| 9th WSS General Meeting |

IoT WORKSHOP

By:

Mahdi Bahreiny

روند پیشرفت کارگاه

جلسه 1

راه اندازی ESP و
کنترل آن باسیم

جلسه 2

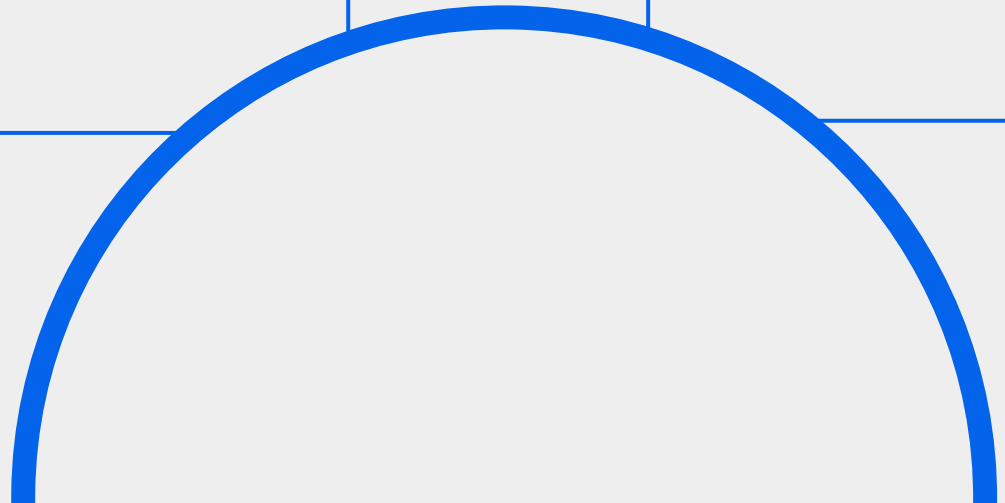
اتصال ESP به شبکه و
کنترل آن به صورت بی
سیم و لوکال

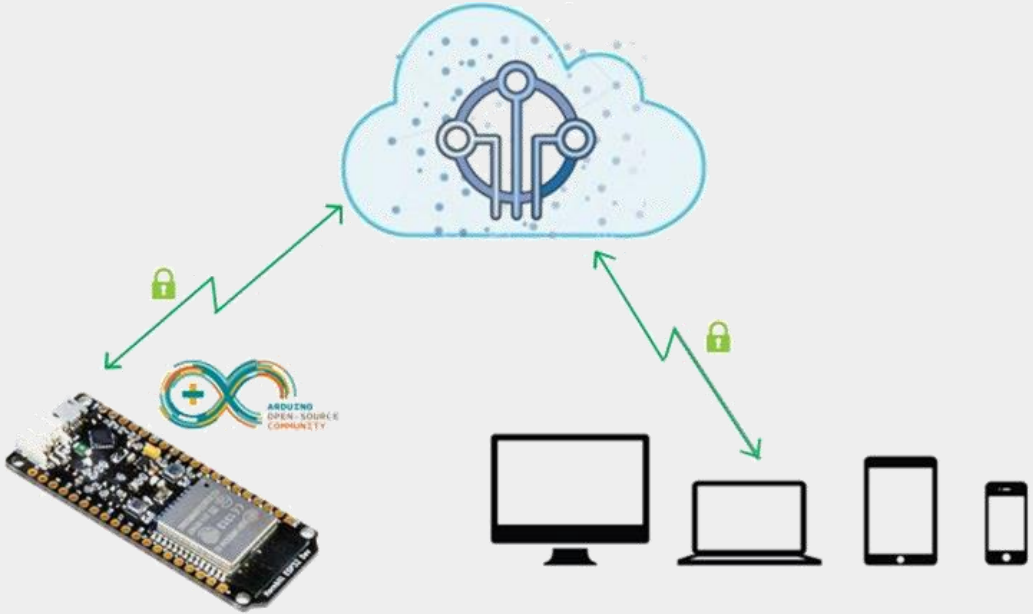
جلسه 3

اتصال ESP به اینترنت
و کنترل آن به صورت
بی سیم و از راه دور

جلسه 4

راه اندازی پنل کاربری
برای کنترل ESP از
طریق اینترنت





جلسه سوم:

اتصال ESP32 به اینترنت و پروتکل MQTT

پروتکل MQTT

یک پروتکل پرکاربرد پیام رسانی در اینترنت اشیاء

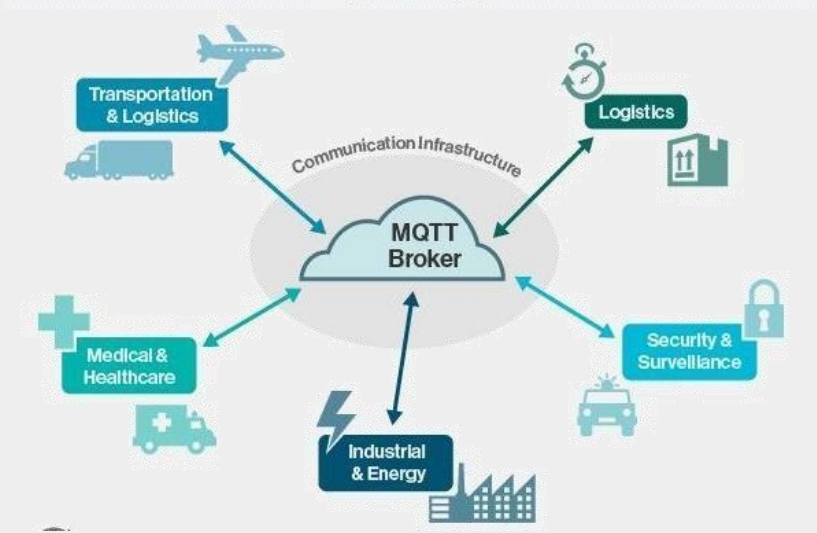
سبک و کار آمد مناسب برای پهنای باند کم

بر پایه انتشار/اشتراک (Publish/Subscribe)

