

Cartes EyeFi - hack

[Retour à la page principale](#) → [Open Path View](#)

Note pour la clé wifi Brest-wireless faut installer :

```
sudo apt-get install zd1211-firmware
```

Créer un AP wifi sur la Raspberry Pi

```
apt-get install hostapd
```

Ajouter/modifier dans /etc/default/hostapd :

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"  
RUN_DAEMON="YES"
```

Fichier de conf /etc/hostapd/hostapd.conf :

```
# interface wlan du Wi-Fi  
interface=wlan0  
  
# nl80211 avec tous les drivers Linux mac80211  
driver=nl80211  
  
# Nom du spot Wi-Fi  
ssid=OpenStreetView  
  
# mode Wi-Fi (a = IEEE 802.11a, b = IEEE 802.11b, g = IEEE 802.11g)  
hw_mode=g  
  
# canal de fréquence Wi-Fi (1-14)  
channel=6  
  
# Wi-Fi ouvert, pas d'authentification !  
auth_algs=1  
  
# Beacon interval in us (1.024 ms)  
beacon_int=100  
  
# DTIM (delivery trafic information message)  
dtim_period=2  
  
# Maximum number of stations allowed in station table  
max_num_sta=15  
  
# RTS/CTS threshold; 2347 = disabled (default)
```

```
rts_threshold=2347

# Fragmentation threshold; 2346 = disabled (default)
fragm_threshold=2346
```

Lancer :

```
sudo hostapd -B /etc/hostapd/hostapd.conf
```

Pour du wifi n

La version de hostapd présente dans les dépôts ne gère pas le wifi n (ce serait trop simple sinon). Donc il faut aller chercher la dernière version et la compiler.

Pour compiler suivre la doc ici : <http://doc.ubuntu-fr.org/hostapd>

On a pas pu tester pour l'instant tout simplement car nous n'avons pas de clé wifi n que l'on puisse passer en mode "master".

Eviter d'utiliser la version compilée, ça fout la merdre avec le driver nl80211 et plus moyen de repartir sur la version dans les dépôts après.

Serveur DHCP

```
sudo apt-get install udhcpd
```

Plage d'adresses et interface

Dans /etc/udhcpd.conf :

```
# The start and end of the IP lease block

start      192.168.42.100    #default: 192.168.0.20
end        192.168.42.150    #default: 192.168.0.254

# The interface that udhcpd will use

interface  wlan0            #default: eth0
```

IP fixes pour des tests

Pour des besoins de tests nous avons mis des IP fixes pour chaque carte EyeFi.

Ainsi vous trouverez un script python qui effectue un ping sur l'ensemble des cartes EyeFi (ayant des IP de 192.168.42.100 192.168.42.105) à pour tester leur fonctionnement.

Dans `/etc/udhcpd.conf` :

```
# --- EyeFi Cards ---
static_lease 00:18:56:2e:74:20 192.168.42.100 # EyeFi 0
static_lease 00:18:56:2f:79:98 192.168.42.101 # EyeFi 1
static_lease 00:18:56:2f:73:52 192.168.42.102 # EyeFi 2
static_lease 00:18:56:2f:77:03 192.168.42.103 # EyeFi 3
static_lease 00:18:56:2f:75:35 192.168.42.104 # EyeFi 4
static_lease 00:18:56:2e:73:95 192.168.42.105 # EyeFi 5
```

Ajout d'une connexion à internet

Nous aurons besoin d'internet sur l'AP juste pour l'ajouter sur les cartes EyeFi car elles vont essayer de se connecter au serveur eyefi, si elles n'arrivent pas on ne pourra pas ajouter l'AP dans la conf.

Editez `/etc/sysctl.conf` vérifiez que la ligne suivante n'est pas commentée :

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

Ajoutez dans `/etc/rc.local` (avant `exit 0`) :

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
exit 0
```

Normalement si tout c'est bien passé, vous pouvez relancer votre raspberry pi, l'AP va démarrer automatiquement. Vérifiez bien que vous pouvez vous y connecter et que vous recevez bien un adresse IP et que vous avez internet dessus.

Source : <http://thenewbieblog.wordpress.com/2012/05/01/wifi-hotspot-setup-on-ubuntu/>

"faux" serveur eyeFi

Nous sommes partis du code publié par tachang sur [github](#). Ce serveur une lancé sur le même réseau qu'une carte EyeFi réceptionne les images et vidéos.

Comment ça marche ?

La carte EyeFi communique en HTTP avec l'EyeFi Center pour lui transférer les images et vidéos.

