



- Introduction
- Partie 1
- Partie 2
- Conclusion
- Retour Index

Introduction

Cet article va vous proposer la procédure d'installation pour un scanner EPSON USB, et traitera du cas particulier de l'excellent Perfection 610. Il est grandement inspiré des travaux de [Karl Heinz Kremer](#), le "mainteneur" officiel du backend EPSON, avec son aimable autorisation. Ses travaux sont consultables en version originale (en anglais) sur <http://www.freecolormanagement.com/sane/index.html>.

Vous aurez bien sûr besoin d'un sous-système USB fonctionnel, mais cet article suppose que vous utilisez un noyau qui supporte directement le standard USB. C'est le cas à partir du noyau 2.2.18. Pour les noyaux antérieurs à cette version (mais seulement à partir du 2.2.13 si mes souvenirs sont bons), il existe des patches à leur appliquer pour leur communiquer les drivers USB expérimentaux des noyaux 2.3.x.

Compilation d'un noyau

Dans tous les cas, il vous faudra recompiler votre noyau. Veuillez donc à installer les sources du noyau de votre distribution, ou à récupérer le dernier noyau sur <http://www.kernel.org>. Généralement, les sources du noyau sont placés dans `/usr/src`. Si vous devez décompresser vous même les sources de votre noyau, tapez en tant que root:

```
cd /usr/src
rm -f linux
tar xvzf linux-<version de votre noyau>.tar.gz
ln -s linux-<version de votre noyau> linux
chown -R root linux
chgrp -R root linux
```

Pour configurer votre noyau, lancez:

```
make menuconfig (si vous êtes en mode console)
ou
make xconfig (si vous êtes sous X)
```

Puis rendez vous dans la partie concernant le support de l'USB (USB support), afin d'activer les options suivantes ('Y' fixe le driver en 'dur' dans le noyau, 'M' le compile en tant que module) :

- Support for USB ('Y')
- USB verbose debug messages ('Y')
- Preliminary USB Device Filesystems ('Y')
- UHCI support **ou** UHCI Alternate Driver support **ou** OHCI support ('Y' ou 'M')
- USB Scanner support
- Autres options en fonction de vos autres périphériques USB...



USB verbose debug messages n'est à compiler qu'à des fins de mise au point, car les informations qu'affiche cette option au démarrage (voir `/var/log/boot.msg`) peuvent être utiles pour comprendre la cause de l'échec de l'activation d'un sous-système USB.



Choisissez l'option UHCI, OHCI, etc. en fonction de votre carte mère. La plupart de celles-ci doivent utiliser l'UHCI, mais les Compaq, les Mac et en particulier les PC équipés d'une carte-mère ALi doivent utiliser l'OHCI. Dans le doute, si vous n'êtes pas puriste, vous pouvez compiler chacune de ces trois options en module.

La compilation du noyau en elle-même ne présente pas de difficulté. Toujours depuis `/usr/src/linux`,

tapez :

```
make dep clean bzImage modules modules_install
cp arch/i386/boot/bzImage /boot/<nom du nouveau noyau ; ex: vmlinuz.usb>
cp System.map /boot/<nom du nouveau fichier de symboles ; ex: System.usb.map>
```

Et il ne vous reste plus qu'à informer lilo que vous avez effectué des manipulations sur les noyaux. Pour ce faire, éditez le fichier `/etc/lilo.conf` et ajoutez la section suivante :

```
image=/boot/<nom du nouveau noyau>
  label=usb
  root=/dev/<votre partition qui charge linux ; ex: hda5>
  initrd=/boot/initrd
```

Tapez `/sbin/lilo` afin que le chargeur prenne en compte les changements apportés à votre système. Redémarrez, et tapez `usb` à l'invite de lilo pour lancer le système avec votre noyau spécial USB. Prêtez une attention particulière aux messages qui apparaîtront, car ils vous renseigneront sur le caractère fonctionnel de votre sous-système USB. Vous pouvez les retrouver dans `/var/log/boot.msg`.

Lectures utiles

- Le USB HOWTO disponible sur <http://linuxusbguide.sourceforge.net> peut se révéler utile à l'établissement d'un sous-système USB fonctionnel, ou tout simplement à titre de culture générale sur les périphériques USB.
- Le très pratique site [LéaLinux](http://lea-linux.org) propose un article générique sur l'installation d'un scanner USB sur <http://lea-linux.org/hardware/usbscan.php3>. Il est doublement utile puisqu'il s'attarde sur l'installation des scanners SnapScan, et présente des détails sur la compilation du noyau 2.2.15 avec les drivers USB du noyau 2.3.x.
- Et bien sûr la page qui a directement inspiré celle-ci : <http://www.freecolormanagement.com/sane/index.html>. Vous y trouverez en plus la liste de tous les scanners EPSON à ce jour supportés par son backend. Lecture indispensable avant tout achat de matériel !



