

Tutoriel ESP8266

**Configuration et mise en place
de l'ESP8266 pour une utilisation
avec WebRepl**

Club de Robotique et d'Electronique
Programmable de Ploemeur

octobre 2020

Contact pour l'information

Document réalisé en Latex par Nicolas Le Guerroué
pour le Club de Robotique et d'Electronique Programmable de Ploemeur (CREPP)
Telephone : 06.20.88.75.12
E-mail : nicolasleguerroue@gmail.com
Version du 31 octobre 2020

Permission vous est donnée de copier, distribuer et/ou modifier ce document sous quelque forme et de quelque manière que ce soit.

Table des matières

I	Préambule	3
1	Introduction	4
1.1	Présentation	4
1.2	Conventions	4
2	Prérequis	6
2.1	Matériel	6
2.2	Mise à jour des systèmes UNIX	6
2.2.1	Mise à jour de la liste des paquets	7
2.2.2	Mise à jour des logiciels	7
2.2.3	Mise à jour de Python	7
II	Installation pour Linux	8
3	Effacer la mémoire de l'ESP8266	9
3.1	Installation de Pip3	9
3.2	Installation de Esptool	9
3.3	Récupération du port USB	10
3.4	Effacer la mémoire	10
4	Installer le firmware sur l'ESP8266	12
4.1	Récupération du logiciel	12
4.2	Installation du logiciel	13
5	Configurer le mot de passe WebRepl	15
5.1	Installation de WebRepl	15
5.2	Installation de screen	16
5.3	Création du mot de passe	16
5.3.1	Utilisation de screen	16
5.4	Connexion à WebRepl	18

5.4.1	Connexion au réseau de l'ESP8266	18
5.4.2	Lancement de WebRepl	18
6	Se connecter automatiquement à un réseau Wifi	19
	Bibliographie	19

Première partie

Préambule

Section 1

Introduction

1.1 Présentation

Ce document a pour but de configurer un ESP8266-12E (NodeMCU) afin que ce dernier puisse être accessible en tant que réseau Wifi.

Ce tutoriel s'adresse également dans le cas où vous avez perdu vos mots de passe d'accès (réseau wifi ou WebRepl) ou bien que vous souhaitez partir sur des bases saines.

Information

Le temps estimé pour configurer l'ESP8266 est de 25 min

1.2 Conventions

Commandes

Les commandes à saisir sont dans des encadrés similaires :



```
sudo apt-get update
```

Exemple de commande

Références et repères

Dans un souci de clarté :

- Les fichiers sont indiqués par le repère **FILE** nom du fichier
- Les dossiers sont indiqués par le repère **DIR** nom du dossier

- Les touches du clavier et le texte à saisir au clavier sont indiqués par le repère  Raccourci clavier ou texte à saisir
- Les bibliothèques, logiciels et utilitaires sont indiqués par le repère  nom de l'utilitaire

Section 2

Prérequis

2.1 Matériel

Pour réaliser ce tutoriel, vous aurez besoin de

- Un ordinateur (Linux, Apple ou Windows)
- Un ESP8266 (NodeMCU)
- Un câble USB Micro Type-B



FIGURE 2.1 – Un câble USB Micro type-B

2.2 Mise à jour des systèmes UNIX

Avant toute chose, il convient de mettre à jour la liste des paquets et de mettre à jour les logiciels déjà présents sur votre ordinateur si ce dernier est sous LINUX (UNIX). Les commandes suivantes sont à saisir dans un terminal.

2.2.1 Mise à jour de la liste des paquets

```
sudo apt-get update
```

Mise à jour de la liste des paquets

2.2.2 Mise à jour des logiciels

```
sudo apt-get -y upgrade
```

Mise à jour des logiciels

Le -y sert à accepter automatiquement la mise à jour.

