ASSIGNMENT-4

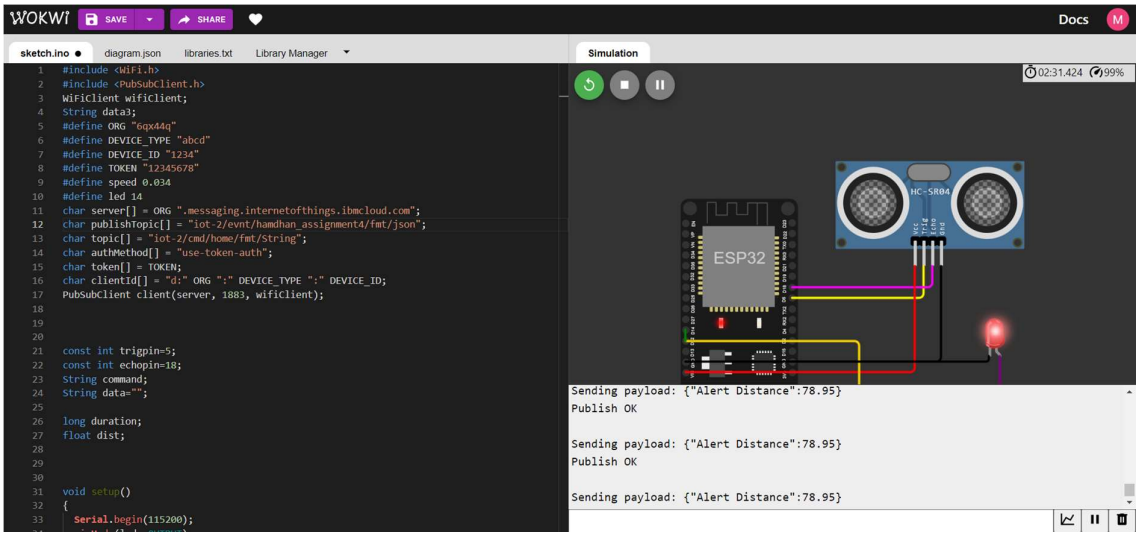
NAME:H.M.MOHAMAD HAMDHAN

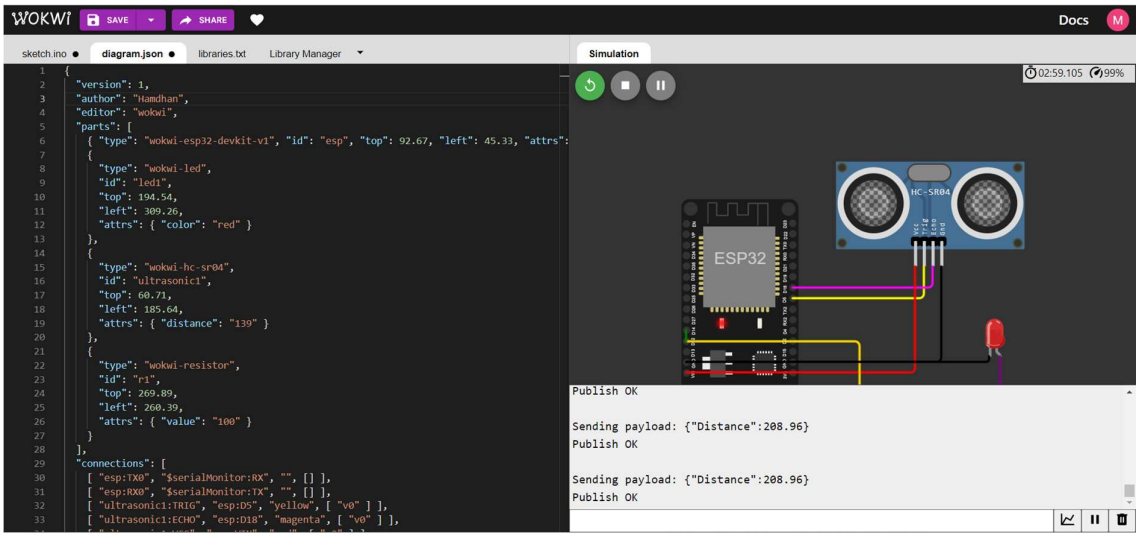
TEAM ID: PNT2022TMID21674

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "6qx44q"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "1234"
#define TOKEN "12345678"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evnt/hamdhan_assignment4/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);



const int trigpin=5;
const int echopin=18;
String command;
String data="";

long duration;
float dist;



void setup()
{
  Serial.begin(115200);
  pinMode(led, OUTPUT);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}

