

# Connecter son appareil photo USB à sa machine Debian

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 9 mars 2007. Dernière mise à jour le 12 mars 2007

<https://www.bortzmeyer.org/appareil-photo-usb.html>

---

Tout le monde ayant désormais un appareil photo numérique avec connexion USB, voici quelques notes sur la configuration de la connexion du mien, si cela peut aider d'autres personnes.

Les appareils photo dont la mémoire se présente comme un disque USB forment la majorité de l'offre. Comme ils s'appuient sur un standard, ils peuvent fonctionner sur plusieurs systèmes d'exploitation, sans avoir besoin d'un pilote spécifique. Il reste à gérer le problème de l'ajout ou retrait de l'appareil. Il existe plusieurs approches pour cela. Les environnements intégrés comme KDE ou Gnome gèrent tout cela automatiquement mais, personnellement, j'aime bien pouvoir travailler directement avec le shell. Ces appareils photo se présentant comme des disques durs, je veux pouvoir utiliser `ls` et compagnie.

Je travaille essentiellement sur des machines Debian donc la configuration porte surtout sur ce système.

La solution plus simple est peut être `usbmount` <<http://usbmount.alioth.debian.org/>>. Une fois installé ce programme (`aptitude install usbmount`), sans aucune configuration (mais on peut la changer si on n'est pas d'accord avec les choix par défaut), la mémoire de l'appareil photo est montée automatiquement et se voit comme un disque normal :

```
% mount
/dev/sdb1 on /media/usb0 type vfat (rw,noexec,nodev,async,noatime,gid=1000,uid=1000)
```

On peut alors faire ce qu'on veut :

```
% cd /media/usb0/dcim/100_pana
% ls
...
% jpeginfo p1000062.jpg
p1000062.jpg 2560 x 1920 24bit Exif N 2312245
...
# Afficher tout (feh est dans le paquetage du même nom et est très
pratique pour visualiser un grand nombre de photos).
% feh --geometry 800x600 *
...
# Réduire le nombre de pixels (et donc la taille, mais aussi la
# qualité). convert est dans le paquetage imagemagick.
% convert -verbose -geometry 800x600 p1000049.jpg ~/tmp/gateau-aux-fraises.jpg
```

L'examen du journal (`/var/log/syslog`) nous montre bien `usbmount` détectant et montant la mémoire de l'appareil photo :

```
Mar 9 10:24:55 batilda usbmount[10965]: testing whether /dev/sdb1 is readable
Mar 9 10:24:55 batilda usbmount[10965]: /dev/sdb1 contains a filesystem or disklabel
Mar 9 10:24:56 batilda usbmount[10965]: /dev/sdb1 contains filesystem type vfat
Mar 9 10:24:56 batilda usbmount[10965]: mountpoint /media/usb0 is available for /dev/sdb1
Mar 9 10:24:56 batilda usbmount[10965]: executing command: mount -tvfat -osync,noexec,nodev,noatime,gid=bor
Mar 9 10:24:56 batilda usbmount[10965]: executing command:run-parts /etc/usbmount/mount.d
```

Le fait que l'`uid` et le `gid` soient le mien me permet de détruire des fichiers, pas seulement de les lire. Cela a été configuré dans `usbmount.conf` :

```
FS_MOUNTOPTIONS="-fstype=vfat,gid=bortzmeyer,uid=bortzmeyer"
```

Pour information, voici les messages complets du système, montrant la détection de l'appareil (un Panasonic DMC-TZ1) :

```
Mar 9 10:24:49 batilda kernel: usb 2-2: new full speed USB device using uhci_hcd and address 2
Mar 9 10:24:49 batilda kernel: usb 2-2: configuration #1 chosen from 1 choice
Mar 9 10:24:49 batilda kernel: Initializing USB Mass Storage driver...
Mar 9 10:24:49 batilda kernel: scsi2 : SCSI emulation for USB Mass Storage devices
Mar 9 10:24:49 batilda kernel: usbcore: registered new driver usb-storage
Mar 9 10:24:49 batilda kernel: USB Mass Storage support registered.
Mar 9 10:24:54 batilda kernel: Vendor: MATSHITA Model: DMC-TZ1 Rev: 0100
Mar 9 10:24:54 batilda kernel: Type: Direct-Access ANSI SCSI revision: 02
Mar 9 10:24:54 batilda kernel: SCSI device sdb: 245504 512-byte hdwr sectors (126 MB)
Mar 9 10:24:54 batilda kernel: sdb: Write Protect is off
Mar 9 10:24:54 batilda kernel: SCSI device sdb: 245504 512-byte hdwr sectors (126 MB)
Mar 9 10:24:54 batilda kernel: sdb: Write Protect is off
Mar 9 10:24:54 batilda kernel: sdb: sdb1
Mar 9 10:24:54 batilda kernel: sd 2:0:0:0: Attached scsi removable disk sdb
```

Une autre solution est d'utiliser `udev` et `automount` <<http://tldp.org/HOWTO/Automount.html>>. `automount` permet de monter automatiquement un système de fichiers lorsqu'on veut y accéder. Donc, un simple `cd /mnt/camera` va monter `/mnt/camera` s'il ne l'était pas. Pour cela, le fichier `/etc/auto.master` contient :

```
/mnt /etc/auto.misc --timeout=60 --ghost
```

et, à son tour, `auto.misc` contient :

```
camera    -fstype=vfat, sync, noatime, uid=stephane, gid=stephane    :/dev/camera
```

Mais vers quoi pointe `/dev/camera` alors que le périphérique Unix correspondant à l'appareil photo peut avoir des numéros différents à chaque fois ? C'est là qu'`udev` rentre en jeu, en garantissant que `/dev/camera` pointera toujours vers l'appareil photo. Sa configuration en `/etc/udev/rules.d/camera.rules` contient :

```
# Mon appareil photo Pentax Optio 50L
#BUS="scsi", SYSFS{vendor}="PENTAX", NAME="camera"
# Mon appareil photo Digital 4.1 Megapixel
BUS="scsi", SYSFS{vendor}="Sunplus", KERNEL="sdb1", NAME="camera"
# Mon appareil photo Panasonic TZ1
BUS="scsi", SYSFS{vendor}="MATSHITA", KERNEL="sdb1", NAME="camera"
```

Ansì, le même nom est toujours donné lorsqu'un appareil photo d'un des vendeurs indiqués est connecté.