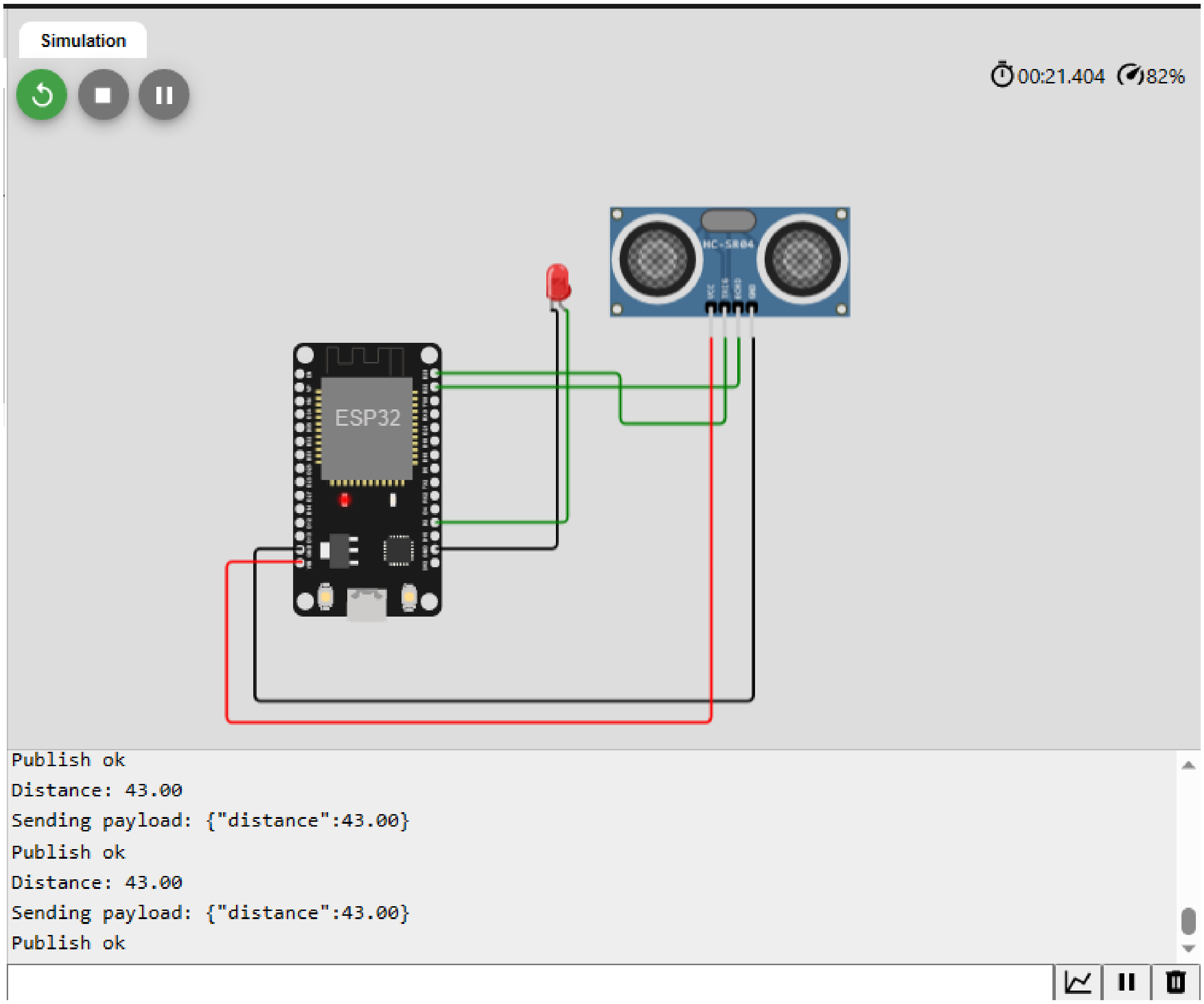
Assignment-3

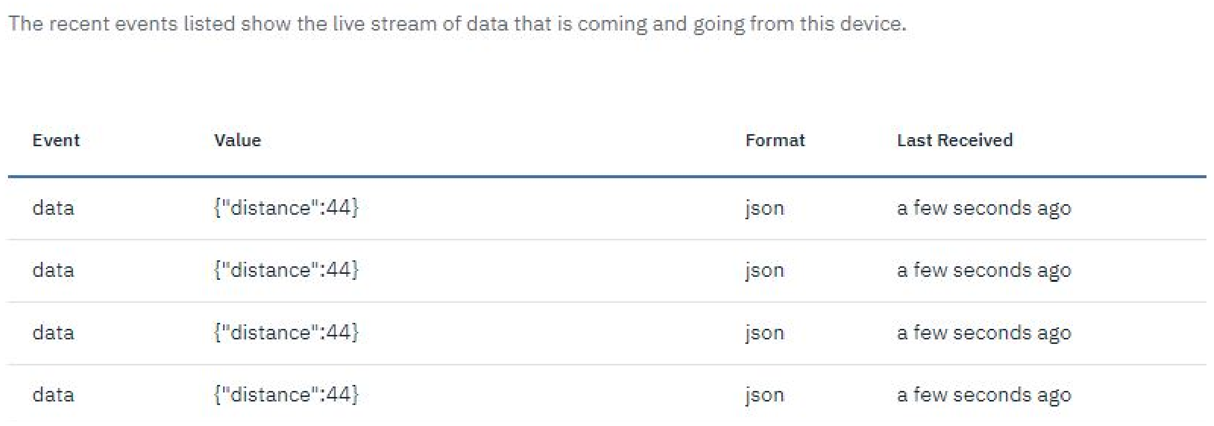
Name: M.Dhivyadharshini

Reg no: 711620104007

https://wokwi.com/projects/364627887422108673



```
Publish ok
Distance: 43.00
Sending payload: {"distance":43.00}
Publish ok
Distance: 43.00
Sending payload: {"distance":43.00}
Publish ok
```



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":44}	json	a few seconds ago
data	{"distance":44}	json	a few seconds ago
data	{"distance":44}	json	a few seconds ago
data	{"distance":44}	json	a few seconds ago

Sketch.Ino:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "Ultrasonic.h"
#define TRIG_PIN 23
#define ECHO_PIN 22
#define LED_PIN 2
Ultrasonic ultrasonic(TRIG_PIN, ECHO_PIN);
// IBM Watson IoT Platform credentials
#define ORG "ljd9ml"
#define DEVICE_TYPE "abcd"
#define DEVICE_ID "1234"
#define TOKEN "12345678"
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
void setup() {
Serial.begin(115200);
pinMode(LED_PIN, OUTPUT);
```

```
wificonnect();
mqttconnect();
}
void loop() {
float distance = ultrasonic.read();
Serial.print("Distance: ");
Serial.println(distance);
if (distance < 100) {
String payload = "{\"distance\":";
payload += distance;
payload += "}";
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish ok");
digitalWrite(LED_PIN, HIGH);
delay(500);
digitalWrite(LED_PIN, LOW);
} else {
Serial.println("Publish failed");
