

1. Installer les outils

Je vous conseille au départ de travailler avec l'IDE Arduino. C'est un IDE très simple qui vous permettra de rapidement tester votre installation à l'aide d'exemple directement accessibles. La prise en charge des bibliothèques et la création d'un projet sont également facilités. Ensuite vous pourrez utiliser un IDE professionnel comme Eclipse si vous souhaitez avoir plus de fonctionnalité au prix d'une plus grande complexité d'installation.

1/ Installez l'IDE Arduino correspondant à votre OS : <https://www.arduino.cc/en/main/software>
Prendre la dernière version stable (1.8.12 le 18 avril 2020)

Au besoin consultez : <https://www.arduino.cc/en/Guide/HomePage>

Pour Linux, il faut que l'utilisateur soit dans le groupe dialout. Le plus simple est d'exécuter les scripts fournis depuis le dossier téléchargé puis de redémarrer la machine :

```
philippe@jupiter:~/arduino-1.8.12$ ./arduino-linux-setup.sh $USER
philippe@jupiter:~/arduino-1.8.12$ ./install.sh
```

Nos cartes à base d'ESP8266 ne sont pas supportées par défaut. Il faut installer la chaîne de développement spécifique : <https://github.com/esp8266/Arduino>

La méthode la plus simple est l'installation à partir de l'IDE Arduino, suivre le paragraphe :

Installing with Boards Manager

Si vous travaillez sur un bureau métallique ou à proximité d'objets métalliques, attention aux courts-circuits qui seraient fatals pour votre carte.

2/ Branchez la carte en USB : Vous devez obtenir un nouveau port série.

Selon votre OS, la notification peut varier.

Sous Linux, le plus simple est de vérifier les messages du noyau dans un terminal :

```
philippe@jupiter:~$ dmesg
```

```
.....
[ 3188.348966] usb 2-4.2: Product: USB2.0-Serial
[ 3188.349300] ch341 2-4.2:1.0: ch341-uart converter detected
[ 3188.351038] usb 2-4.2: ch341-uart converter now attached to ttyUSB3
```

```
philippe@jupiter:~$
```

Ici on voit que la carte est reconnue sous la forme d'un port série /dev/ttyUSB3.

Pour windows ce devrait être un port série nommé COMx

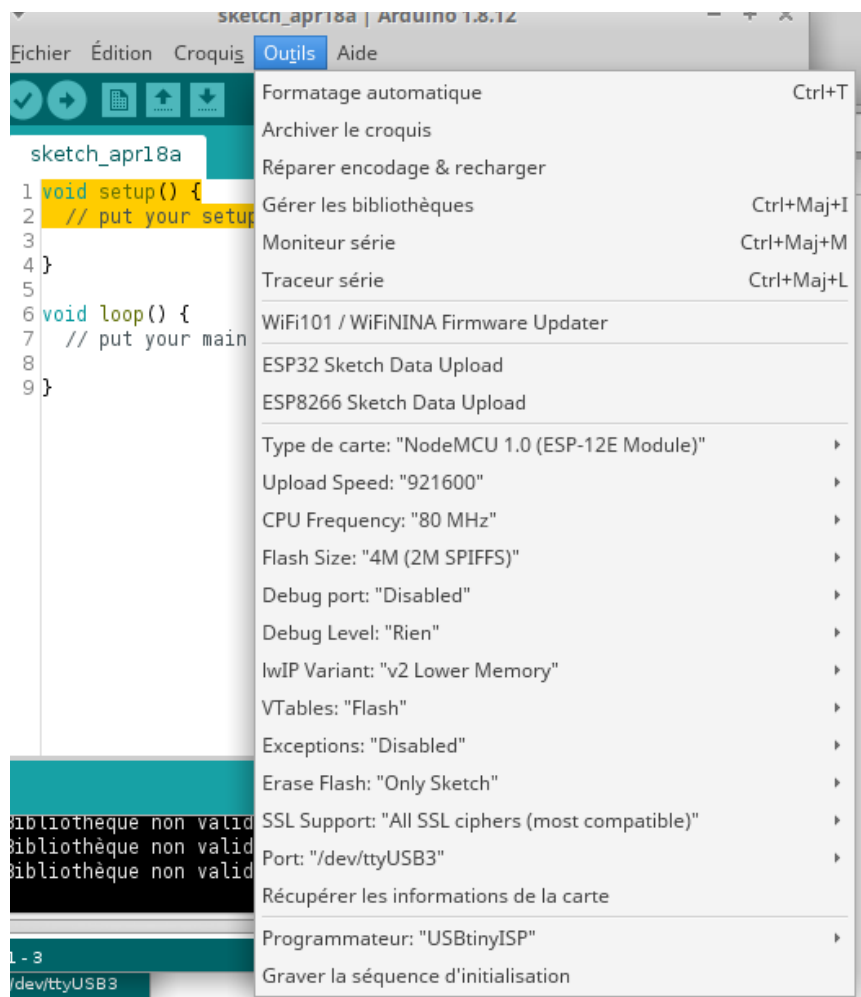
Cette étape est fondamentale. Ça ne sert à rien de continuer si vous ne détectez pas un port série. Cela signifie probablement qu'il vous manque un driver. Il faut alors l'installer et recommencer jusqu'à l'apparition du port série.