با ثبات در ارسال پیام

مناسب برای شبکه های پرنوسان

امنیت بالا با استفاده از TLS



معماری MQTT

- **Broker**

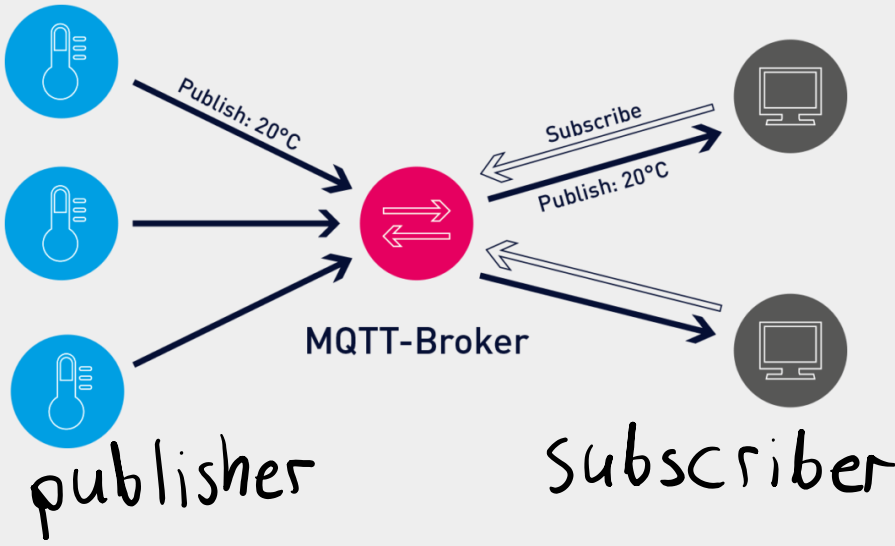
مسئول مدیریت شبکه clientها است که ترکیبی از Publisher و Subscriber است.

- **Publisher**

دستگاهی است که پیام ها را به سرور ارسال می کند (منتشر می کند). این پیام ها با یک نام "topic" مشخص می شوند.

- **Subscriber**

دستگاهی است که در یک "topic" مشترک می شود و پیام های آن را دریافت می کند.



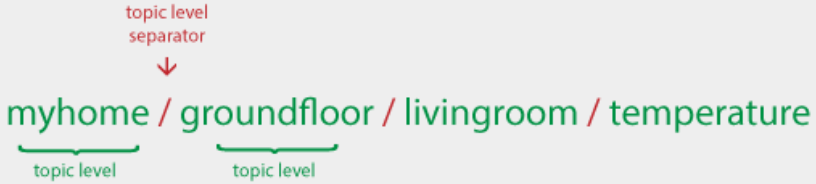
TOPICها و Messageها

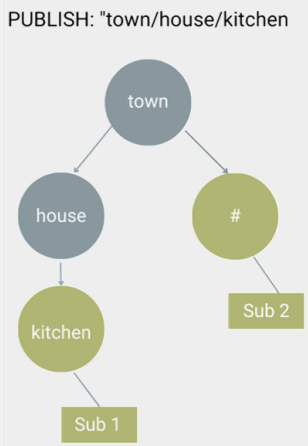
TOPIC

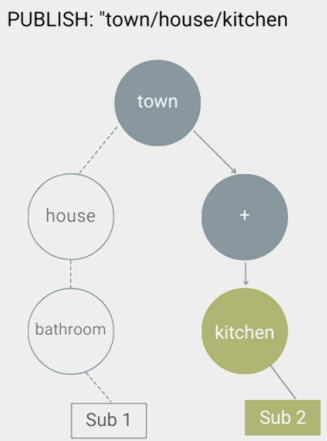
- موضوعی است که فرستنده تحت آن عنوان پیام را ارسال می‌کند.

گیرنده می‌تواند با گوش دادن بر روی آن تاپیک پیام را دریافت کند.

- با رشته هایی که با یک اسلش(/) جدا شده اند نشان داده می شوند. هر اسلش رو به جلو سطح موضوع را نشان می دهد.







Message

اطلاعاتی که بین دستگاه مبادله می شود. برای مثال فرمان یا داده‌هایی مانند قرائت سنسور

یک سناریو استفاده از MQTT

دستگاه ESP32

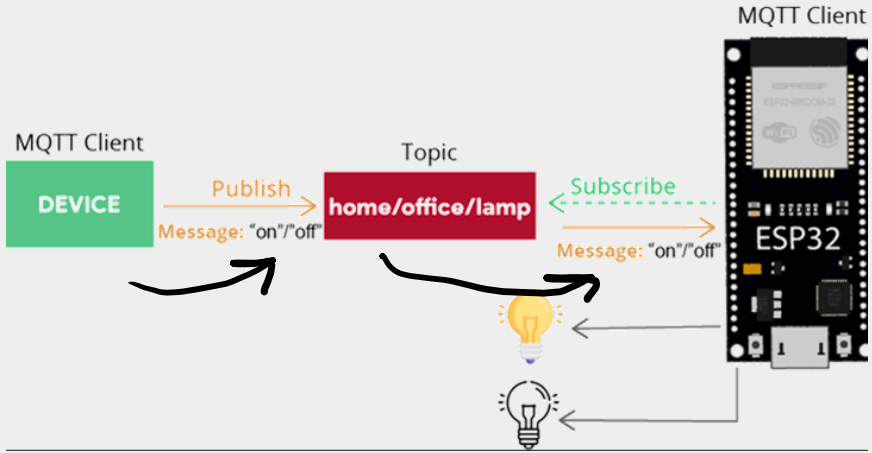
این دستگاه در تاپیک home/office/lamp مشترک است

دستگاه موبایل

پیام "on" و یا "off" را با تاپیک home/office/lamp ارسال می‌کند

نتیجه

هنگامی که یک پیام جدید در مورد آن موضوع منتشر می شود، ESP32پیام "on" و یا "off" را دریافت می کند و لامپ را روشن یا خاموش می کند



مفهوم کیفیت خدمات (QoS)

