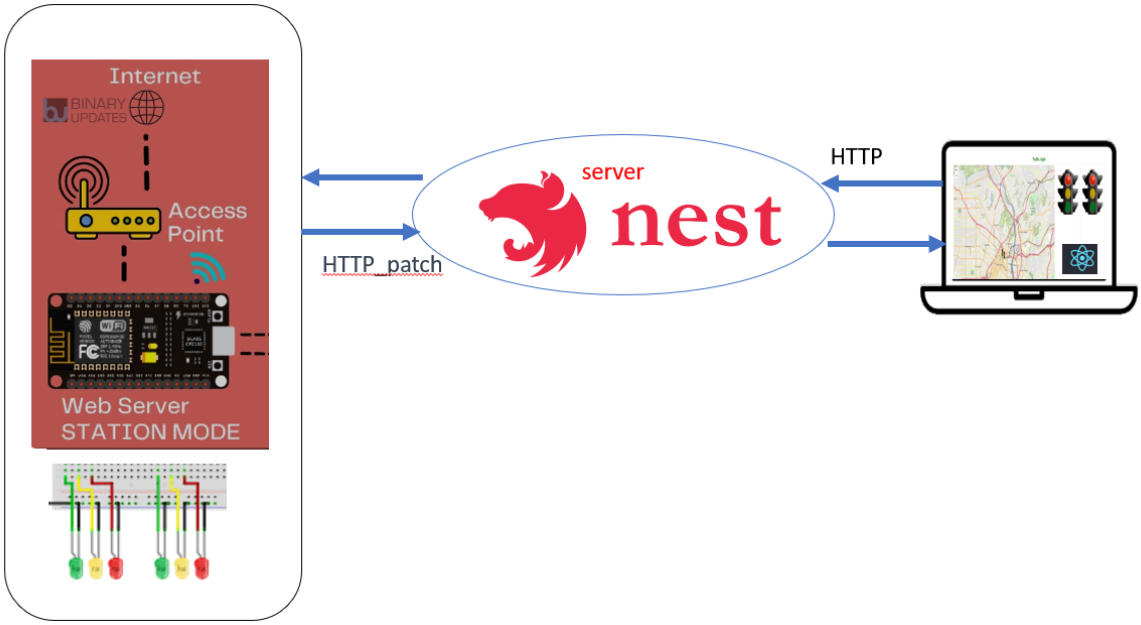
ATOUK ASMAA – NADIJ SALWA -HIMEUR MANAR

Lien de video demo https://youtube.com/shorts/IQnVncdzRrU

Projet Internet des Objets et Systèmes Mobiles Intelligents :

