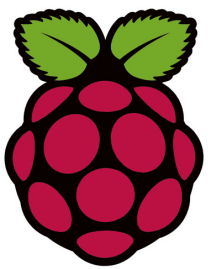
# Atelier StarinuX Raspberry Pi

## Partie 3

Pascal Bessonneau

12/2016



# Privoxy

Privoxy est un proxy qui ne met pas en cache mais filtre le contenu des pages lues pour enlever notamment les publicités et peut également modifier les pages web.

# Privoxy

Pour l'installation il suffit de faire un :

```
$sudo apt-get install privoxy
```

# Installation

Il y a quelques réglages à faire. Tout d'abord il faut le configurer pour qu'il écoute le réseau local.

```
$sudo nano +761 /etc/privoxy/config
listen-address  192.168.0.56:8118
```

# Configuration

On va enlever le logging de privoxy pour éviter de stocker votre historique Web (bien que par défaut privoxy ne stocke que les évènements graves). Vous devez le réactiver si vous avez des problèmes de navigation pour le débogage.

```
$sudo nano +455
#logfile logfile
```

# Configuration

Pour limiter l'accès à votre réseau local (c'est mieux...), il suffit de faire :

```
$sudo nano +1062
permit-access  192.168.0.0/24
```

# Configuration

On redémarre le serveur pour tenir compte des changements :

```
$sudo service privoxy restart
```

# Pare-feu

Ensuite il y a un nouveau port ouvert, le port 8118, il faut donc modifier le firewall. Soit avec :

```
$sudo dpkg-reconfigure arno-iptables-firewall
```

# Pare-feu

ou bien dans le fichier du firewall

```
$IPTABLES -A TCP -p tcp --dport 8118 -j ACCEPT
```

# Dans le navigateur

Ensuite il faut dans votre navigateur, régler le proxy :

- **Firefox** je vous conseille d'utiliser l'extension FoxyProxy qui permet de changer à la volée le proxy
- **Chromium** SwitchyOmega qui permet de changer à la volée le proxy

# Dans le navigateur

Si vous voulez que le proxy se configure automatiquement pour les machines de votre réseau. Il faut pouvoir ajouter une ligne à la configuration de votre serveur DHCP et mettre à disposition un fichier sur le serveur web. La manipulation est expliquée ici.

# Filtrage

Les fichiers “user.action” et “user.filter” sont là que vous allez faire vos modifications.
Par exemple si vous souhaitez qu'un site ne soit pas filtré, comme numerama un super site que vous voulez aider en acceptant la pub.
Il faut faire :

```
$sudo nano +158 user.action
.numerama.com
```

# Filtrage

Par défaut, les cookies sont autorisés seulement s'il s'agit de cookies de session, si vous voulez qu'un cookie à plus long terme soit conservé, il faut ajouter le site dans la partie adéquate :

```
$sudo nano +90 user.action
.github.com
```

# Filtrage

Vous pouvez utiliser le fichier trust pour limiter les web à visiter comme par exemple limiter l'accès seulement à quelques sites, pour les enfants par exemple.

```
$sudo nano +511 config
trustfile trust
```

Dans trust, vous pouvez les sites qui sont autorisés :

```
~.starinux.org
```

N'oubliez pas de relancer le serveur.

# Qu'est ce que qu'un VPN ?

Le but est de créer un serveur VPN. Qu'est ce que le VPN ? C'est l'acronyme de Virtual Private Network. Il va créer un pont crypté entre deux ordinateurs et les requêtes réseaux vont être déportés sur le serveur. Toutes les requêtes réseaux du client vont sortir seulement à travers du tunnel crypté.
Son utilité ? Avoir son serveur VPN est utile par exemple quand vous êtes dans un Starbucks ou dans un hotel.
Il est à noter que des serveurs VPN commerciaux existent.

# Qu'est ce que qu'un VPN ?

Nous ferons un VPN utilisant le protocole le plus sécurisé : le protocole OpenVPN.
La cryptage est asymétrique avec une paire clef privée et clef publique. Il y a une paire de clefs pour le serveur et une paire pour le client.