توافق بین کلاینت ها در مورد سطح خدمات

- **QoS0**

بدون ACK، ارسال کن و فراموشش کن، سریع ترین

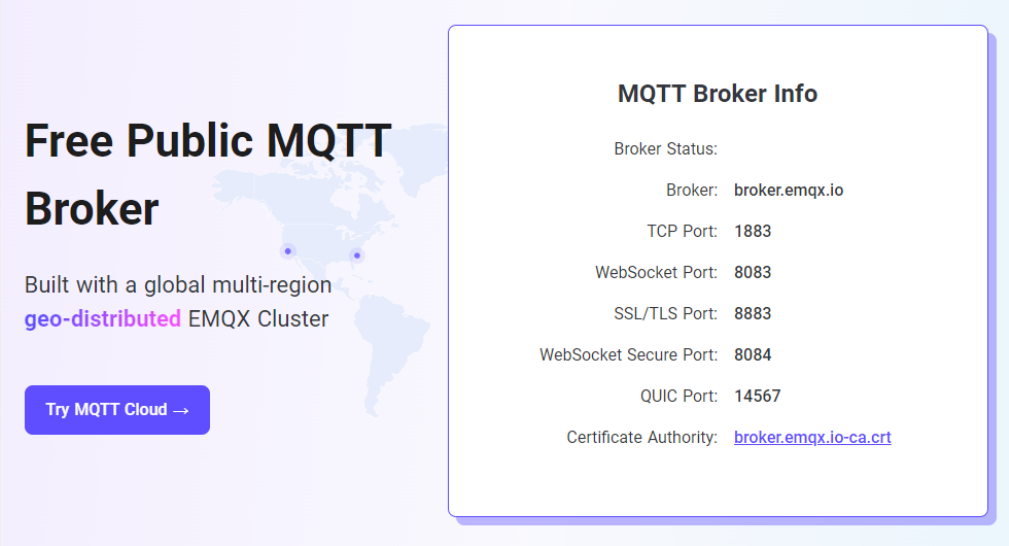
- **QoS1**

یک ACK، همیشه حداقل یک بار تحویل داده می شود، کیفیت معقول

- **QoS2**

دو ACK، دقیقا یک بار تحویل داده می شوند، قابل اطمینان ترین و کندترین

MQTT Broker



- Mosquitto
- Mosca
- emqttd
- **EMQX**
  - سهولت استفاده
  - سرعت مناسب
  - پشتیبانی از ورژن ها به روز MQTT
- Emitter

MQTT Client

- **MQTTX**

  نرم افزار جامع با امکانات مناسب

- **Paho-MQTT**

  امکان تلفیق با سایر برنامه های پایتون مانند جنگو

  سهولت استفاده

- **ESP32**

  ماژول وایفای با قیمت مناسب

  عمکلرد خوب در کاربردهای IoT

قالب پیام JSON

فرمتی برای به‌اشتراک‌گذاشتن داده‌ها

داده‌ها در جفت‌هایی به‌شکل نام/مقدار Key/Value قرار گرفته و با علامت کاما از یکدیگر جدا می‌شوند

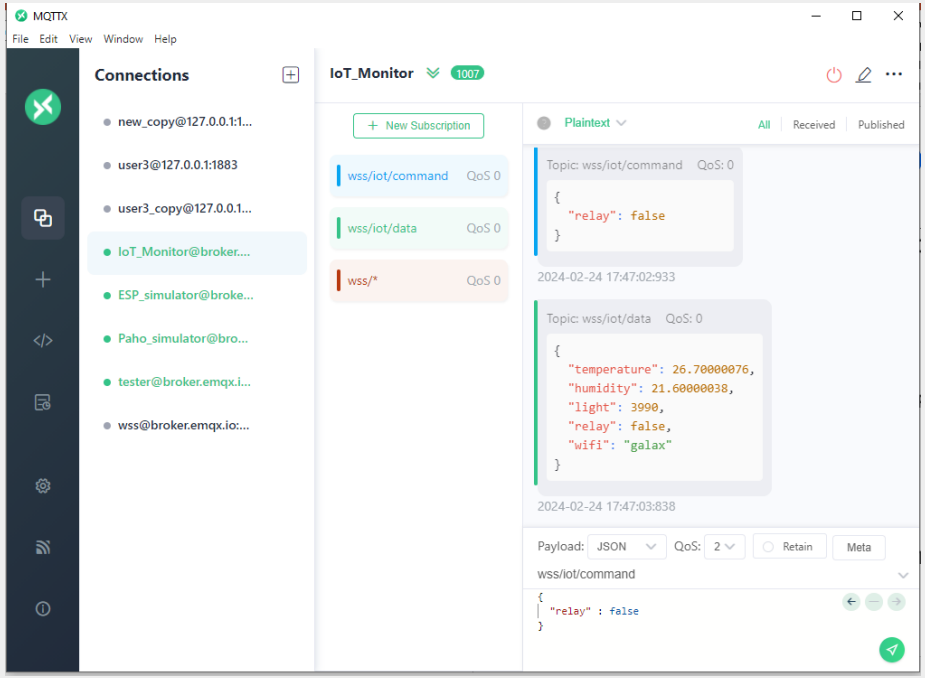
علامت‌های { } نگه‌دارنده آبجکت‌ها و [ ] نگه‌دارنده آرایه‌ها هستند

جفت‌های Key-Value یک دو نقطه (:) میان خود دارند

```
{
  "first_name" : "Sammy",
  "last_name" : "Shark",
  "location" : "Ocean",
  "online" : true,
  "followers" : 987
}
```


```
{
  "DocumentType": 1,
  "No.": "S-ORD101001",
  "SellToCustNo": "10000",
  "PostingDate": "2023-04-02",
  "Lines": [
    {
      "LineNo": 10000,
      "Type": 2,
      "No": "1996-S",
      "Quantity": 12,
      "UnitPrice": 1397.3
    },
    {
      "LineNo": 20000,
      "Type": 2,
      "No": "1900-S",
      "Quantity": 4,
      "UnitPrice": 192.8
    }
  ]
}
```