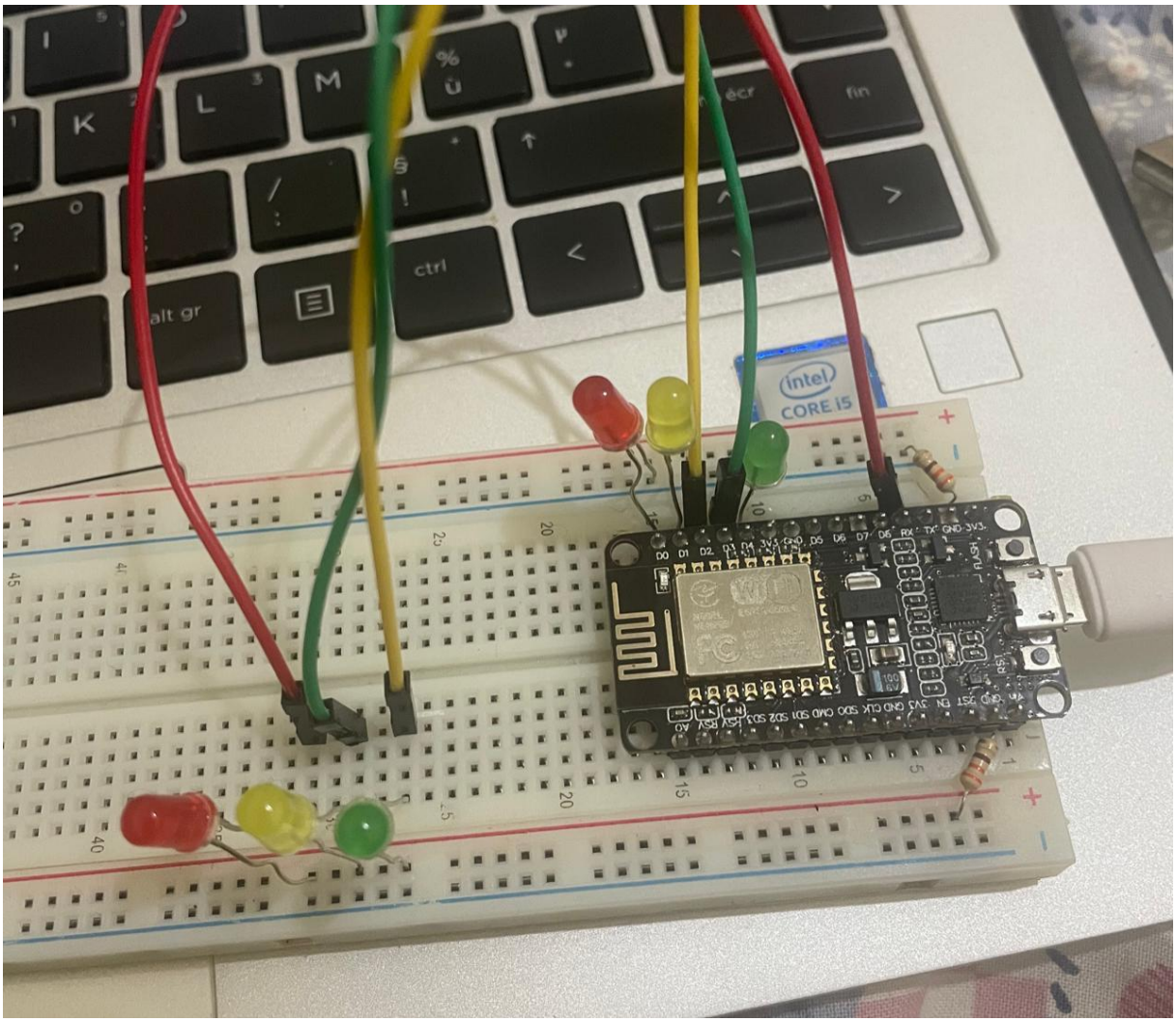
- **Architecture server(NodeMCU) to server(nest.js) with a react app :**



Ce projet met en place une architecture de communication entre un serveur NodeMCU, un serveur Nest.js et une application React implique la création d'un système dans lequel le serveur NodeMCU communique avec le serveur Nest.js, et l'application React interagit avec le serveur Nest.js pour récupérer et afficher l'etat du traffic light.

- Le serveur NodeMCU est connecté aux leds simulant deux traffics lights,et utiliser des requêtes HTTP (par exemple, HTTP POST) pour envoyer des données au serveur Nest.js.
- Le serveur Nest.js agit comme un intermédiaire entre le serveur NodeMCU et l'application React.
- L'application React est une interface utilisateur qui interagit avec le serveur Nest.js.
- Elle utilise des requêtes HTTP (par exemple, HTTP GET) pour récupérer des données auprès du serveur Nest.js.

Montage :



