

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

17/03/2019

# Compte rendu projet Arduino

Internet of things

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Romain Jaffé - Mohamed Khchiou

MASTER 1 MIAGE – UNIVERSITÉ NICE SOPHIA-ANTIPOLIS

## Installation et mise en place de l'environnement

- Installez Node JS : <https://nodejs.org/en/download/>
- Exécutez la commande « Node -v » pour vérifier si l'installation s'est correctement effectuée.
- Dézippez « ArduinoServer »
- Lancez la commande « node server.j » à l'intérieur du dossier pour lancer le serveur.
- Ouvrez le projet Arduino sur l'IDE « projetArduino »
  - o Modifiez l'adresse IP par celle de votre machine aux endroits indiqués. (Ipconfig).
  - o Modifier dans la méthode connect\_wifi(), les informations de connexion Wifi pour votre réseau.
  - o Compilez et téléversez le code dans votre Arduino.
- Sur votre navigateur ouvrez la page « <http://<Votrelp>:8080/esp32.html> »  
Exemple : <http://192.168.43.66:8080/esp32.html>

## Utilisation fonctionnelle

Vous arrivez sur la page web de visualisation et de contrôle de l'Arduino.

Présenté sous la forme d'un tableau, vous avez accès aux différents composants de l'Arduino, avec le pin sur lequel le composant est connecté (GPIO), sa valeur / état actuel et une action associée s'il y en a une.

Exemple pour la LED :

Branchée sur le pin 19,

Valeur : ON / OFF (allumée ou éteinte)

Action : Allumer ou éteindre la LED.

(On peut également supposer que la LED représente un petit moteur qui pourrait contrôler le verrou d'une porte pour l'ouvrir ou la fermer par exemple)

Les composants d'acquisition de données (Température et Luminosité), eux voient leurs valeurs actualisées toutes les 5 secondes.

## Fonctionnement

La communication entre l'Arduino et la page web, s'effectue grâce aux librairies `arduinoWebSockets` du côté de l'Arduino et `WebSocket` pour le serveur.

`arduinoWebSockets`: <https://github.com/Links2004/arduinoWebSockets/>

`WebSocket`: <https://github.com/theturtle32/WebSocket-Node/blob/HEAD/docs/index.md>

Comme leurs noms l'indiquent elles permettent la manipulation de sockets.

Lors du lancement du serveur, celui-ci va attendre que quelqu'un effectue une requête dessus, lorsque l'Arduino se lance, il va se connecter en envoyant un message permettant de l'identifier, cela créera le premier point de communication.

En ouvrant le navigateur à la page de contrôle, celle-ci va également envoyer un message l'identifiant et donc créera le deuxième point de communication.

Les requêtes vers l'Arduino sont envoyées sur le socket sous la forme d'un char allant de 0 à 4. Chaque numéro correspond à une demande :

- 0 ➔ récupération de l'adresse IP de l'Arduino.
- 1 ➔ récupération de l'adresse MAC de l'Arduino.
- 2 ➔ Changer l'état de la LED (allumer / éteinte) + récupération du nouvel état.
- 3 ➔ récupération de la température.
- 4 ➔ récupération de l'adresse luminosité.

Chaque donnée est envoyée sous le format `<type/valeur>`. Exemple :

« ip/192.75.1.66 »

La page web quant à elle, va récupérer chaque donnée, la split avec le séparateur '/' et remplir les champs correspondant en fonction du type.

\*Note : On aurait pu envoyer nos requêtes sous forme de string et non d'un simple char, mais par souci de simplicité un simple char est plus facile à traiter car le switch de l'Arduino ne permet pas de comparer des strings.

## Choix de techno

Notre choix s'est porté sur le Node JS pour de multiples raisons.

Tout d'abord, il faut savoir que dans notre première version, le serveur était hébergé par l'Arduino. Voulant avoir un site "un minimum beau et ergonomique", nous avons décidé de trouver un moyen pour héberger le serveur sur PC pour avoir un vrai fichier html (et plus si affinité) et faire de l'Arduino un client.

Ensuite, concernant notre choix, on voulait une techno "légère" et simple à implémenter. Ayant vu le Node JS en cours au 1<sup>er</sup> semestre, notre choix s'est porté dessus.

Nous tenons à préciser que nous avons reçu de l'aide de la part de Sofian Chaibi pour l'implémentation et l'utilisation du serveur.

Nous nous excusons également pour la partie MQTT qui est manquante, nous n'avons pas réussi à l'implémenter et par souci de temps, nous nous sommes concentrés uniquement sur la 1<sup>re</sup> partie.