MQTTX



مناسب برای تست سریع سامانه های مبتنی بر MQTT

دارای امکانات کامل جهت مشترک شدن در یک تاپیک

و پابلیش کردن پیام

پشتیبانی از قالب هایی مانند json

Paho-MQTT

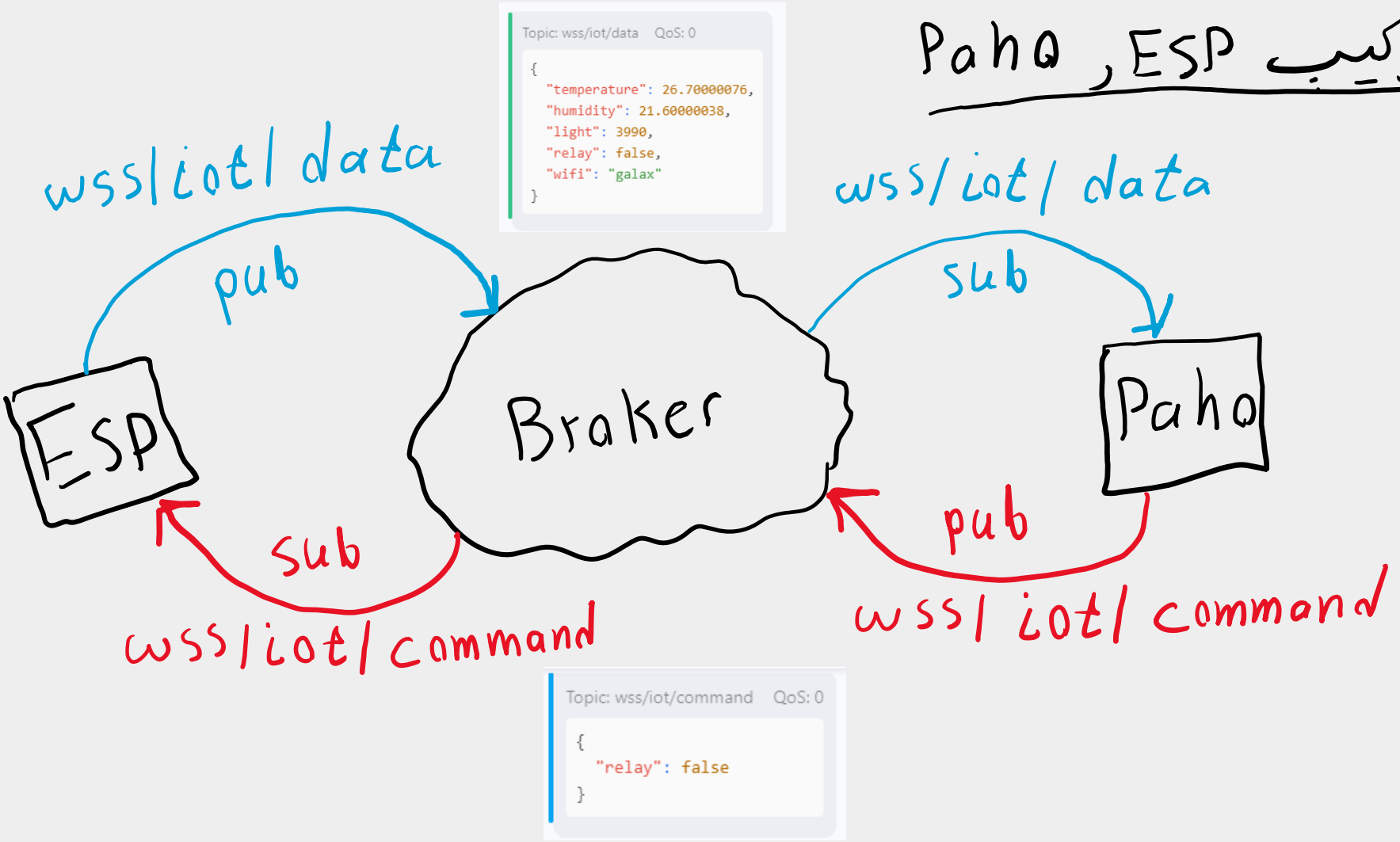
```python
def on_message(client, userdata, msg):
    try:
        data = json.loads(msg.payload)
        result = str(data)
    except:
        result = str(msg.payload)
    print("\n\nNew Message!\n_____________")
    print("topic:: " + msg.topic)
    print("message::\n" + result)

mqttClient = mqtt.Client(mqtt.CallbackAPIVersion.VERSION2)
mqttClient.on_connect = on_connect
mqttClient.on_message = on_message
# mqttClient.username_pw_set("username", "password")
mqttClient.connect("broker.emqx.io", 1883, 60)
mqttClient.loop_start()
try:
    while True:
        topic = input("enter topic: ")
        message = input("enter message: ")
        mqttClient.publish(topic, message)
        print("\n\n")
except:
    print("exiting")
    mqttClient.disconnect()
    mqttClient.loop_stop()
```

