**.println(readDistanceCM());

  if(distance<100)

  {

    String payload = "{\"d\":{\"Name\":\"" DEVICE\_ID "\"";

    payload += ",\"Distance\":";

    payload += distance;

    payload += "}}";

**Serial**.print("Sending value: ");

**Serial**.println(payload);

    if (client.publish(pubTopic, (char\*) payload.c\_str()))

    {

**Serial**.println("Publish Success");

    }

    else

    {

**Serial**.println("Publish Failed");

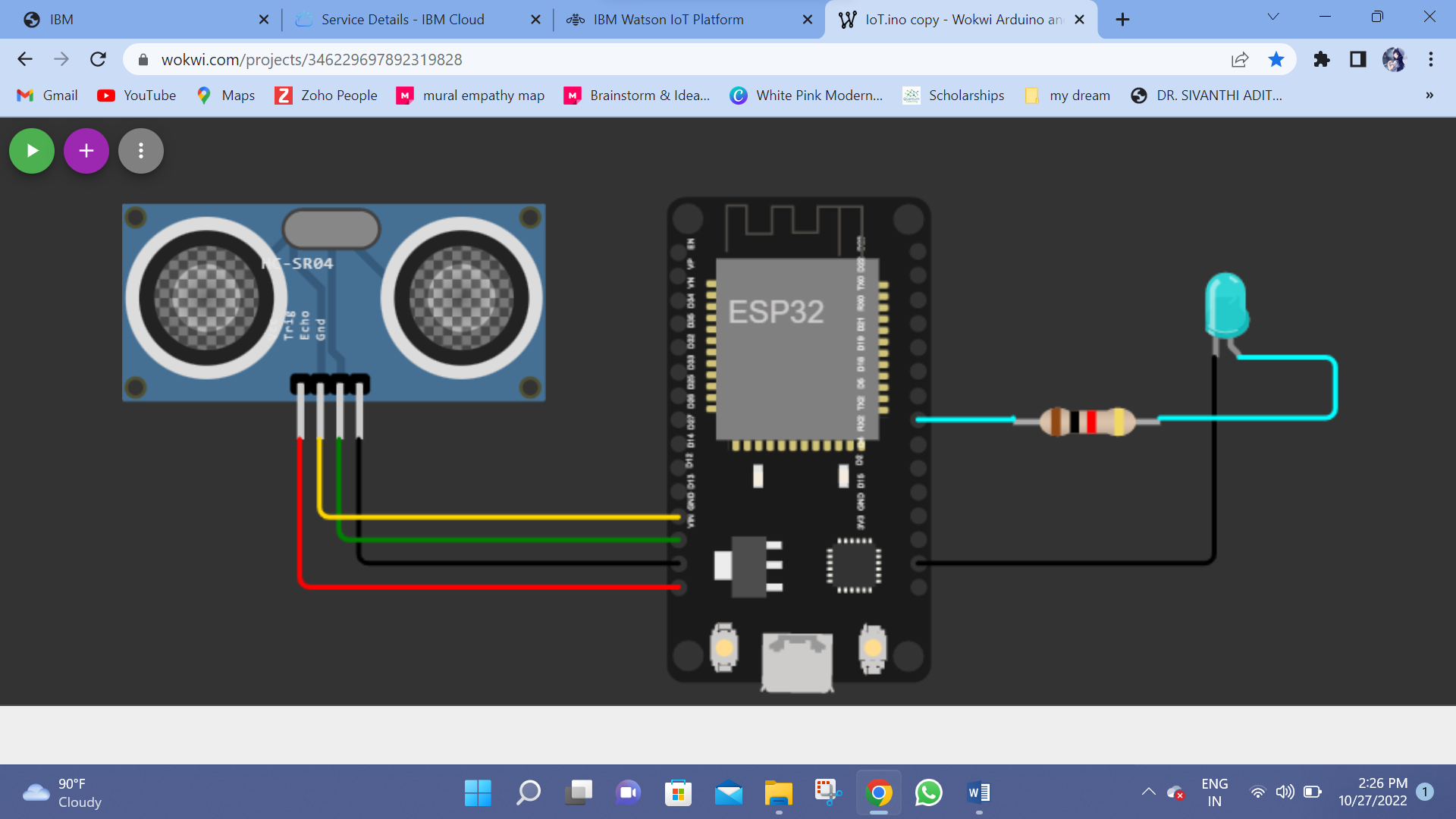
    }

  }

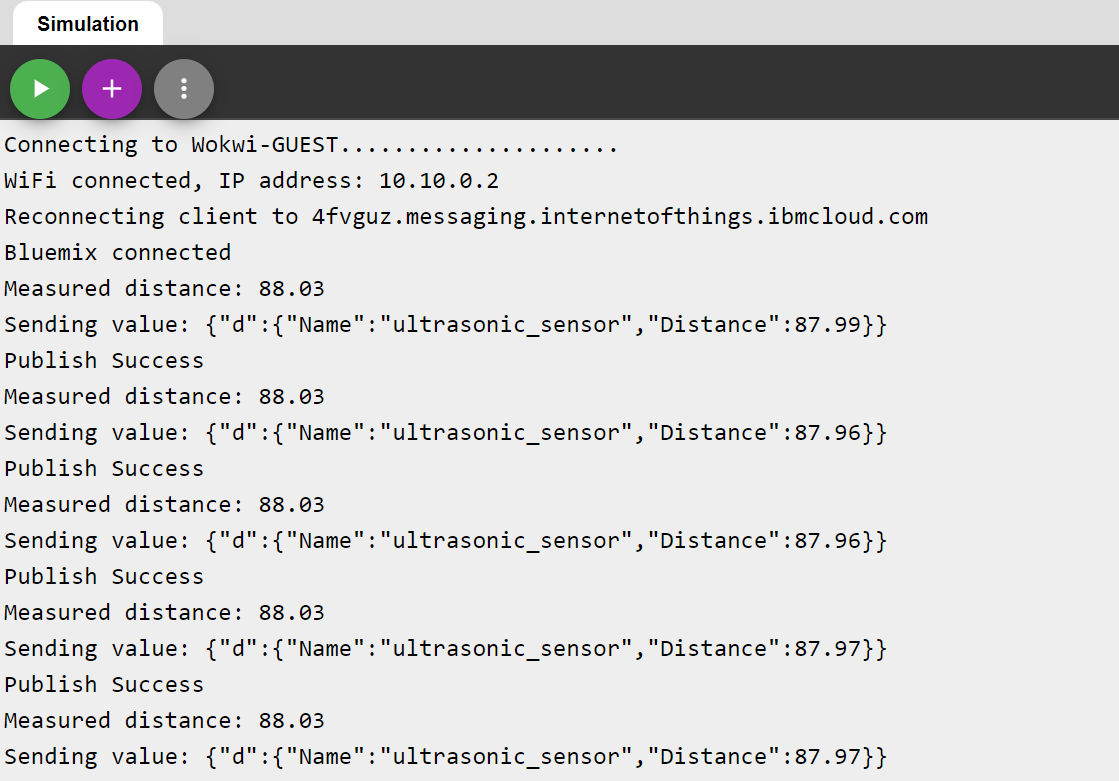
  delay(100);

}

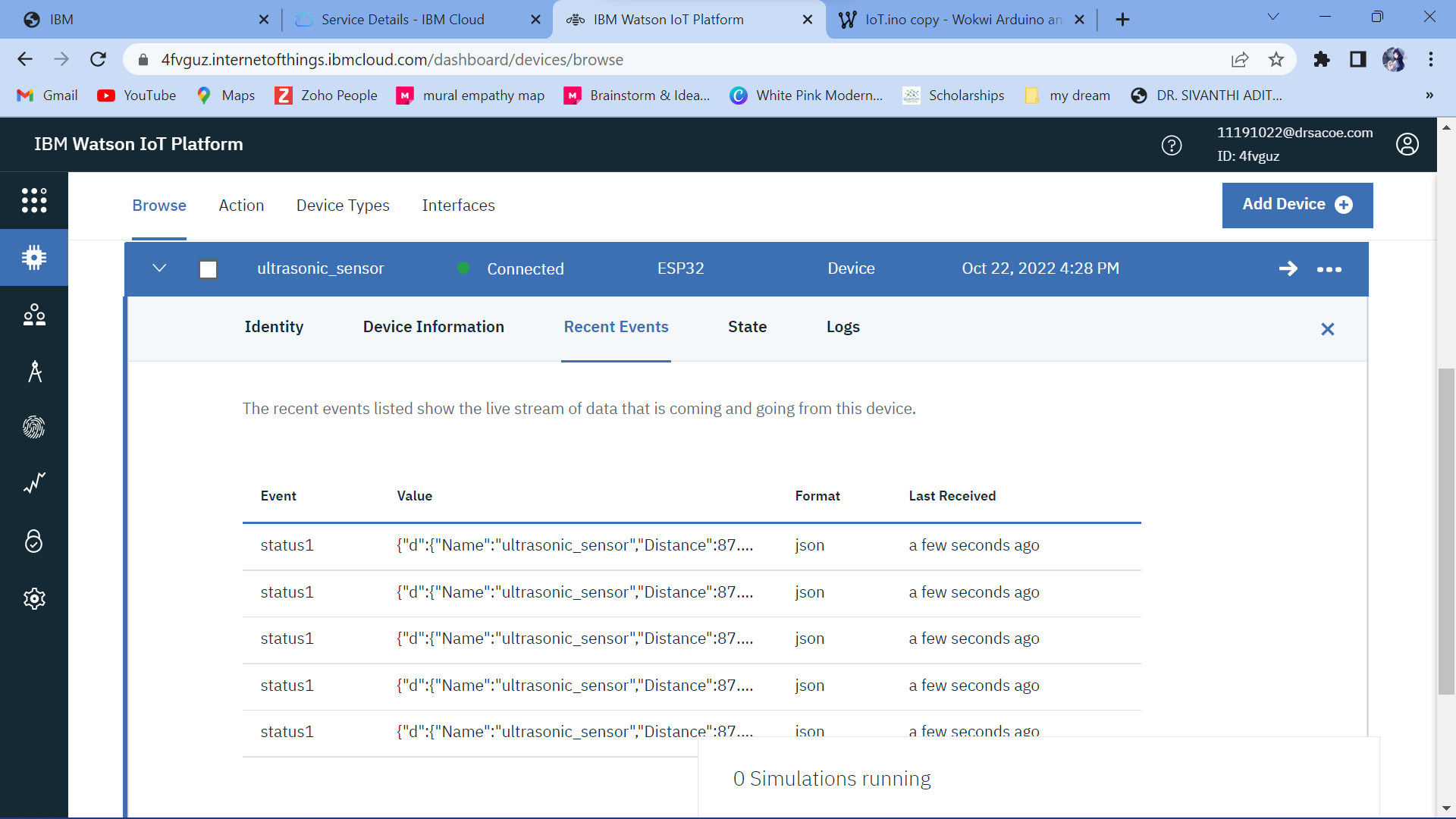
**Connections:**

****

**Output:**

****

**IBM Cloud:**

****