Les chips USB/UART courants sur ces cartes sont : CH340, FTDI, CP2102. Ces noms peuvent vous aider dans la recherche d'un driver pour votre OS.

2. Un premier essai : Blink

Dans le menu Outils sélectionnez votre carte : Ce devrait être "NodeMCU 1.0 (ESP12E Module)" ou une version des "LOLIN WEMOS D1".

Sélectionnez le port correspondant. (Si vous n'avez qu'une seule carte branchée ce devrait être le seul port USB disponible).



Les autres options importantes sont :

Vitesse de téléchargement : Mettre le maximum 921600, réduire en cas de problème de téléversement. Attention cette vitesse est indépendante de la vitesse du port série (115200 en général).

Taille de la mémoire Flash : Lorsqu'on utilisera la mémoire SPIFFS il faudra préciser la taille utilisée, l'autre partie étant réservée au programme.

Mode d'effacement : Only Sketch : les fichiers en SPIFFS ne sont pas effacés lors du chargement d'un nouveau programme (à utiliser par défaut).

Sketch + WiFi settings : peut être utile si les paramètres Wifi ne sont pas les bons.

All Flash contents : toute la mémoire est effacée lors du chargement d'un nouveau programme.

Pour avoir un affichage pendant que le programme travaille (compilation, téléversement) et pour voir les messages d'erreurs éventuels, dans le menu Fichier → Préférences cocher les options :

"Afficher les résultats détaillés pendant" la "compilation" et le "téléversement".

Essayer le programme **Blink** :

Fichier → Exemples → 01.Basics → Blink

Croquis → Vérifier/Compiler : la compilation doit se terminer sans erreur.

Croquis → Téléverser : le programme est transféré par la liaison USB.

La led bleue doit clignoter.

Pour les cartes ESP12E la led bleue est D0 et la led près de l'antenne est D4.

Essayez en ajoutant dans votre code :

```
#define LED D0
```

```
#define LED_ANT D4
```

Si cette étape fonctionne vous avez validé votre carte et l'installation de la chaîne de compilation.

Ça ne sert à rien de continuer si cette étape ne fonctionne pas correctement.

Si vous avez un problème, notez bien les messages d'erreur. Identifiez la partie problématique : détection du port, compilation, téléversement,...

3. Essai du Wifi

Ouvrez le programme Fichier → Exemples → ESP8266WiFi → WiFiScan

Compilez et Téléversez

Ouvrez le moniteur série (Outils → Moniteur Série) et sélectionner la vitesse adaptée (115200 baud).

Au bout de quelques secondes vous devriez obtenir l'affichage des réseaux reçus :

```
scan start
scan done
4 networks found
1: FreeWifi (-71)
2: FreeWifi_secure (-71)*
3: HomeNetwork (-72)*
4: NokiaPhone (-65)*
```

On a le nom du réseau (SSID) et le niveau de réception en dBm. Au dessus de -80dBm, il ne devrait pas y avoir de problème pour se connecter.

On peut maintenant vérifier la connexion à votre réseau WiFi.

Ouvrir le programme Fichier → Exemples → ESP8266WebServer → HelloServer

Modifier les lignes :

```
#define STASSID "your-ssid"
```

```
#define STAPSK "your-password"
```

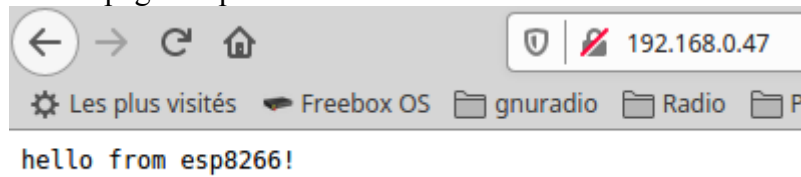
Pour mettre le SSID et le mot de passe du réseau auquel vous souhaitez vous connecter.

Compilez et Téléversez

Le moniteur série indique le SSID et l'ip obtenue (mode DHCP) :

```
.....
Connected to HomeNetwork
IP address: 192.168.0.47
MDNS responder started
HTTP server started
```

Sur une machine **faisant partie du même réseau**, tapez l'adresse de l'ESP dans un navigateur. Vous devriez obtenir la page simple



En cas de problème :

Vérifier que vous êtes bien dans le même réseau en regardant votre adresse.

Pour connaître votre adresse IP locale :

`ifconfig` dans un terminal sur Linux et MAC,

`ipconfig` dans un terminal pour Windows,

"Paramètres → A propos du téléphone → État → Adresse IP" pour un téléphone Android.

Par exemple pour un réseau de classe C, les trois premiers octets doivent être identiques. Ici 192.168.0

4. Étudier les autres exemples

Vous pouvez essayer les autres exemples fournis. Les premiers (numérotés) sont généraux de l'environnement Arduino, d'autres sont spécifiques à l'ESP8266. Certains malheureusement nécessitent un matériel supplémentaire (capteurs, écran,...).

Documentation :

Le site de référence pour les fonctions et les classes de l'environnement Arduino

<https://www.arduino.cc/reference/en>

Le site de référence pour les spécificités du portage Arduino sur l'ESP8266

<https://arduino-esp8266.readthedocs.io>