main.py

نوشتن یک برنامه برای ارسال پیام در یک تاپیک دلخواه از طریق کنسول

نمایش همزمان پیام دریافت شده در تاپیک های عضو شده در کنسول

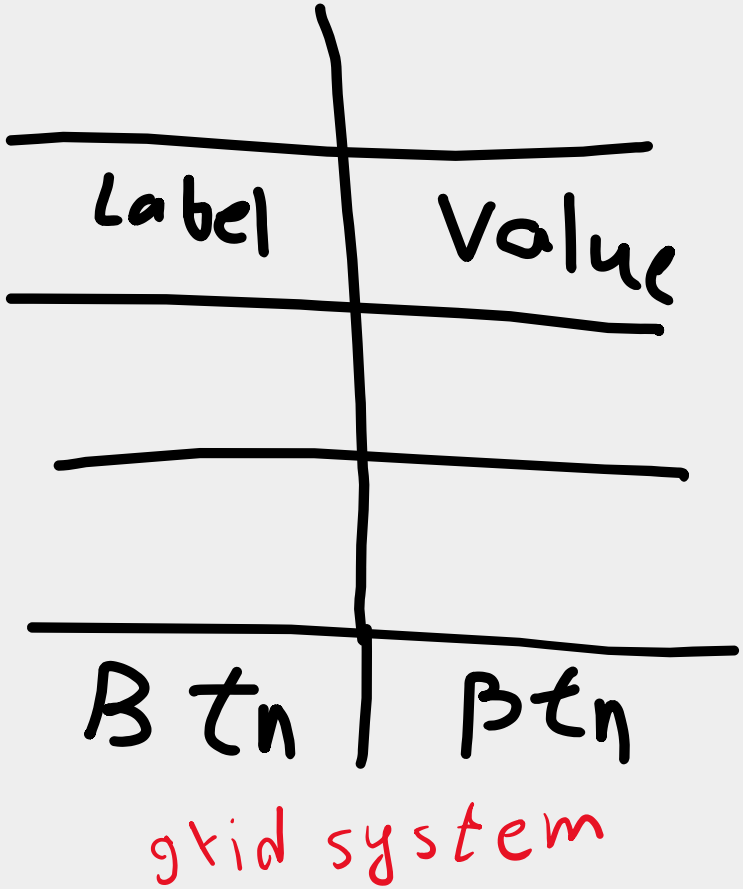
ترکیب Paho ,ESP



طراحی رابط کاربری با tkinter

```python
root =Tk()
p=Label(root, text="WSS ESP32 Through MQTT", font=("Arial", 15))
p.grid(row=0,column=0)
tmpLabel=Label(root, text="Temperature", fg="white", bg="red", height=4, font=("Arial", 15) , width=20)
tmpVal=Label(root, text="0°C", fg="white", bg="red", height=4, font=("Arial", 15) , width=20)
tmpLabel.grid(row=1,column=0)
tmpVal.grid(row=1,column=1)
humLabel=Label(root, text="Humidity", fg="white", bg="blue", height=4, font=("Arial", 15) , width=20)
humVal=Label(root, text="0%", fg="white", bg="blue", height=4, font=("Arial", 15) , width=20)
humLabel.grid(row=2,column=0)
humVal.grid(row=2,column=1)
lightLabel=Label(root, text="Light", fg="white", bg="orange", height=4, font=("Arial", 15) , width=20)
lightVal=Label(root, text="0", fg="white", bg="orange", height=4, font=("Arial", 15) , width=20)
lightLabel.grid(row=3,column=0)
lightVal.grid(row=3,column=1)
wifiLabel=Label(root, text="WiFi Network", fg="white", bg="green", height=4, font=("Arial", 15) , width=20)
wifiVal=Label(root, text="Name", fg="white", bg="green", height=4, font=("Arial", 15) , width=20)
wifiLabel.grid(row=4,column=0)
wifiVal.grid(row=4,column=1)
relayLabel=Label(root, text="Relay", fg="white", bg="magenta", height=4, font=("Arial", 15) , width=20)
relayVal=Label(root, text="n/a", fg="white", bg="magenta", height=4, font=("Arial", 15) , width=20)
relayLabel.grid(row=5,column=0)
relayVal.grid(row=5,column=1)
relayOn=Button(root, text="Turn On",relief=RAISED, height=4, font=("Arial", 15) , width=20, command=turnOnRelay)
relayOff=Button(root, text="Turn Off",relief=RAISED, height=4, font=("Arial", 15) , width=20, command=turnOffRelay)
relayOn.grid(row=6,column=0)
relayOff.grid(row=6,column=1)
t=Label(root, text="Last updated time: null", font=("Arial", 15))
t.grid(row=7,column=0,columnspan=2)
root.mainloop()
```

mainUI.py



ترکیب Paho با tkinter

خواندن پیام در قالب json، استخراج پارامترهای آن و به روزرسانی رابط کاربری

```python
def on_message(client, userdata, msg):
    try:
        data = json.loads(msg.payload)
        tmpVal['text'] = str(round(data['temperature'],2)) + "°C"
        humVal['text'] = str(round(data['humidity'],2)) + "%"
        lightVal['text'] = str(data['light']) + " out of 4095"
        relayVal['text'] = "on" if data['relay'] else "off"
        wifiVal['text'] = data['wifi']
        named_tuple = time.localtime() # get struct_time
        time_string = time.strftime("%m/%d/%Y, %H:%M:%S", named_tuple)
        t['text'] = "Last updated time: " + time_string
        result = str(data)
    except:
        result = str(msg.payload)
    print("\n\nNew Message!\n_____________")
    print("topic:: " + msg.topic)
    print("message::\n" + result)
```

ارسال پیام در قالب json برای بروکر در صورت زدن کلید در رابط کاربری

```python
def turnOnRelay():
    x = {"relay": True}
    y = json.dumps(x)
    mqttClient.publish("wss/iot/command", y)
def turnOffRelay():
    x = {"relay": False}
    y = json.dumps(x)
    mqttClient.publish("wss/iot/command", y)
```

mainUI.py

MQTT با سریال در ESP32

برنامه برای دریافت دستور در سریال جهت مشترک شدن در یک تاپیک یا ارسال پیام در یک تاپیک

استفاده از کتابخانه PubSubClient

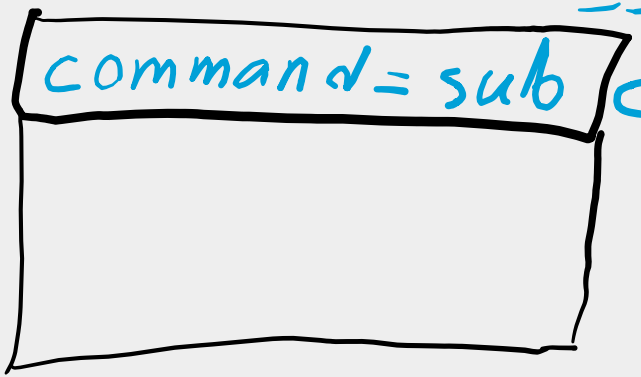
```
void processCommand(String command) {
  String params[4][2];
  uint8_t i = 0;
  int eqIdx, andIdx;
  do {
    eqIdx = command.indexOf("=");
    andIdx = command.indexOf("&");
    String key = command.substring(0, eqIdx);
    String value = command.substring(eqIdx + 1, andIdx);
    //Serial.println("key:" + key + "\tvalue: " + value);
    params[i][0] = key;
    params[i][1] = value;
    i++;
    command = command.substring(andIdx + 1);
  } while (andIdx != -1);
  //subscribing for a topic using this command in serial:
  //command=sub&topic=anything
  if (params[0][1] == "sub") {
    String topic = params[1][1];
    Serial.println("Subsrcibing in topic:" + topic);
    mqttClient.subscribe(topic.c_str());
  }
  //Sending a message in a topic using this command in serial:
  //command=pub&topic=anything&message=anythingElse
  else if (params[0][1] == "pub") {
    String topic = params[1][1];
    String message = params[2][1];
    Serial.println("Publishing message in topic:" + topic + "
    with payload: " + message);
    mqttClient.publish(topic.c_str(), message.c_str());
  }
```

نمایش پیام های دریافتی در سریال

```
void mqttCallback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived in topic: ");
  Serial.println(topic);
  Serial.print("Message:");
  String mqttMessage;
  for (int i = 0; i < length; i++)
    mqttMessage += (char)payload[i];
  Serial.println(mqttMessage);
}
```

Serial MQTT

مشترک شدن در یک تاپیک



ارسال پیام در تاپیک

command=pub & topic=topicName & message=text

برنامه نهایی ESP32

- کلید فشاری برای کنترل رله به صورت سخت افزاری
- وب سرور برای کنترل لوکال از طریق پروتکل HTTP
- کلاینت MQTT برای ارسال داده ها و کنترل از طریق اینترنت