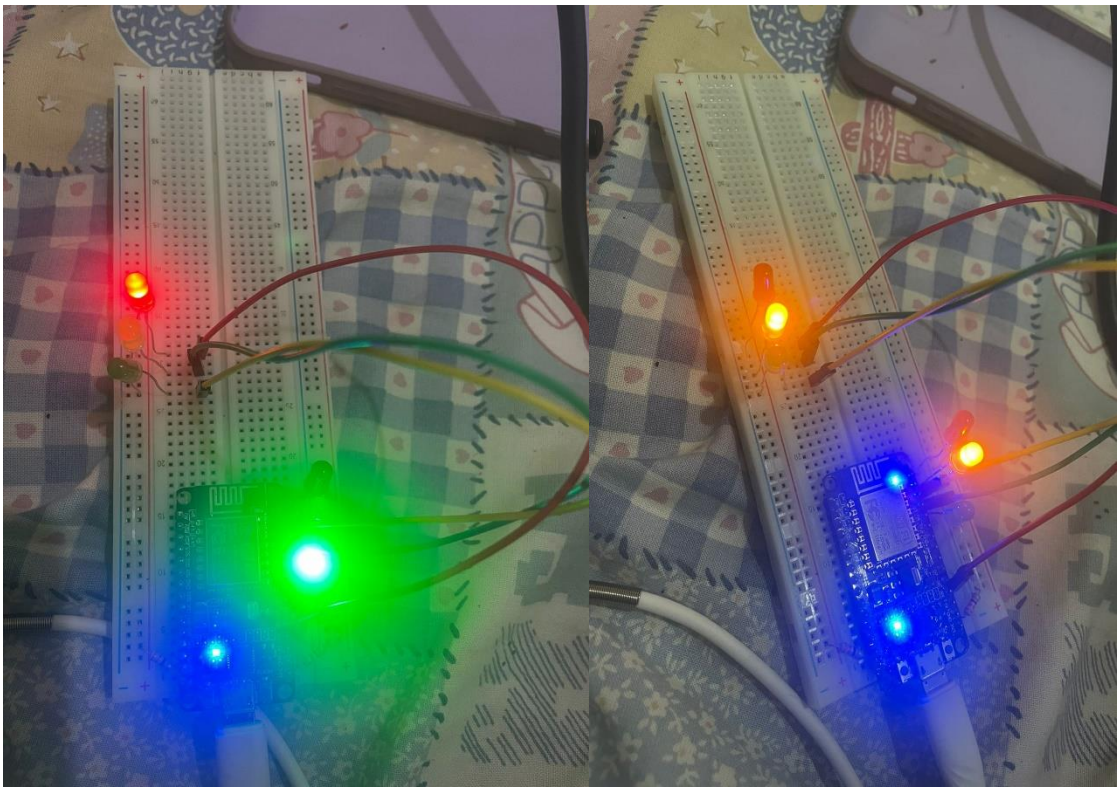

```
red1Pin = 16; //d0
orange1Pin = 5; // D1
reen2Pin = 2; // D4
red2Pin = 3; // RX
green1Pin = 4; // D2
orange2Pin = 0; // D3
```

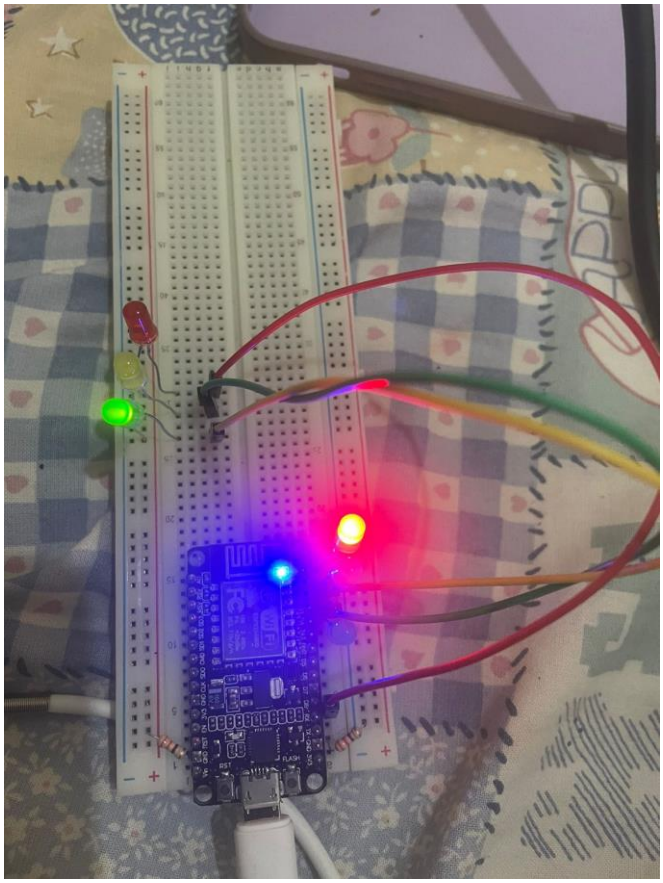
La méthode permet à l'ESP8266 d'obtenir l'état des LED depuis le serveur Nest à partir de l'application web qui gère l'état des LED est **`sendTrafficState`**

```
void sendTrafficState() {
  WiFiClient client;
  String color[2]={"gro",""};

  if (ledOrder[currentLedIndex] == red1Pin) {
      color[0]="red";
      color[1]="green";
  } else if (ledOrder[currentLedIndex] == green2Pin) {
      color[1]="red";
      color[0]="green";
  } else if (ledOrder[currentLedIndex] == orange1Pin) {
      color[0]="orange";
      color[1]="orange";
  }

  for (int i = 0; i < 2; ++i) {
    // Construct the URL for the PATCH request with color as a query parameter
    String url = "http://192.168.0.130:5000/traffic/" + trafficLightIds[i] + "?color=" + color[i];

    // Send the PATCH request
    HTTPClient http;
    http.begin(client, url);
    int httpResponseCode = http.PATCH("");

    Serial.print("HTTP Response code for ");
    Serial.print(trafficLightIds[i]);  // Use index 'i' to select the correct ID
    Serial.print(": ");
    Serial.println(httpResponseCode);
    Serial.println(url);

    http.end();
    delay(100);  // Add a small delay between requests to avoid issues
  }
```