On remplace donc l'EyeFi Center par un server web python. Si les spécifications vous intéresses elles sont en parti disponibles [ici](#) (c'est du reverse ingenering elles ne seront probablement jamais complètes).

Nous avons modifié ce code pour prendre en compte plusieurs cartes EyeFi en même temps.

Le code source

Le code est dispo ici : https://github.com/Benvii/osv_eyefi

C'est pas encore très propre comme code mais ça marche :)
Il faudrait mieux commenter les modifications virer ce qui est inutile.

Installation

```
sudo apt-get install git
git clone https://github.com/Benvii/osv_eyefi.git
```

Configuration

Nous utilisons pour l'instant le fichier de configuration DebugSetting.ini.
Après avoir configuré vos cartes SD EyeFi sous windows vous devez récupérer les 'UploadKey' de chaque carte elles sont dans le fichier **/Users/benjamin/Application Data/Eye-Fi/Settings.xml**.

Par exemple pour la carte dont l'adresse MAC est '00-18-56-2e-73-95' :

```
<Card MacAddress="00-18-56-2e-73-95">
  <UploadKey>cd3bce559bd150c39428ce0438b853e9</UploadKey>
  <DownsyncKey>d214fe653fb6ca9bf4f894e5b35cfb49</DownsyncKey>
  <TransferMode>546</TransferMode>
  <TransferModeTimestamp>1373120453</TransferModeTimestamp>
  <MediaTypes>
    <Media Type="1">
      <Folder>C:\Users\benjamin\Pictures\Eye-Fi</Folder>
      <AddDate>1</AddDate>
      <DateType>1</DateType>
      <CustomDateFormat></CustomDateFormat>
    </Media>
    <Media Type="2">
      <Folder>C:\Users\benjamin\Videos\Eye-Fi</Folder>
      <AddDate>1</AddDate>
      <DateType>1</DateType>
      <CustomDateFormat></CustomDateFormat>
    </Media>
  </MediaTypes>
</Card>
```

L'UploadKey est **cd3bce559bd150c39428ce0438b853e9**.

On ajoute donc dans le fichier ini de configuration :

```
[0018562e7395]
UploadKey=cd3bce559bd150c39428ce0438b853e9
```

Utilisation

Lancer le serveur (dans un screen) à l'aide de pyhon2.7 (déjà installé sur une RPi) :

```
python2.7 EyeFiServer.py -c DebugSettings.ini
```

Configuration des cartes EyeFi - Sous windob

Vous aurez besoin d'un pc avec windob de fonctionnel pour utiliser l'EyeFi Center et configurer les cartes SD pour qu'elles se connecte sur votre raspberryPi.

Configuration des cartes sous windob



[Insérer la carte EyeFi avec de préférence le lecteur qui est fournis avec.](#)

Installer le logiciel EyFi disponible sur leur site.

Cliquez sur configuration :



Respectez la configuration suivante (cliquez pour agrandir les images) :



Ce qu'il faut surtout retenir :

- Ajouter l'AP wifi de la raspberry Pi "OpenStreetView", ne surtout pas en mettre d'autre (risque de choix du mauvais AP par la carte)
- Activer les transferts (photo qui nous intéresse surtout) vers l'ordinateur
- Désactiver les transfert en ligne, sinon c'est EyeFi.com qui va récupérer vos images
- Activer l'Endless Memory, se mode supprime les images transférés quand la carte dépasse un seuil (fixé à 50% sur les photos d'écran)

Lancer tout ça au démarrage

TO DO

Réparer l'erreur de lecture seul

Si par exemple la carte est retiré de la GoPro alors qu'elle est allumé, le système de fichier de la carte peut planter et se retrouver en lecture seul, pour corriger cela, il faut faire un backup de la carte (recommandé en cas de plantage de la restauration) :

```
sudo dd if=/dev/sdf of=/home/mondossier/monimage.img
```

Puis on répare :

```
sudo umount /dev/sdf1
sudo dosfsck -yfv /dev/sdf1
```

Source : http://doc.ubuntu-fr.org/tutoriel/comment_reparer_clef_usb

Réparation carte SD

Pour réparer une carte SD (sur /dev/sdb1) après un arrêt brutal :

```
sudo umount /dev/sdb1
sudo fsck -v /dev/sdb1
```

Eventuellement faire le fsck sur toute les partitions de la carte SD.

From:

<https://wiki.openpathview.fr/> - **Open Path View**

Permanent link:

<https://wiki.openpathview.fr/doku.php?id=eyefi>

Last update: **2017/03/12 16:04**