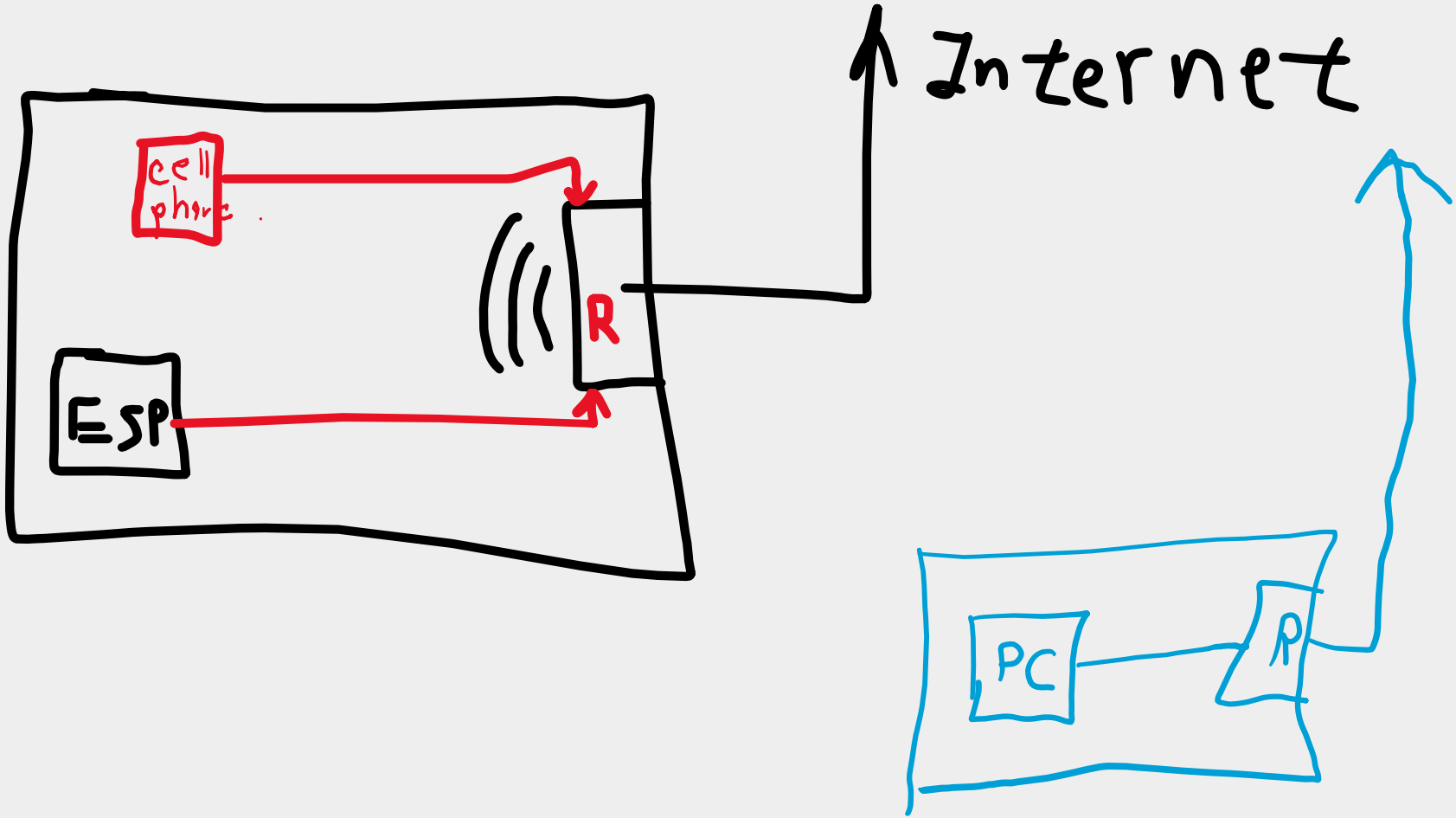
(در بازه زمانی 10 ثانیه ای یا درصورت رخداد تغییر در سیستم)

- استفاده از فرمت JSON برای ارسال و دریافت اطلاعات

```
//This function executes every time interval and whenever new event occurs on device to sync data with other clients
void publishDataToBroker() {
  if(!mqttClient.connected())
    mqttClientSetup();
  JsonDocument doc;
  String jsonMessage;
  temperature = dht22.getTemperature();
  humidity = dht22.getHumidity();
  light = analogRead(LDR_PIN);
  doc["temperature"] = temperature;
  doc["humidity"] = humidity;
  doc["light"] = light;
  doc["relay"] = relay;
  doc["wifi"] = ssid;
  //json format is used for data
  serializeJson(doc, jsonMessage);
  mqttClient.publish(PUBLISH_TOPIC, jsonMessage.c_str());
  lastTimeDataSent = millis();
}
```

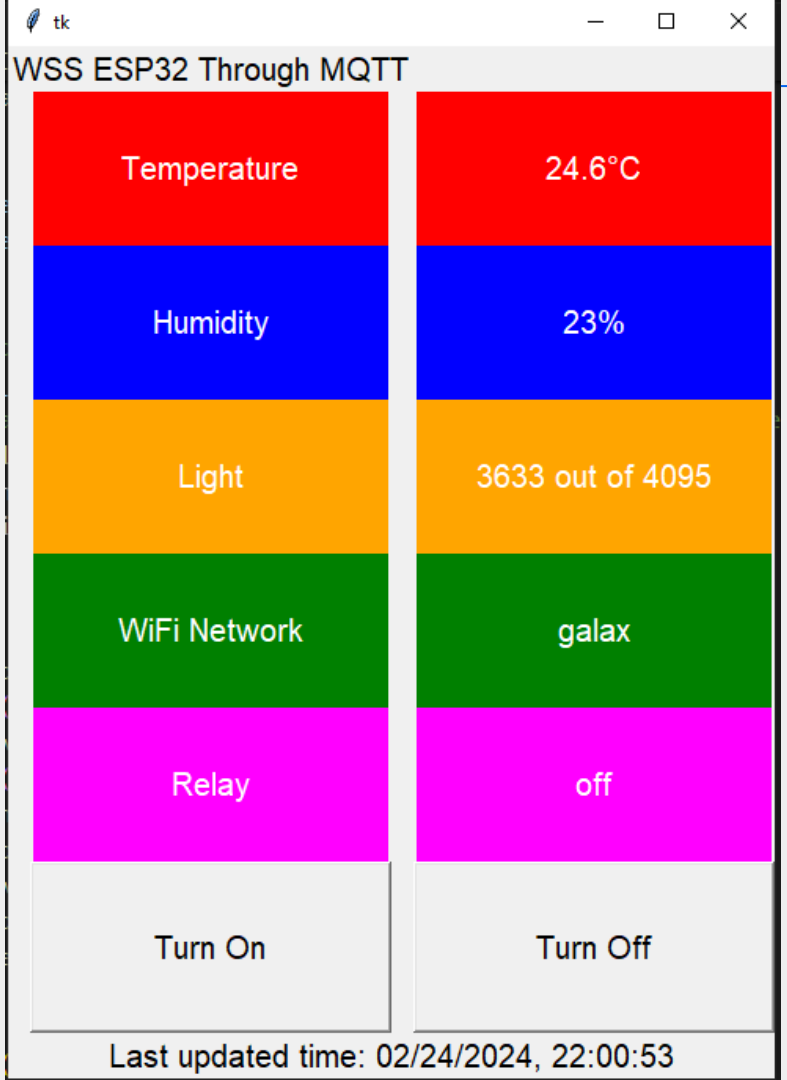
```
arrives
void mqttCallback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived in topic: ");
  Serial.println(topic);
  Serial.print("Message:");
  String jsonMessage;
  for (int i = 0; i < length; i++) {
    jsonMessage += (char)payload[i];
  }
  Serial.println(jsonMessage);
  JsonDocument doc;
  deserializeJson(doc, jsonMessage);
  //{"relay":true}
  if (doc.containsKey("relay")) {
    relay = doc["relay"];
    digitalWrite(RELAY_PIN, !relay);
    publishDataToBroker();
  }
}
```

```
void loop(void) {
  //toggle relay when button is pressed
  if (buttonPressedFlag) {
    if (digitalRead(BUTTON_PIN)) {
      relay = !relay;
      digitalWrite(RELAY_PIN, !relay);
      buttonPressedFlag = false;
      Serial.println("key was pressed");
      publishDataToBroker();
    }
  }
  //Send data to broker every time interval
  if(millis() - lastTimeDataSent > DATA_TIME_INTERVAL)
    publishDataToBroker();
```