- Introduction
- Partie 1
- Partie 2
- Conclusion
- Retour Index

Partie 1 : La gestion du périphérique USB

Si vous voulez utiliser une version USB d'un scanner supporté, attendez vous à un peu plus de travail qu'il n'en serait nécessaire avec une version SCSI du même scanner.



Si vous n'êtes pas un expert de Linux, choisissez, lors de la compilation de votre noyau, d'intégrer le support de votre scanner USB comme module.

Commencez par monter votre système de périphériques USB en tapant :

```
mount -t usbdevfs /proc/bus/usb /proc/bus/usb
```

Si vous voulez vous simplifier la vie, vous pouvez tout aussi bien ajouter la ligne suivante au fichier `/etc/fstab` :

```
none /proc/bus/usb usbdevfs defaults 0 0
```

Rassembler les informations sur le périphérique

Affichez les informations sur les périphériques USB. Vous pouvez faire ceci en saisissant la commande suivante dans la console une fois que le block `usbdevfs` est monté :

```
cat /proc/bus/usb/devices
```

Cela affichera des informations sur tous vos périphériques USB, les hubs, les souris, les claviers, les caméras, les tablettes graphiques, etc., ainsi que bien sur les scanner. Cherchez les lignes correspondant au scanner, et en particulier la ligne "P" présentant les informations au sujet de l'identité du vendeur (Vendor ID) et du produit (Product ID). Notez ces informations avant de passer à la section suivante de cette première partie. Si vous avez un ou plusieurs des scanners suivants, vous pouvez également relever ces informations dans la table qui suit :

Modèle	Vendor ID	Product ID
Perfection 610U	0x04b8	0x0103
Perfection 636U	0x04b8	0x0101
Perfection 1200U	0x04b8	0x0104
Perfection 1200Photo	0x04b8	0x0104



La commande `'cat /proc/bus/usb/devices'` vous permet de lister les périphériques USB détectés par le noyau, tandis que la commande `'cat /proc/bus/usb/drivers'` dresse la liste des pilotes USB chargés par celui-ci. Ces informations peuvent être utiles au diagnostic.

... si vous avez compilé le support du scanner USB en tant que module : Paramètres à appliquer au module 'scanner'

Il s'agit de la méthode recommandée, surtout si vous ne sentez pas expert.

Options de la ligne de commande 'Modprobe'

Le pilote de périphérique USB nécessite deux paramètres optionels pour en tirer le meilleur parti :

- `vendor` – USB vendor ID (identité du vendeur du périphérique USB) spécifié par l'utilisateur
- `product` – USB product ID (identité du périphérique USB) spécifié par l'utilisateur

Ces informations sont à utiliser conjointement à la commande modprobe :

```
modprobe scanner vendor=<votre vendor ID> product=<votre product ID>
```

Modification du fichier modules.conf

Bien sûr, il devient vite ennuyeux de charger le module manuellement à chaque fois que vous voulez utiliser votre scanner USB. Il est bien plus convivial d'ajouter les paramètres appropriés au fichier `/etc/modules.conf` et de laisser modprobe s'occuper de tout le reste.

```
options scanner vendor=<votre vendor ID> product=<votre product ID>
```

... si vous avez compilé le support du scanner USB en 'dur' dans le noyau : Patcher le Code Source du noyau

Il faut bien sûr informer le noyau des paramètres précédents pour qu'il puisse les employer efficacement avec votre scanner. Dans cette partie réservée aux utilisateurs experts de Linux, nous supposons que vous savez parfaitement ce que vous faites. Cherchez les lignes (déjà existantes) qui concernent les périphériques "Epson" dans le fichier source `etc/src/linux/drivers/usb/scanner.c` et ajoutez les informations concernant votre scanner. Cela ne fonctionne qu'avec les versions les plus récentes de support USB. Si vous utilisez un ancien noyau patché, vous devez rechercher `0x1606` (il s'agit en fait du VendorID du scanner UMAX) et ajouter les VendorID et ProductID de votre EPSON.

Création du fichier de périphérique

Créez le fichier de périphérique dans `/dev` conformément aux instructions du Guide de l'USB. Le nom de périphérique que vous allez définir est très important pour la suite.

Voici les commandes généralement utilisées pour la création de ce fichier :

```
mknod /dev/usbscanner c 180 48  
chmod 666 /dev/usbscanner
```



Si vous utilisez une distribution récente (de type SuSE 7.0), il y a de fortes probabilités que `/dev/usbscanner` soit déjà créé.