Dans cette fonction, l'ESP8266 construit une URL pour chaque feu de circulation (identifié par trafficLightIds[i]) et envoie une requête PATCH au serveur avec la couleur actuelle de la LED correspondante (identifiée par color[i]). La requête PATCH est envoyée au serveur en utilisant la bibliothèque HTTPClient.

Ainsi, pour obtenir l'état des LEDs depuis le serveur, l'ESP8266 construit des URL spécifiques pour chaque feu de circulation en fonction des identifiants (trafficLightIds[i]) et envoie des requêtes PATCH au serveur pour mettre à jour l'état des LEDs.





Installation de Nest.js :

1. Ouvrez votre terminal dans visual studio
2. Exécutez la commande suivante pour installer le CLI Nest.js de manière globale :

3.
```
PS C:\Users\ASMAA ATOUK> cd Desktop\my-nest-project
PS C:\Users\ASMAA ATOUK\Desktop\my-nest-project> npm install -g @nestjs/cli
[...................] - idealTree:npm: sill idealTree buildDeps
```

Ensuite, définir la stratégie d'exécution créez un nouveau projet Nest.js en exécutant les commandes suivantes :

4.
```
PS C:\Users\ASMAA ATOUK\Desktop\my-nest-project> Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned
PS C:\Users\ASMAA ATOUK\Desktop\my-nest-project> nest new my-nest-project
 ⚡  We will scaffold your app in a few seconds..

? Which package manager would you ❤️ to use? npm
CREATE my-nest-project/.eslintrc.js (663 bytes)
CREATE my-nest-project/.prettierrc (51 bytes)
CREATE my-nest-project/nest-cli.json (171 bytes)
CREATE my-nest-project/package.json (1956 bytes)
```

Naviguez dans le répertoire du nouveau projet et Créez un contrôleur pour gérer les requêtes relatives aux feux de signalisation :

```
PS C:\Users\ASMAA ATOUK\Desktop\my-nest-project> cd my-nest-project
PS C:\Users\ASMAA ATOUK\Desktop\my-nest-project\my-nest-project> nest generate controller led
CREATE src/led/led.controller.ts (95 bytes)
CREATE src/led/led.controller.spec.ts (471 bytes)
```

5. Exécutez la commande suivante pour lancer le serveur de développement :

```
PS C:\Users\ASMAA ATOUK\Downloads\my-nest-project> nest start
[Nest] 9168  - 27/12/2023 01:28:57     LOG [NestFactory] Starting Nest application...
[Nest] 9168  - 27/12/2023 01:28:57     LOG [InstanceLoader] HttpModule dependencies initialized +21ms
[Nest] 9168  - 27/12/2023 01:28:57     LOG [InstanceLoader] AppModule dependencies initialized +2ms
[Nest] 9168  - 27/12/2023 01:28:57     LOG [RoutesResolver] AppController {/}: +35ms
[Nest] 9168  - 27/12/2023 01:28:57     LOG [RouterExplorer] Mapped {/, GET} route +23ms
[Nest] 9168  - 27/12/2023 01:28:57     LOG [RoutesResolver] LedController {/led}: +2ms
[Nest] 9168  - 27/12/2023 01:28:57     LOG [RouterExplorer] Mapped {/led/on, GET} route +2ms
[Nest] 9168  - 27/12/2023 01:28:57     LOG [RouterExplorer] Mapped {/led/off, GET} route +2ms
[Nest] 9168  - 27/12/2023 01:28:57     LOG [NestApplication] Nest application successfully started +3ms
```

Application Web :

Traffic light