}
}
if (!client.loop()) {
mqttconnect();
}
delay(500);
}
void mqttconnect() {
if (!client.connected()) {
Serial.print("Connecting to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token)) {
Serial.print(".");
delay(500);
}
Serial.println("connected");
}
}
void wificonnect() {
Serial.print("Connecting to WiFi");
WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) {
Serial.print(".");
delay(500);
}
```

```
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}
```

Diagram.json:

```
{
  "version": 1,
  "author": "Anonymous maker",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 6, "left": -116,
"attrs": {} },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": -56.67,
      "left": 52.33,
      "attrs": { "color": "red" }
    },
    { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -89.17, "left":
109.37, "attrs": {} }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "esp:GND.1", "led1:C", "black", [ "h85.78", "v3.32" ] ],
    [ "esp:D2", "led1:A", "green", [ "h93.12", "v-0.75" ] ],
    [ "esp:D23", "ultrasonic1:TRIG", "green", [ "h130.64", "v36.04", "h75.21" ]
],
    [ "esp:D22", "ultrasonic1:ECHO", "green", [ "h214.91", "v8.83" ] ],
    [ "esp:GND.2", "ultrasonic1:GND", "black", [ "h-33.13", "v108.59", "h389.65"
] ],
    [ "esp:VIN", "ultrasonic1:VCC", "red", [ "h-53.07", "v114.49", "h339.82" ] ]
  ],
  "dependencies": {}
}
```