Internet
cell phone
R
ESP
PC
R

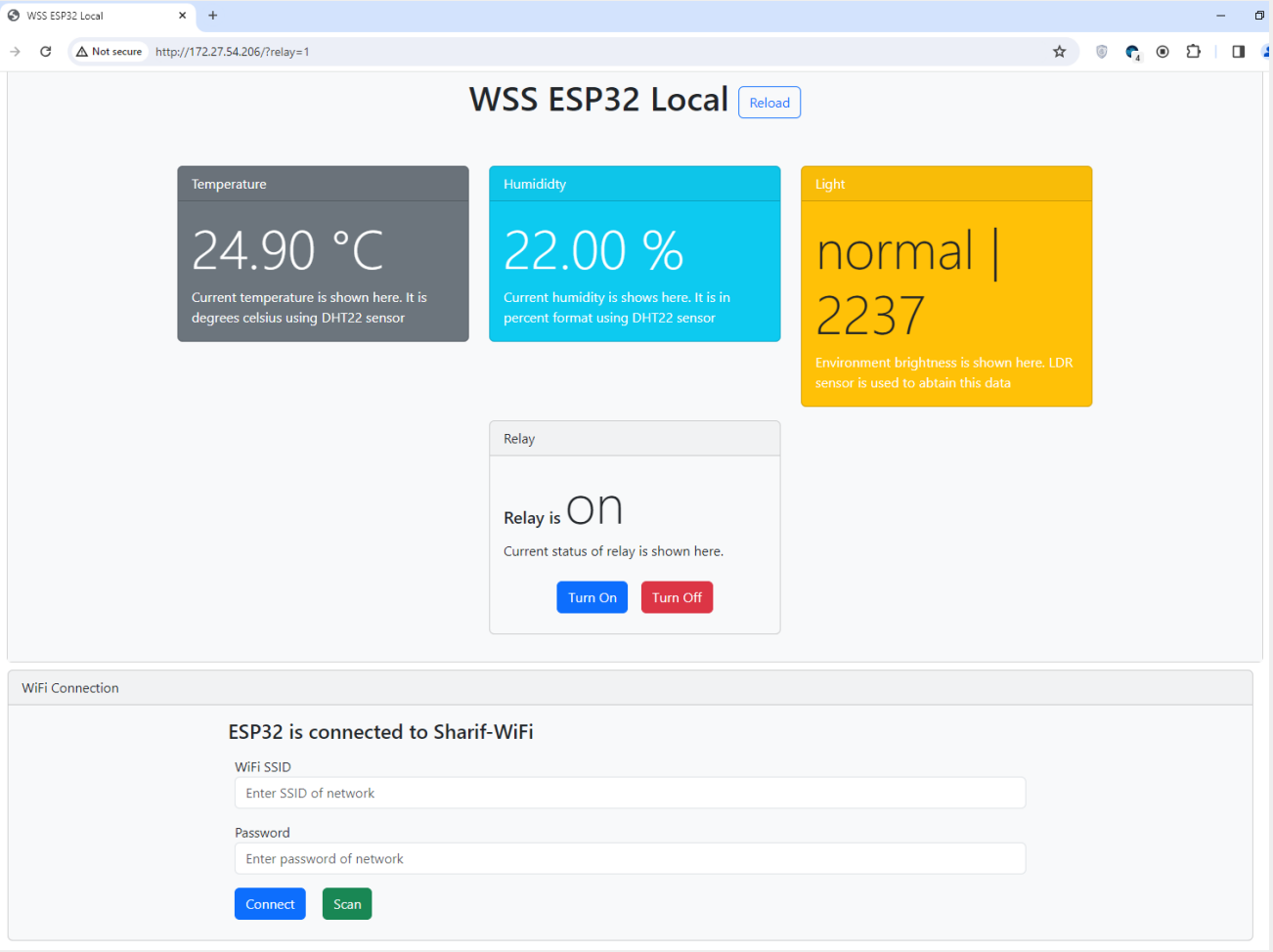
برنامه نهایی پایتون

- داده های نشان داده شده به کمک Paho-MQTT در پایتون از طریق اینترنت و پروتکل MQTT