# Installation

L'installation à la main supposerait :

1. télécharger les paquets nécessaires
2. créer les deux paires de clefs
3. configurer le serveur
4. préparer le fichier client
5. installer la connexion VPN sur le client avec le fichier client

# Installation

Fort heureusement un script est disponible ici. Il facilite la manœuvre et le rend interactif :

```
$wget https://git.io/vpn -O openvpn-install.sh
$sudo bash openvpn-install.sh
```

# Installation

Le premier renseignement demandé est l'IP du Raspberry sur le réseau interne. Pour moi 192.168.0.56.
Le second est le port par défaut. Ce n'est pas la peine de le changer.
Ensuite vous devez choisir les DNS qui seront utilisés pour que le serveur puisse résoudre les noms de domaine en sortie du VPN (OpenNIC de préférence)

# Installation

Ensuite il vous demande le nom du fichier à créer pour configurer le client.
Il vous demande de fournir l'"external IP", c'est qu'il a remarqué que vous étiez derrière un NAT donc il demande l'adresse publique.
Enfin il vous demande le mot de passe pour vous connecter au VPN.

# Installation

Après le script va télécharger et installer les paquets. Il installe notamment :

- OpenVPN, qui est le programme permettant de se connecter à un VPN ou qui permet de faire un serveur VPN.
- Easy-RSA, sur un dépot git, c'est le programme qui permet de créer facilement des clefs RSA pour le client et pour le serveur.

L'étape la plus longue est la création de la clef privée du serveur qui peut parfois prendre dix minutes sur un Raspberry Pi 2 si il y a un manque d'entropie.

# Configuration

Dans le répertoire courant, un nouveau fichier ".ovpn" est apparu. C'est le fichier client qui servira à configurer votre portable.

```
client
dev tun
proto udp
...
```

# Configuration

Dans le répertoire “/etc/openvpn”, vous trouverez les clefs du serveur ainsi que le fichier de configuration, “server.conf”, auquel il faut faire des modifications.
Il faut modifier la première ligne dans ce fichier de configuration en rajoutant :

```
local 192.168.0.56
```

# Configuration

Ce qui donne comme fichier de configuration :

```
local 192.168.0.56
port 1194
proto udp
...
```

# Pare-feu

Le serveur openvpn se comporte pour ses clients comme un serveur DHCP.
La ligne “server” indique une plage d’adresse doit être une plage d’adresse qui est *absolument différente* de la plage d’adresse que vous utilisez pour votre réseau interne.
Cette plage d’adresse contient les IPs des clients du VPN.

# Pare-feu

Vous savez donc que cette plage d'adresse est le réseau "interne" pour le firewall. Vous avez aussi besoin de savoir que l'interface réseau créée par un VPN est "tun0". Vous pouvez le vérifier en tapant :

```
$ifconfig
```

# Pare-feu

Il faut modifier le firewall pour qu'il fasse du NAT c'est-à-dire qu'il fasse suivre les paquets du réseau créé par le VPN vers les adresses extérieures.
Pour ça, si vous utilisez arno-iptables-firewall, il suffit de modifier le fichier de configuration :

```
$sudo vi /etc/arno-iptables-firewall/conf.d/00debconf.conf
```

# Pare-feu

Et le fichier doit avoir cette tête :

```
EXT_IF="eth0"
EXT_IF_DHCP_IP=1
OPEN_TCP="22 8118"
OPEN_UDP="1194"
INT_IF="tun0"
NAT=1
INTERNAL_NET=""
NAT_INTERNAL_NET="10.8.0.0/24"
OPEN_ICMP=0
```

# Pare-feu

Vous remarquez les réglages pour le réseau interne et le fait qu'on ouvre un port en UDP au port 1194, c'est le port du serveur.
Attention, le protocole est le protocole UDP et non TCP.
Ensuite il faudra que sur votre box, vous mettiez une redirection de ports. La manipulation se trouve partout sur internet car c'est la même que pour les applications de partage de fichiers.
Ici la redirection sera le port 1194 en UDP vers l'adresse du Pi sur le port 1194.