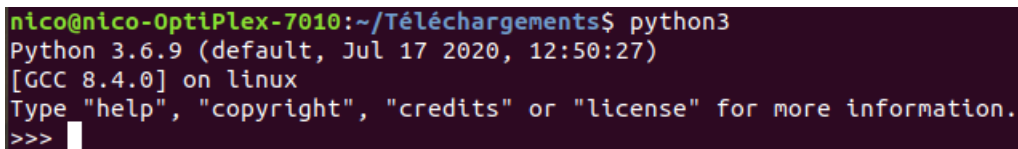
2.2.3 Mise à jour de Python

Il conviendra d'installer au minimum la version 3.6 de Python.
Pour vérifier votre version, ouvrez un terminal et saisissez la commande

```
python3
```

Vérification de la version de python

Si l'invité de commande Python suivant apparaît, la version est présente.
Pour quitter l'interpréteur python, il suffit de saisir `exit()` dans l'interpreteur ou bien de faire `Ctrl +z`



```
nico@nico-OptiPlex-7010:~/Téléchargements$ python3
Python 3.6.9 (default, Jul 17 2020, 12:50:27)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

FIGURE 2.2 – Invité de commande Python

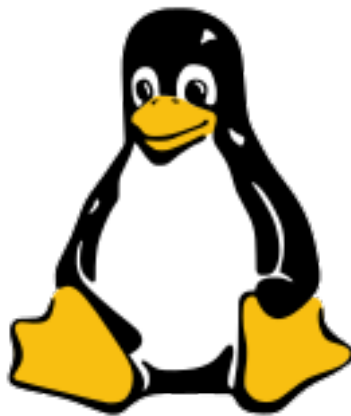
Le cas échéant, je vous invite à saisir la commande suivante :

```
sudo apt-get -y install python3.7
```

Installation de Python 3.7

Deuxième partie

Installation pour Linux



Section 3

Effacer la mémoire de l'ESP8266


Dans un premier temps, nous allons effacer le contenu de la puce ESP8266. Ceci nous permettra de partir sur des bases saines.

Point-clé

Maintenant, vous pouvez brancher votre ESP8266 sur un des ports USB de votre ordinateur.

Il convient ensuite d'installer les outils adéquats.

3.1 Installation de Pip3

 **pip3** est un utilitaire Python qui va nous permettre d'installer le petit programme pour effacer l'ESP8266.

On l'installe de la manière suivante :

```
sudo apt-get -y install python3-pip
```

Installation de Pip3

3.2 Installation de Esptool

L'utilitaire qui va se charger d'exécuter cette opération s'appelle  **esptool**.

Pour l'installer, on effectue

```
pip3 install esptool
```

Installation de Esptool

Voici le résultat de la commande sur le terminal :

```
nico@nico-OptiPlex-7010:~$ pip3 install esptool
Collecting esptool
  Downloading https://files.pythonhosted.org/packages/68/91/08c102f66fa3f12a96e754ae8ec7762abb2d778429834638f5746f81977a/esptool-2.8.tar.gz (84kB)
    100% |#####| 92kB 2.3MB/s
Collecting pyserial>=3.0 (from esptool)
  Cache entry deserialization failed, entry ignored
  Downloading https://files.pythonhosted.org/packages/0d/e4/2a744d9e3be04a0c0907414e2a01a7c88bb3915cbe3c8cc06e209f59c30/pyserial-3.4-py2.py3-none-any.whl (193kB)
    100% |#####| 194kB 3.3MB/s
Collecting pyaes (from esptool)
  Downloading https://files.pythonhosted.org/packages/44/66/2c17bae31c906613795711fc78045c285048168919ace2220daa372c7d72/pyaes-1.6.1.tar.gz
Collecting ecdsa (from esptool)
  Downloading https://files.pythonhosted.org/packages/b8/11/4b4d30e4746584684c758d8f1ddc1fa5ab1470b6bf70bce4d9b235965e99/ecdsa-0.15-py2.py3-none-any.whl (100kB)
    100% |#####| 102kB 4.3MB/s
Collecting six>=1.9.0 (from ecdsa->esptool)
  Downloading https://files.pythonhosted.org/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1c52d924a8a4cb04078a/six-1.15.0-py2.py3-none-any.whl
Building wheels for collected packages: esptool, pyaes
  Running setup.py bdist_wheel for esptool ... done
  Stored in directory: /home/nico/.cache/pip/wheels/56/9e/fd/06e784bf9c77e9278297536f3df36a46941c885eb23593bb16
  Running setup.py bdist_wheel for pyaes ... done
  Stored in directory: /home/nico/.cache/pip/wheels/bd/cf/7b/ced9e8f28c50ed666728e8ab178ffedeb9d06f6a10f85d6432
Successfully built esptool pyaes
Installing collected packages: pyserial, pyaes, six, ecdsa, esptool
Successfully installed ecdsa-0.15 esptool-2.8 pyaes-1.6.1 pyserial-3.4 six-1.15.0
nico@nico-OptiPlex-7010:~$
```

FIGURE 3.1 – Résultat de l'installation de pip3

3.3 Récupération du port USB

L'ESP8266 étant raccordé, l'ordinateur lui a affecté un nom de port de type */dev/ttyUSBx* avec x représentant le numéro du périphérique USB.

Pour récupérer la valeur de ce numéro, nous allons lancer la commande suivante :

