

Using a Minolta Dimage 5 camera under Linux



by Guido Socher ([homepage](#))

About the author:

Guido loves Linux not only because it is fun to discover the great possibilities of this systems but also because of the people involved in its design.



Abstract:

Recently I decided to buy a new digital camera. As usual I looked for cameras supported by Linux first. I never buy any hardware unless it is supported by Linux. I generally recommend that strategy to everybody, even if they still have another operating system installed. It gives you more freedom and flexibility. You will not need to buy new hardware when one day you decide to work only with Linux. www.gphoto.org provided the answer: Yes, Minolta Dimage 5, one of the cameras that I had in mind, was fully supported.

I bought the Dimage 5 and wanted to get the images out off the camera into the computer but surprise, surprise: All Linux sites talking about Dimage 5 said it would work with Linux but I could not find a single word on how to get it working @!?!%

That was the moment I decided that I will write a short article about Linux and Minolta Dimage.

Introduction

The Dimage 5 camera works perfectly under Linux but to get there you need to configure the usb-storage system. Specifically you need to re-compile the kernel.

I will concentrate in this very short article only on the configuration and installation part. I will not

explain how to view, edit or archive digital images.

The idea

The Dimage 5 and Dimage 7 cameras use the USB mass storage system. That means they look from Linux point of view like a hard disk. You can copy, move, remove images with any file manager (e.g. konqueror) or with shell commands. In other words you do not need anything special to retrieve the images. You only need to convince Linux to "talk" to your camera.

Step by step

I have used Kernel 2.4.17 for this article other 2.4.x kernels will probably work as well but 2.2.x will not work since the usb mass storage system has not been ported back to 2.2.x.

Edit the file `unusual_devs.h` from `/usr/src/linux/drivers/usb/storage/` and add the following 2 entries:

```
UNUSUAL_DEV( 0x0686, 0x4008, 0x0001, 0x0001,
"Minolta",
"Dimage 5",
US_SC_SCSI, US_PR_BULK, NULL, US_FL_START_STOP ),
UNUSUAL_DEV( 0x0686, 0x4006, 0x0001, 0x0001,
"Minolta",
"Dimage 7",
US_SC_SCSI, US_PR_BULK, NULL, US_FL_START_STOP ),
```

That was the hardest part! The rest is straight forward. I informed the maintainer of the file about these entries. Hopefully some future kernel version will have those lines included.

Now recompile the kernel and make sure that you have selected the following options in addition to the things that you need normally for you hardware.

```
# General setup
CONFIG_HOTPLUG=y
```

```
# SCSI support
CONFIG_SCSI=y
CONFIG_BLK_DEV_SD=y
CONFIG_SD_EXTRA_DEVS=40
CONFIG_SR_EXTRA_DEVS=4
CONFIG_CHR_DEV_SG=m
CONFIG_SCSI_CONSTANTS=y
CONFIG_SCSI_LOGGING=y
```

```
# File systems
CONFIG_FAT_FS=m
CONFIG_VFAT_FS=m
```

```
# USB support
CONFIG_USB=y
CONFIG_USB_DEVICEFS=y
```

```
# USB Controllers
CONFIG_USB_UHCI=m
CONFIG_USB_UHCI_ALT=m
CONFIG_USB_OHCI=m
```

```
# USB Device Class drivers
CONFIG_USB_STORAGE=m
CONFIG_USB_STORAGE_DATAFAB=y
CONFIG_USB_STORAGE_DPCM=y
CONFIG_USB_STORAGE_SDDR09=y
CONFIG_USB_STORAGE_JUMPSHOT=y
```

You can check for these options in the file `/usr/src/linux/.config`. I don't think that you really need the `CONFIG_USB_STORAGE_DATAFAB`, `CONFIG_USB_STORAGE_DPCM`, `CONFIG_USB_STORAGE_SDDR09` and `CONFIG_USB_STORAGE_JUMPSHOT` options but these are some flash card reader devices and it might be good to have them when you own a camera.

SCSI support is needed because the `usb-storage` will appear to the user as a SCSI disk.

Now compile and install the kernel. There is a lot of documentation on how to compile a kernel (Kernel-HOWTO from www.linuxdoc.org or this [Linuxfocus](#) article) I will therefore not explain any details. Basically you need to do the following:

```
#create links in /usr/include for:
scsi -> ../src/linux/include/scsi
asm -> ../src/linux/include/asm-i386
linux -> ../src/linux/include/linux
```

```
# configure:
make xconfig
# compile:
make dep
make clean
make bzImage
make modules
make modules_install
```

```
cp /usr/src/linux/arch/i386/boot/zImage /boot/vmlinuz-2.4.17
cp /usr/src/linux/System.map /boot/System.map-2.4.17
cp /boot/System.map-2.4.17 /boot/System.map
cp /usr/src/linux/.config /boot/Config-2.4.17
```

```
#edit /etc/lilo.conf
```

```
# run lilo (or grub if you use it)
/sbin/lilo
```

```
#reboot the computer
```

Erratum:

Thanks to Seth W. Klein for pointing out that it is no longer best to make symlinks from /usr/include. For a detailed explanation see this mail from Linus.