# Pare-feu

Pour un firewall “à la main” il faudrait rajouter quelque chose comme ça en fin de fichier de configuration du firewall :

```
$iptables --table nat --append POSTROUTING --out-interface
$iptables --append FORWARD --in-interface tun0 -j ACCEPT
```

# Pare-feu

Ensuite il faut activer le NAT dans le kernel :

```
$sudo nano /etc/sysctl.conf
```

et décommentez la ligne :

```
net.ipv4.ip_forward=1
```

puis redémarrer ou exécuter

```
$sudo sysctl -p /etc/sysctl.conf
```

# Ajouter un nouveau client

Pour ajouter un nouveau client, un autre ordinateur à la liste des ordinateurs pouvant se connecter, il faut simplement relancer le script et suivre les instructions.

Il vous faut le fichier ".ovpn" créé tout à l'heure. Il contient toutes les informations pour vous connecter au serveur.
Sur Linux, il faut installer le paquet "openvpn" et si vous utilisez GNOME les paquets pour le gestionnaire de réseaux.

```
$sudo apt-get install openvpn
$sudo apt-get install network-manager-openvpn
$sudo apt-get install network-manager-openvpn-gnome
```

Pour KDE, pas besoin du paquet gnome...

# Côté client

Pour les gens sous Windows, le logiciel à installer est à télécharger sur le site d'OpenVPN. Il était dans le package que j'avais préparé en avance.

# Le GPIO *General Purpose Input Output*

C'est la série de broches situé sur un côté du Raspberry. Sa taille a varié entre le premier Raspberry et les suivants. On a gagné quelques broches.
Sur la figure 1, du site element14, est représenté le GPIO pour le Raspberry Pi 3, il faut que vous cherchiez sur internet celui qui correspond à votre modèle de Pi si nécessaire.