```
ls /dev/ttyUSB*
```

Récupération du numéro du port série

```
nico@nico-OptiPlex-7010:~$ ls /dev/ttyUSB*
/dev/ttyUSB0
```

FIGURE 3.2 – Résultat de la commande

Dans le cas présent, le nom du port est */dev/ttyUSB0*

3.4 Effacer la mémoire


Lancer la commande suivante :

```
esptool.py --port /dev/ttyUSB0 erase_flash
```

Effacer la mémoire de l'ESP8266

Évidemment, si vous avez un autre numéro de port avec la commande **esptool.py flash_id**, vous mettez votre numéro.

Voici le résultat de la commande sur le terminal :



```
nico@nico-OptiPlex-7010:~$ esptool.py --port /dev/ttyUSB0 erase_flash
esptool.py v2.8
Serial port /dev/ttyUSB0
Connecting....
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 80:7d:3a:69:69:53
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 7.2s
Hard resetting via RTS pin...
nico@nico-OptiPlex-7010:~$
```

FIGURE 3.3 – Résultat de la commande pour effacer l'ESP8266

Section 4

Installer le firmware sur l'ESP8266

Maintenant que l'ESP8266 est vide, il nous reste à installer son logiciel (firmware) fin qu'il puisse utiliser le Wifi selon deux modes :

- Point d'accès : l'ESP8266 créer son propre réseau Wifi
- Connexion : l'ESP8266 peut se connecter à un réseau Wifi pour dialoguer

4.1 Récupération du logiciel

Le logiciel se présente sous fichier binaire (.bin) et est disponible à l'adresse suivante :

<http://micropython.org/download/esp8266/>

Je vous invite à télécharger la dernière version stable (latest)

Stable firmware, 1M or more of flash


The following files are stable firmware for the ESP8266. Program your board using the esptool.py program as described [in the tutorial](#).

- [esp8266-20191220-v1.12.bin](#) (elf, map) (latest)
- [esp8266-20190529-v1.11.bin](#) (elf, map)
- [esp8266-20190125-v1.10.bin](#) (elf, map)

FIGURE 4.1 – Récupération du logiciel pour l'ESP8266

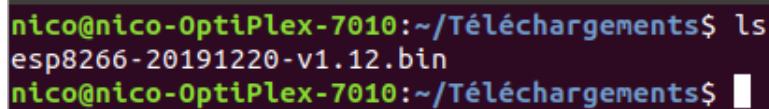
4.2 Installation du logiciel

Tout d'abord, placez-vous dans le même répertoire que votre fichier binaire installé précédemment et ouvrez un terminal.

La commande  `ls` devrait confirmer votre contenu du répertoire.