void loop() {
  bool isNearby = dist < 100;
  digitalWrite(led, isNearby);
```

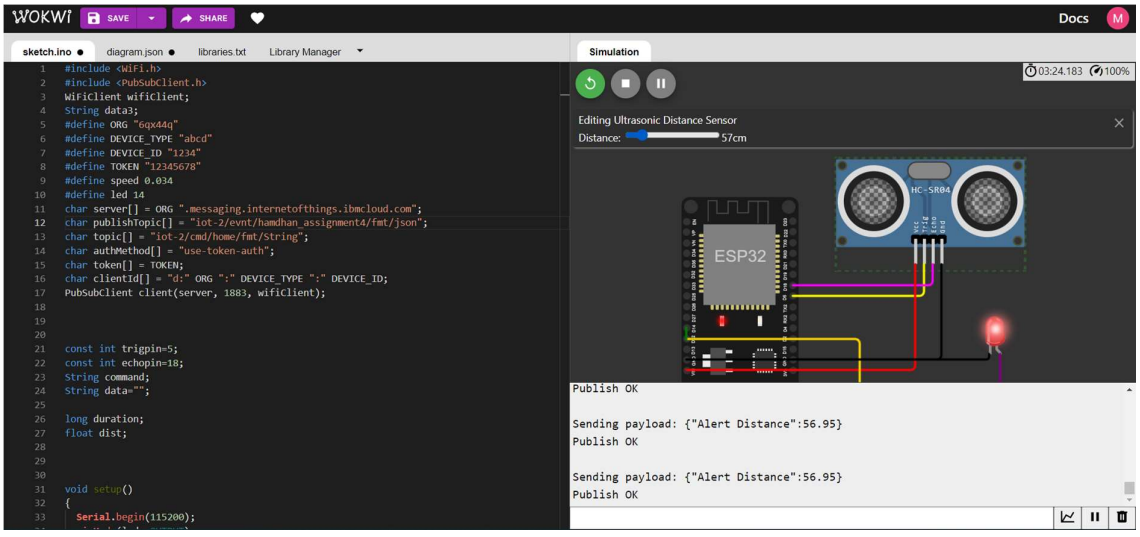
```
  publishData();
  delay(500);

  if (!client.loop()) {
    mqttConnect();
  }
}

void wifiConnect() {
  Serial.print("Connecting to "); Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
Serial.println(WiFi.localIP());
}

void mqttConnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting MQTT client to "); Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void initManagedDevice() {
  if (client.subscribe(topic)) {
    // Serial.println(client.subscribe(topic));
    Serial.println("IBM subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}
void publishData()
{
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  dist=duration*speed/2;
  if(dist<100){
```

```
  String payload = "{\"Alert Distance\":";
  payload += dist;
  payload += "}";

  Serial.print("\n");
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
  }

}
  if(dist>100){
  String payload = "{\"Distance\":";
  payload += dist;
  payload += "}";

  Serial.print("\n");
  Serial.print("Sending payload: ");
  Serial.println(payload);
   if(client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
  }else {
    Serial.println("Publish FAILED");
  }