# Le GPIO

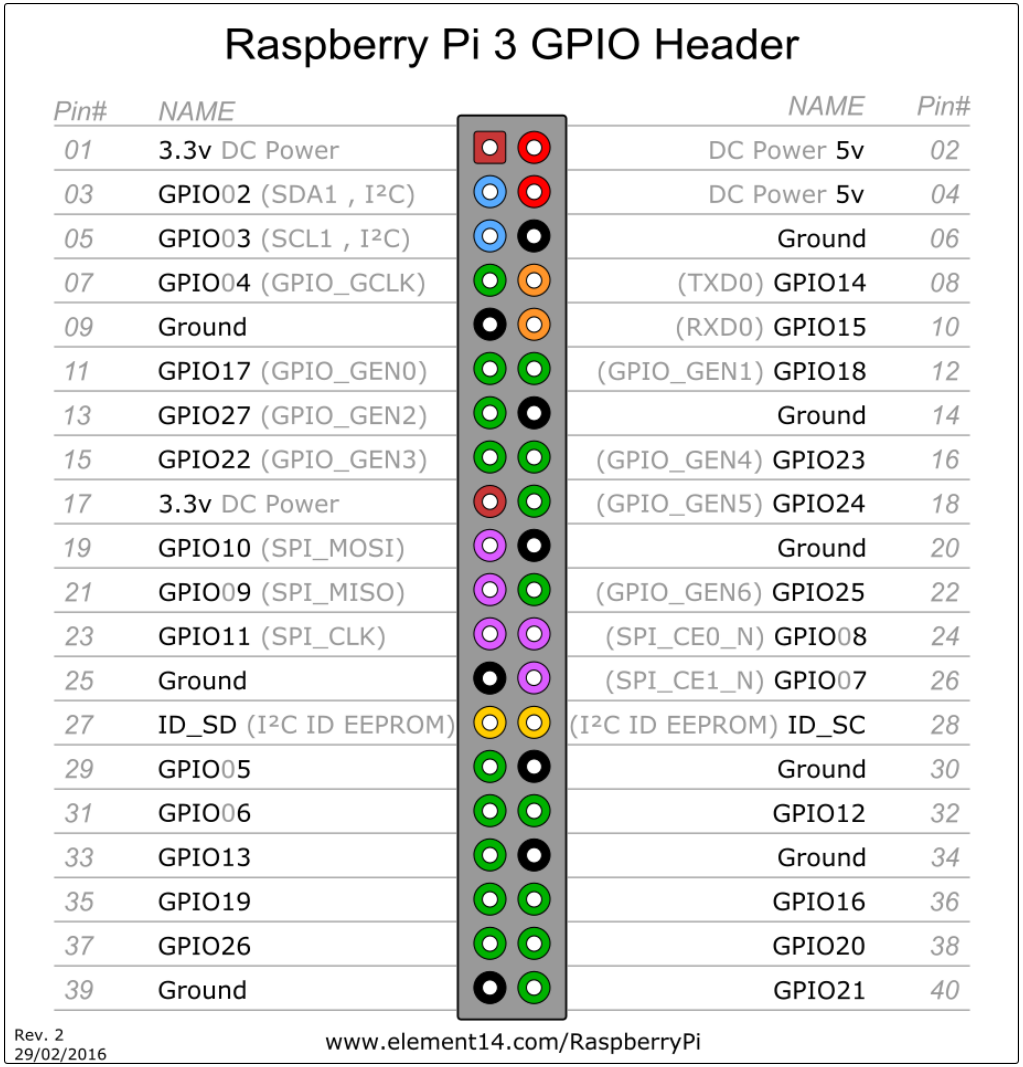



FIGURE: Le GPIO du Raspberry Pi 3

# Le GPIO

Vous pouvez remarquer qu'il y a des alimentations de 3,3V et de 5V (pin 01, 02, 04, 17) selon les périphériques que vous voulez brancher. Et Il y a des masses *Ground*.
Les autres broches sont sur du courants de 3,3V. Le courant en sortie peut varier de 2 à 16mA.

# Le GPIO

Le GPIO permet de soit de contrôler soi même des éléments électroniques soit de mettre des extensions (HAT) pour avoir des périphériques. Vous en aurez à disposition le jour de l'atelier en petit nombre.
Le plus simple pour gérer ces sorties est le langage Python. Une bibliothèque Python est fourni pour contrôler le GPIO et aussi les périphériques qu'on branche dessus (le plus souvent).
Sinon vous avez la possibilité de contrôler le GPIO via un programme en C.

# Une diode

C'est la partie que je maitrise pas trop, c'est la partie électronique.

# L'aveu

Pour allumer une diode, il faut mettre la résistance adéquate pour ne pas griller la diode. Pour une tension de 3,3V et avec une diode qui fait baisser la tension de 0,7V il faut absorber $3,3 - 0,7 = 2,6V$. Avec la loi d'Ohms, $U = R/I$, on calcule pour un courant de 5mA la résistance nécessaire :
$2,6/(5/1000) = 520ohms$. La résistance la plus proche est de 570 ohms.
Il faut chercher la résistance qui va bien. Installer fil, diode et résistance sur la breadboard.

# Une diode

Ensuite le programme Python est assez simple :

```
import Rpi.GPIO as GPIO
import time
...
```

# Une diode

Pour faire un programme qui va faire clignoter indéfiniment la diode.

```
import Rpi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)

while TRUE:
...
```

# Une diode

Attention en l'arrêtant avec controle+C vous allez laisser la broche dans un état ambigu.
Il vaut mieux quitter proprement :

```
import Rpi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)

while TRUE:
    try:
...
```

# Le PiFaceCAD

Le PiFace est une carte avec un afficheur deux lignes 16 caractères, des boutons poussoirs et un un capteur infra-rouge.
Pour l'utiliser il faut activer la sortie SPI dans 'raspi-config', redémarrer puis installer le paquet python-piface.

```
$sudo apt-get install python-pifacecad
```

# Le PiFaceCAD

```
import pifacecad
import time
import netifaces as ni
...
```

# Le PiFaceCAD

Pour utiliser les boutons poussoirs, on crée un processus qui va surveiller et faire un “goto” dans une fonction pour traiter l’information.

```
import pifacecad
import time

...
```

# Sense HAT

Il s'agit d'un "chapeau" ayant comme capteur :

- Gyroscope
- Accélèrometre
- Magnetomètre
- Capteur de temperature
- Capteur d'humidité
- Capteur de pression barometric

# Sense HAT

Il possède aussi un affichage avec une matrice de 8x8 en RGB.
Il est célèbre pour avoir voyagé dans l'espace.