Connecting the camera

Connect the camera and setup the usb connection. This is described in the manual that comes with the camera.

Check with the command /sbin/lsmmod that you have the modules usb-storage and usb-uhci (or usb-ohci) loaded. Otherwise load them with

```
modprobe usb-uhci
modprobe usb-storage
```

Open the file /proc/bus/usb/devices and check that you see an entry for the camera:

```
T: Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#= 3 Spd=12 MxCh= 0
D: Ver= 1.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
P: Vendor=0686 ProdID=4008 Rev= 0.01
S: Manufacturer=MINOLTA DIMAGE CAMERA
S: Product=DIMAGE CAMERA
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I: If#= 0 Alt= 0 #EPs= 2 Cls=08(stor.) Sub=06 Prot=50 Driver=usb-storage
E: Ad=03(O) Atr=02(Bulk) MxPS= 16 Ivl= 0ms
```

The usb proc file system with the above devices file is usually automatically mounted but if this is not the case then mount it manually with the following command:

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

For those who are curious what the numbers on the driver line in the devices file mean a short explanation. The camera tells the computer what kind of protocol it speaks:

```
Cls=08(stor.) -> Class usb storage
Sub=06 -> USB sub class 06= transparent SCSI =US_SC_SCSI (see protocol.h from kernel sources)
Prot=50 -> protocol usb bulk transfer only =US_PR_BULK (see transport.h from kernel sources)
```

The camera will appear as SCSI disk. If you have already some other SCSI disk then it will be /dev/sdb1 otherwise it is /dev/sda1. With fdisk you can check where the disk is:

```
fdisk -l /dev/sda
```

```
Disk /dev/sda: 4 heads, 32 sectors, 244 cylinders  
Units = cylinders of 128 * 512 bytes
```

```
Device Boot Start End Blocks Id System  
/dev/sda1 * 1 245 15664 1 FAT12
```

Add the following line to `/etc/fstab`:

```
/dev/sda1 /mnt/camera0 vfat rw,noauto,user 0 0
```

Create the directory `/mnt/camera0` and make it world writable:

```
mkdir /mnt/camera0  
chmod 777 /mnt/camera0
```

```
chmod 666 /dev/sda1
```

This setup will allow any user to mount and unmount the camera without switching to root. (Note that `chmod 777 /mnt/camera0` and `chmod 666 /dev/sda1` are not needed if your mount command has the `s`-bit for root set, file permissions of `/sbin/mount -rwsr-xr-x`, if the `s`-bit is set then the `user` option in `/etc/fstab` is sufficient.)

Getting the pictures

We are ready with setup and configuration. Now type:

```
mount /mnt/camera0
```

and get your pictures from the directory tree below `/mnt/camera0`!

A word of warning: Never disconnect the camera without unmounting it first (`umount /mnt/camera0`). It can lockup your computer.

I have written a small script to automatically retrieve all images from the CompactFlash card on the camera. This script also automatically mounts and unmounts the camera. That way you will never unplug the camera and forget to unmount it. I called the script `cfimageget` (download).

Save this file as `cfimageget` and make it executable with `chmod 755 cfimageget`. To copy all images to the current directory (".") simply run

```
cfimageget .
```

About the camera

Perhaps you intend to buy a Minolta Dimage and you are interested in my experience with this camera?

Generally I can say that this is a very good camera and in many aspects it has exceeded my expectations. It interoperates perfectly with Linux once you have gone through the configuration and Kernel re-compilation procedure.

You can get technical specifications about the camera at www.dimage.minolta.com.

The camera has, like many digital cameras, one big problem. It eats batteries. I did some measurements: With only the electronic view finder on it needs about 0.6 Ampere at 6 Volts. When it is about to autofocus and with the little monitor on it consumes in average 1 Ampere. This is 1 Ampere average! With a peak current consumption of up to 3 Ampere. With 4 little AA NiMh batteries (1600 mAh) the camera will operate for about 1 to 1.5 hours if you keep it on and use it. Minolta claims that you can shoot about 200 photos with one set of batteries. I personally think that you have to shoot these 200 photos rather quickly otherwise you will run out of battery. This is because it does not make a big difference if you actually take a picture or not. What counts is how long the camera is on.

References

- The linux usb pages linux-usb.org
- The linux hotplug page linux-hotplug.sourceforge.net
hotplug is a rather complicated set of shell scripts to automatically load modules such as usb-storage. I prefer a simple modprobe entry in /etc/rc.d/rc.local (Redhat/Mandrake) but it is up to you.
- A selection of image viewer programs:
gimp (www.gimp.org),
konqueror (www.kde.org),
gphoto (www.gphoto.org) You do not need the download function of gphoto but you can use it to browse any directory and generate html index pages,
gqview (gqview.sourceforge.net),
xv (www.trilon.com/xv/xv.html) This program did already exist when many computers had only text based desktops but it is still good.
- Casio cameras and usb-storage: www.harald-schreiber.de

Webpages maintained by the LinuxFocus Editor team © Guido Socher "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Translation information: en --> -- : Guido Socher (homepage)
--	---