```
ls
```

Vérification du répertoire



```
nico@nico-OptiPlex-7010:~/Téléchargements$ ls
esp8266-20191220-v1.12.bin
nico@nico-OptiPlex-7010:~/Téléchargements$
```

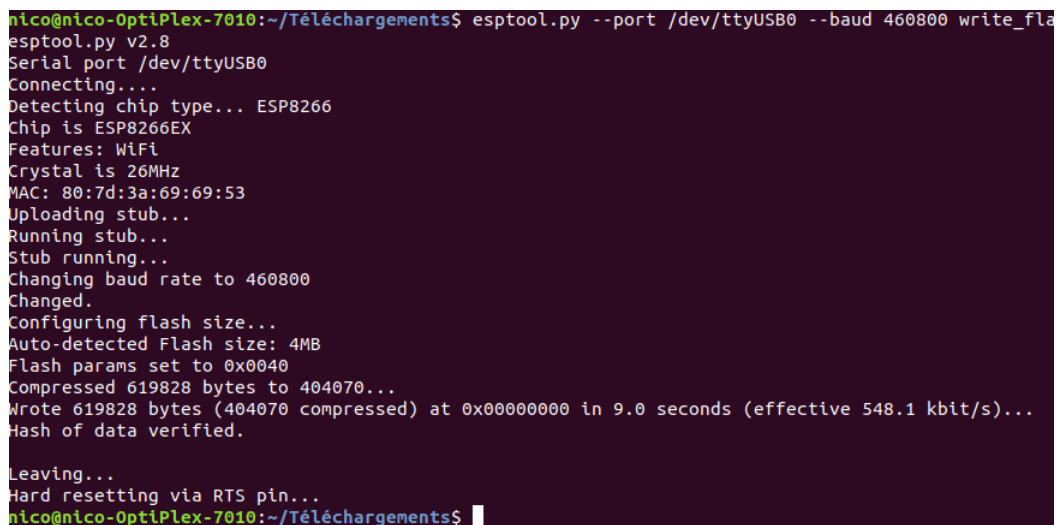
FIGURE 4.2 – Contenu du répertoire

Il ne vous reste plus qu'à saisir la commande pour installer le firmware.

```
esptool.py --port /dev/ttyUSB0 --baud 460800 write_flash --flash_size=
detect 0 esp8266-20190125-v1.10.bin
```

Installation du firmware

Comme précédemment, si vous avez un nom de fichier différent, je vous laisse le soin de changer de nom afin de coïncider avec le vôtre.



```
nico@nico-OptiPlex-7010:~/Téléchargements$ esptool.py --port /dev/ttyUSB0 --baud 460800 write_flash
esptool.py v2.8
Serial port /dev/ttyUSB0
Connecting...
Detecting chip type... ESP8266
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 80:7d:3a:69:69:53
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0040
Compressed 619828 bytes to 404070...
Wrote 619828 bytes (404070 compressed) at 0x00000000 in 9.0 seconds (effective 548.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
nico@nico-OptiPlex-7010:~/Téléchargements$
```

FIGURE 4.3 – Résultat de la commande pour installer le firmware


Après le déploiement du firmware, le module redémarre et il est configuré en point d'accès WiFi avec pour nom **MicroPython-6953**.

Les chiffres correspondent aux quatres derniers chiffres de l'adresse MAC du module.

Section 5

Configurer le mot de passe WebRepl

5.1 Installation de WebRepl

Le logiciel  va nous permettre de se connecter à l'ESP8266 afin de saisir des commandes Python.

Le logiciel est disponible à l'adresse suivante :

<https://github.com/micropython/webrepl>

Cliquez ensuite sur  puis 

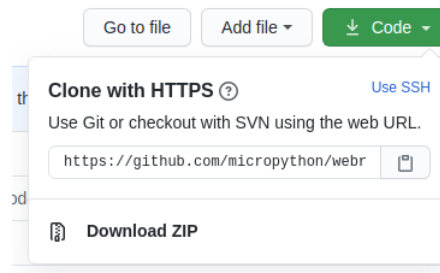


FIGURE 5.1 – Téléchargement de WebRepl sur Github

Veuillez commencer par extraire l'archive. Celle-ci contient les fichiers suivants :

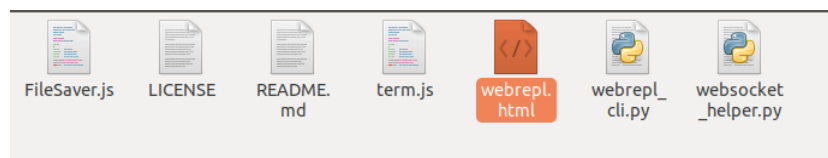


FIGURE 5.2 – Contenu du dossier WebRepl

5.2 Installation de screen

`LIB screen` est un utilitaire qui va nous permettre de se connecter à l'ESP8266 via le câble USB car actuellement, il nous est impossible d'utiliser WebRepl. On installe l'utilitaire avec la commande suivante :