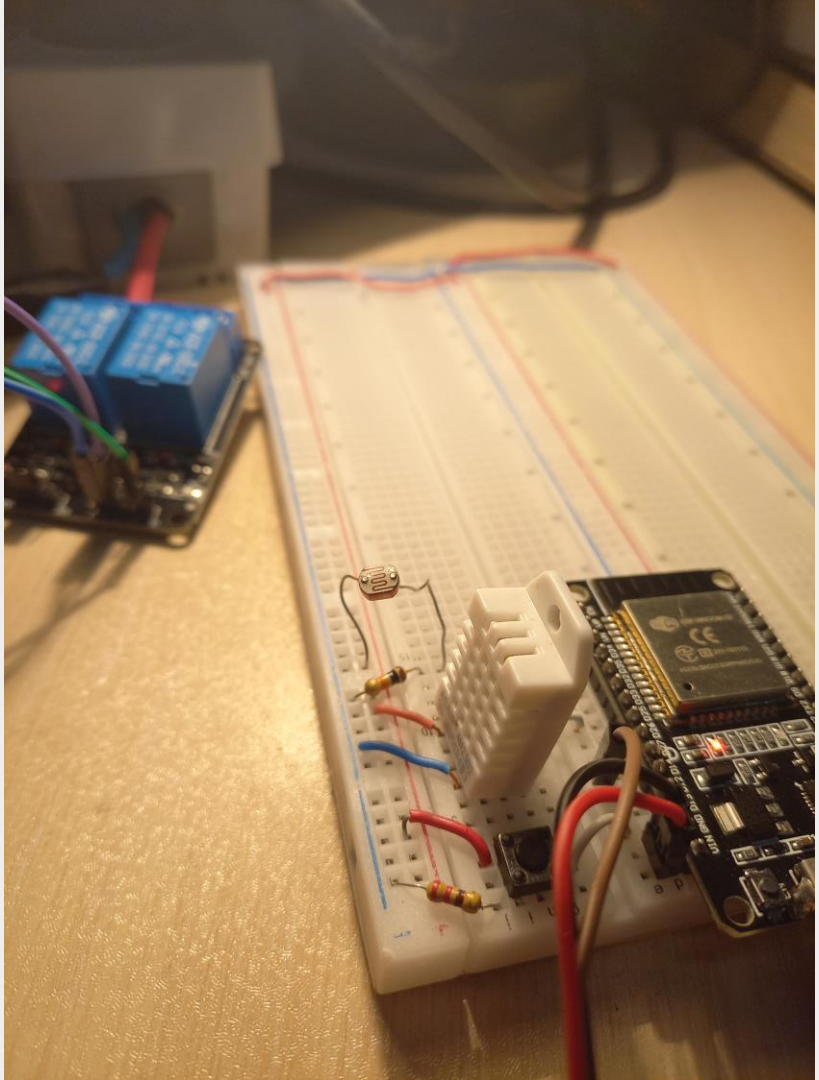


وب سرور لوکال نهایی

- داده های نشان داده شده توسط ESP32 در مرورگر به صورت لوکال و پروتکل HTTP



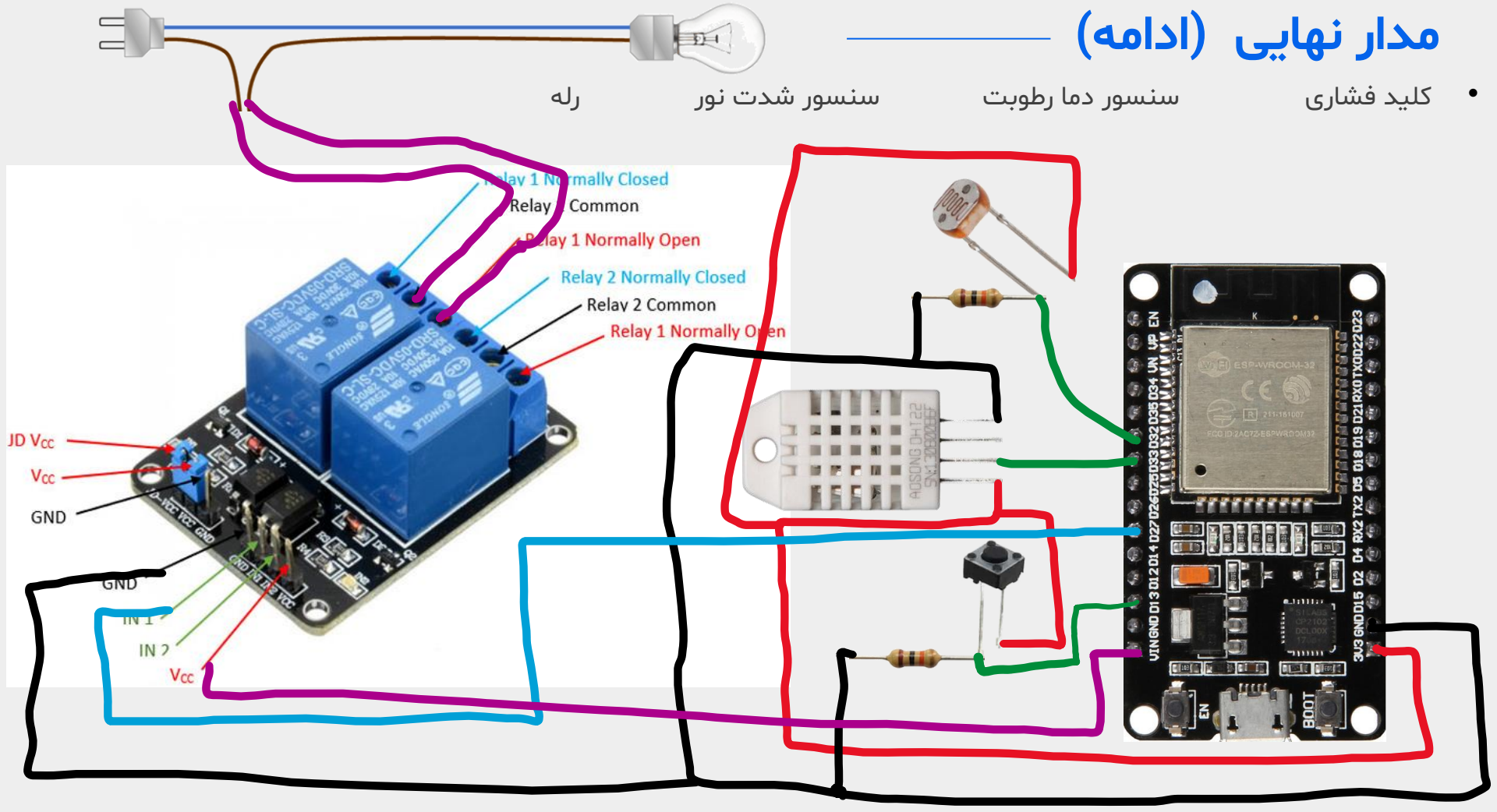
مدار نهایی



- کلید فشاری
- سنسور دما رطوبت
- سنسور شدت نور
- رله

مدار نهایی (ادامه)

- کلید فشاری سنسور دما رطوبت سنسور شدت نور رله



ممنون از
توجه شما