# Sense HAT

Ci-dessous un script pour stocker dans un fichier l'humidité et la température que j'utilise chez moi.

```
#!/usr/bin/python
import sys
import os
import datetime

from sense_hat import SenseHat
...
```

# Description

Les caméras Pi sont de deux générations, les premières ont un capteur de 3 millions de pixels et les secondes de 5 millions.
Attention quand vous manipulez ces caméras, elle sont très sensibles à l'électricité statique. Par conséquent avant de les prendre en main il faut toucher un objet à la terre comme la cage de votre ordinateur ou un radiateur.
Elles se branchent sur le port CSI, entre le port HDMI et le jack audio/video. Les parties conductrices du ruban doivent se trouver vers la prise HDMI.

# Description

Il y a deux types de caméras :

- la caméra standard. C'est une caméra qui prend des photos ou vidéos que dans le visible.
- la caméra IR. C'est une caméra qui peut capturer des infra-rouges. Attention il faut une source de lumière adaptée comme des LEDs infra-rouges ou un dispositif pour les caméras extérieures. Quand la lumière dans le visible est forte elle capture le "visible" comme la caméra standard. Elle a donc un double usage.

Pour activer la caméra il faut lancer "raspi-config" et dans le menu activer la caméra.

# Capturer des images

Le programme pour capturer des images s'appelle raspistill.
pour capturer une image il suffit de faire :

```
raspistill -o testcapture.jpg
```

# Capturer des images

Pour régler la définition, il y a les arguments “w” et “h”.

```
raspistill -w 1920 -h 1080 -o fullhdcapture.jpg
```

# Capturer des images

Attention il y a un temps de latence avant la prise de vue ! Par défaut ce temps est de 5 secondes.
Pour éliminer ce temps de latence il faut utiliser le paramètre “t”.
Le temps à indiquer est un entier en millisecondes.

```
# pas de temps de latence
raspistill -t 1 -o tensecondcapture.jpg
# temps de latence d'une minute
raspistill -t 60000 -o tensecondcapture.jpg
```

Le voyant de la caméra est rouge quand on prends un cliché.

# Capturer des vidéos

Cette fois la commande est “raspivid”. Par défaut elle encode en h264, un format propriétaire.

```
raspivid -o testvideo.h264
```

Le temps de capture par défaut est de 5 secondes.

# Capturer des vidéos

Pour le modifier, il suffit d'utiliser l'argument "t" toujours en millisecondes

```
raspivid -t 60000 -o testvideo.h264
```

# Capturer des vidéos

Pour changer la résolution ce sont les mêmes arguments que pour “raspistill” :

```
raspivid -w 1280 -h 720 -t 60000 -o testvideo.h264
```

Selon les modèles de caméras, les possibilités en matière de définition varient.

# Timelapse

Vous pouvez programmer la caméra pour prendre des photos à intervalle régulier. Par exemple pour prendre durant une minute une image toutes les secondes :

```
raspistill -o frame%08d.jpg -tl 10000 -t 600000
```

Dans ce cas vous verrez des fichiers appelés frame00000001.jpg, frame00000002.jpg, etc. dans le répertoire courant.

# Timelapse

Pour les transformer en vidéo, il faut utiliser un utilitaire “avconv”.
Pour l'installer :

```
$sudo apt-get install libav-tools
```

# Timelapse

Ensuite :

```
avconv -r 10 -i frame%08d.jpg -r 10 -vcodec libx264 timelap
```

L'argument r indique le nombre de frames par secondes.
L'argument vcodec indique le type de compression.
"avconv" est un outil très complexe avec lequel on peut beaucoup de choses.