}


}
```

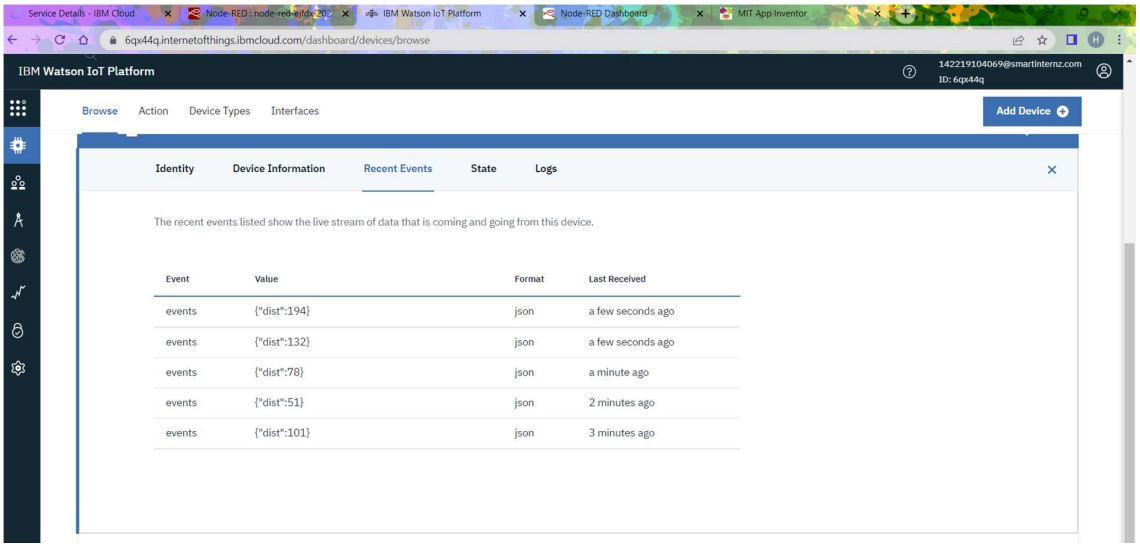


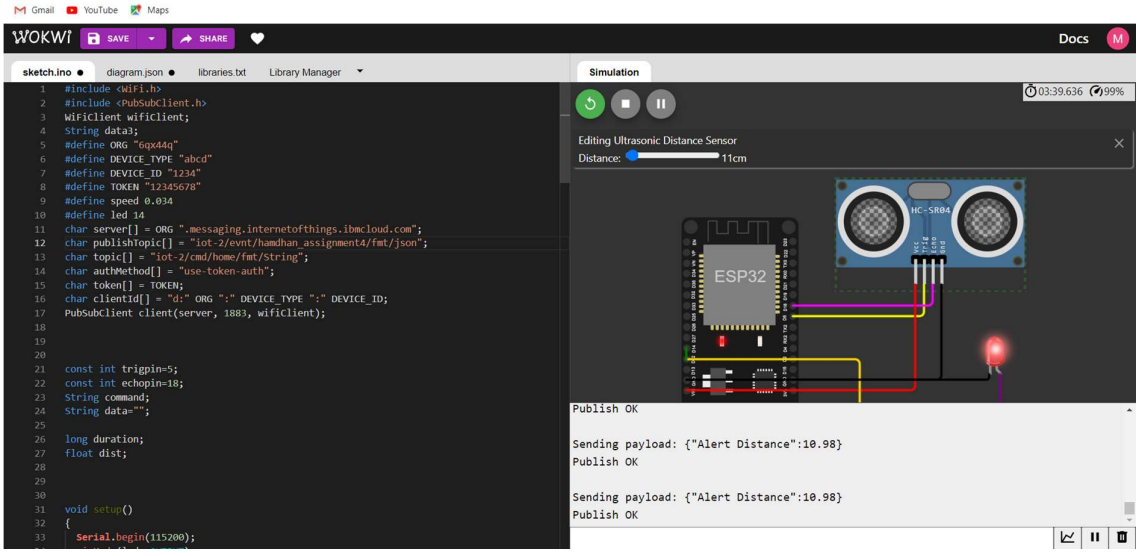



```
{
  "version": 1,
  "author": "Hamdhan",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 92.67, "left": 45.33, "attrs":
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": 194.54,
      "left": 309.26,
      "attrs": { "color": "red" }
    },
    {
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": 60.71,
      "left": 185.64,
      "attrs": { "distance": "139" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": 269.89,
      "left": 260.39,
      "attrs": { "value": "100" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "ultrasonic1:TRIG", "esp:D5", "yellow", [ "v0" ] ],
    [ "ultrasonic1:ECHO", "esp:D18", "magenta", [ "v0" ] ],
```




```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "6qx44q"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "1234"
#define TOKEN "12345678"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evnt/hamdhan_assignment4/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);



const int trigpin=5;
const int echopin=18;
String command;
String data="";

long duration;
float dist;



void setup()
{
  Serial.begin(115200);
```


Service Details - IBM Cloud
Node-RED : node-red-hfdx202
IBM Watson IoT Platform
Node-RED Dashboard
MIT App Inventor
6qx44q.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
142219104069@smartinternz.com
ID: 6qx44q
Browse
Action
Device Types
Interfaces
Add Device
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
events
{"dist":194}
json
a few seconds ago
events
{"dist":132}
json
a few seconds ago
events
{"dist":78}
json
a minute ago
events
{"dist":51}
json
2 minutes ago
events
{"dist":101}
json
3 minutes ago


Gmail
YouTube
Maps
WOKWi
SAVE
SHARE
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "6qx44q"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "1234"
#define TOKEN "12345678"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evnt/hamdhan_assignment4/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration;
float dist;
void setup()
{
Serial.begin(115200);
Simulation
03:39.636
99%
Editing Ultrasonic Distance Sensor
Distance:
11cm
HC-SR04
ESP32
Publish OK
Sending payload: {"Alert Distance":10.98}
Publish OK
Sending payload: {"Alert Distance":10.98}
Publish OK