```
sudo apt-get install -y screen
```

Installation de screen

5.3 Création du mot de passe

5.3.1 Utilisation de screen

On peut accéder à l'interpréteur de commande Python REPL¹ via le port série en tapant la commande suivante dans un terminal :

```
screen /dev/ttyUSB0 115200
```

Execution de screen

Point-clé

Il faut appuyer sur la touche Entrée pour afficher l'invité de commande MicroPython.

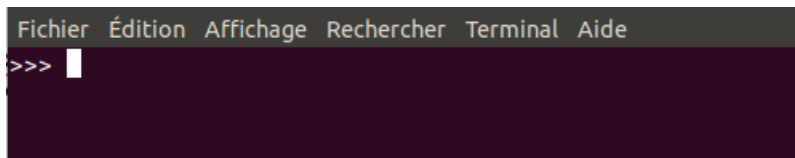


FIGURE 5.3 – Console screen

Pour quitter Screen, il faut appuyer sur les touches `KEY CTRL + a` puis écrire `KEY :quit`

Ensuite, entrez la commande suivante via le terminal Screen :

```
>>> import webrepl_setup
```

Commande pour créer un mot de passe

Le système vous demande tout d'abord d'activer l'accès par le réseau Wifi dès le démarrage en saisissant `KEY E`.

1. Read Evaluate Print Loop

```
>>> import webrepl_setup
WebREPL daemon auto-start status: disabled

Would you like to (E)nable or (D)isable it running on boot?
(Empty line to quit)
> E
```

FIGURE 5.4 – Activation de l'ESP8266 screen

Il vous invite ensuite à saisir le mot de passe pour l'accès à WebRepl. Ici le mot de passe choisi est `KEY crepp`

```
>>> import webrepl_setup
WebREPL daemon auto-start status: disabled

Would you like to (E)nable or (D)isable it running on boot?
(Empty line to quit)
> E
To enable WebREPL, you must set password for it
New password (4-9 chars): crepp
Confirm password: crepp
Changes will be activated after reboot
Would you like to reboot now? (y/n) y
```

FIGURE 5.5 – Activation de l'ESP8266 screen

Enfin, saisissez `KEY y` pour redémarrer l'ESP8266.
A ce moment là, les lignes suivantes apparaissent :

Section 6

Se connecter automatiquement à un réseau Wifi

L'inconvénient avec la méthode présente est de jongler entre les 2 accès WiFi (Wifi classique ou réseau ESP8266).

Or, il est possible de configurer le module pour qu'il se connecte sur votre box WiFi en tant que client afin d'éviter les désagréments des connexions WiFi.

Pour cela il suffit de se connecter à l'ESP8266 via le port série et de taper les commandes suivantes :

```
import network
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect('ssid', 'password')
```

Commandes de connexion

Vous pouvez vérifier la nouvelle adresse IP fournie par votre box en tapant la commande :

```
>>> wlan.ifconfig()
```

Vérification

Par contre il est nécessaire de passer les commandes suivantes après la connexion à votre box :

```
>>> import webrepl
>>> webrepl.start()
```

Initialisation

Ce tutoriel est désormais terminé.

Bibliographie

- [1] *Présentation de l'ESP8266 NodeMCU*
<https://emirchouchane.com/partie-i-microcontrôleur-esp8266/>
- [2] *Installation du firmware de l'ESP8266*
<https://docs.micropython.org/en/latest/esp8266/tutorial/intro.html#intro>
- [3] *Installation du firmware de l'ESP8266 et configuration du mot de passe*
<https://itechnofrance.wordpress.com/2018/02/04/installation-de-micropython-sur-nodemcu-lolin-ou-wemos-d1-mini/>