[Introduction](#)

[Partie 1](#)

[Partie 2](#)

[Conclusion](#)

[Retour Index](#)

Partie 2 : Configuration de SANE

SANE est l'application qui permet aux programmes de Linux de communiquer avec votre scanner. Malheureusement, à la date de rédaction de cet article, la version actuelle de SANE n'a qu'une compatibilité limitée avec les scanners EPSON. Il vous faudra donc, jusqu'à ce que cette injustice soit prochainement réparée par les développeurs de SANE, utiliser un backend spécifique (une sorte de patch à appliquer au code source de SANE pour lui faire prendre en considération les spécificités des scanners EPSON).

Tout d'abord, commencez par désinstaller tout ancien package de SANE que votre distribution pourrait avoir installé. C'est d'autant plus important qu'il pourrait se produire d'étranges interactions si d'anciens fichiers sont conservés. Vous pouvez effacer l'éventuel ancien package à l'aide de la commande (en tant que root) :

```
rpm -e sane  
rpm -e xsane
```

Récupération du backend EPSON

La dernière version disponible du backend EPSON est toujours disponible sur le site de [Karl Heinz Kremer](http://www.freecolormanagement.com/sane/index.html) : <http://www.freecolormanagement.com/sane/index.html>.

Récupération des Sources de SANE

Il y a de grandes chances que vous ayez les sources de SANE livrés avec votre distribution. Sinon, vous pouvez toujours en télécharger la dernière version sur <http://www.mostang.com/sane>.

Récupération des Sources de XSANE

Il y a de grandes chances que vous ayez les sources de XSANE livrés avec votre distribution. Sinon, vous pouvez toujours en télécharger la dernière version sur <http://www.wolfsburg.de/~rauch/sane/sane-xsane.html>.

Compilation de SANE

Décompactez les sources de SANE et le backend EPSON dans les répertoires de votre choix. Dans le répertoire de décompression de SANE se trouve un répertoire `/backend`. Faites-y glisser tous les fichiers contenus par le backend EPSON.

Déplacez vous dans le répertoire qui accueille les sources de SANE. Tapez `./configure` pour lancer la préparation à la compilation.



```
Les utilisateurs d'une distribution SuSE Linux peuvent plus  
spécifiquement taper la commande suivante (elle doit tenir en une  
seule ligne) : ./configure --prefix=/usr --exec-prefix=/usr/X11R6  
--sysconfdir=/etc --libdir=/usr/lib --sbindir=/usr/sbin  
--enable-scsibuffersize=32768 --mandir=/usr/share/man  
i386-suse-linux
```

Après cette étape de configuration, tapez :

```
make  
make install
```

la dernière commande devant être tapée en tant que root.

Configuration de SANE

Il existe deux fichiers à modifier impérativement dans `/etc/sane.d` :

- `dll.conf` ne doit présenter que la ligne suivante :
`epson`
Vous pouvez donc mettre en remarque (#) toutes les autres lignes
- `epson.conf` peut ne présenter que la ligne suivante :
`usb /dev/usbscanner`

Compilation de XSANE

Décompactez les sources de SANE dans le répertoire de votre choix. Depuis celui-ci, tapez les commandes suivantes :

```
./configure  
make  
make install
```



Vous pouvez compiler `xsane` comme un plug-in pour Gimp. C'est généralement ainsi fait par défaut lorsque Gimp est détecté par le script de configuration. Une fois le logiciel compilé et installé, créez simplement un lien dans le répertoire de plug-ins de Gimp vers le binaire de `xsane` :

```
cd /usr/local/lib/gimp/1.1/plugin-ins  
ln -s /usr/local/bin/xsane
```

Utilisez bien sûr les chemins qui correspondent à votre configuration. Une fois ceci fait, Gimp ajoutera le plug-in `xsane` au démarrage, et vous trouverez une nouvelle ligne dans `Fichier>Acquisition`.



[Introduction](#)

[Partie 1](#)

[Partie 2](#)

[Conclusion](#)

[Retour Index](#)

Conclusion

Et voilà ! Nous avons appris, au cours de cet article, comment recompiler un noyau (manoeuvre critique mais pourtant ô combien nécessaire à l'optimisation de tout système linux, et pas si compliqué que ça, en définitive, vous en conviendrez), gérer les périphériques USB et bien sûr installer les scanners de la gamme EPSON.

Il ne vous reste désormais plus qu'à vous familiariser avec l'interface austère mais très fonctionnelle de xsane et scanner tout ce qui vous passe sous la main ! Nous vous proposerons peut-être bientôt un didacticiel sur l'usage de xsane, mais d'ici là, nous vous souhaitons bien du plaisir avec votre nouveau matériel !