

## Wie man die Dimage5 von Minolta unter Linux zum Einsatz bringen kann



by Guido Socher ([homepage](#))

### *About the author:*

Guido liebt sein Linux nicht nur auf Grund der unbegrenzten Möglichkeiten, die es einem bietet, sondern auch gerade wegen der Leute, die sich mit dem System beschäftigen.



### *Abstract:*

Kürzlich habe ich mich dazu entschlossen, eine neue Digitalkamera zu kaufen. Wie üblich habe ich zuerst geschaut, ob die Kameras unter Linux unterstützt werden. Ich kaufe niemals eine Hardware, wenn diese nicht unter Linux zu betreiben ist. Generell empfehle ich diese Strategie jedem, auch wenn man ein anderes Betriebssystem im Einsatz hat. Es verleiht einem mehr Freiheit und Flexibilität. Man ist später nicht gezwungen, sich neue Hardware anzuschaffen, wenn man auf Linux umsteigen will. [www.gphoto.org](http://www.gphoto.org) hatte dann auch die Antwort parat: Die Dimage 5 von Minolta, eines der Geräte, das ich mir vorher angeschaut hatte, wird von Linux unterstützt.

Ich habe mir also diese Kamera gekauft, als ich aber die geschossenen Bilder von der Kamera herunterladen wollte, kam die grosse Überraschung: Alle Linux Seiten sprachen zwar davon, dass dieses unter Linux funktionieren würde, aber ich fand nicht eine Zeile Text, die auch beschrieb, wie das gehen sollte.

Das war der Moment, als ich beschloss, diesen kurzen Artikel über Linux und die Minolta Dimage 5 zu schreiben.

---

## Einleitung

Die Dimage 5 arbeitet perfekt unter Linux, aber, um dorthin zu gelangen, muss man das USB-Storage System richtig konfigurieren und zumeist muss man auch den Kernel entsprechend neu kompilieren.

Ich werde mich in diesem kurzen Artikel auf den Konfigurations- und Installationspart konzentrieren. Ich

werde nicht näher auf Themen wie das Editieren, Anschauen oder Archivieren von digitalen Aufnahmen eingehen.

## Die Idee

Die Dimage 5 und Dimage 7 Kameras benutzen USB Speichermedien. Das bedeutet, dass sie von der Linux Seite aus betrachtet wie eine Festplatte zu benutzen sind. Man kann die Bilder mit jedem Filemanager wie z.B. konqueror kopieren, löschen oder auch verschieben. Oder man erledigt das Ganze mit den entsprechenden Befehlen der Shell. Mit anderen Worten, man muss nicht erst noch irgendwelche speziellen Dinge tun. Einzig seinem Linux muss man beibringen, auch zu der Kamera zu sprechen.

## Schritt für Schritt

Ich habe den Kernel 2.4.17 im Einsatz, aber andere Kernel aus der 2.4er Reihe würden es auch tun. Einzig die 2.2er Kernel scheiden aus, da das USB Storage-System für diese Reihe nicht implementiert wurde.

Öffnen Sie die Datei `unusual_devs.h` im Verzeichnis `/usr/src/linux/drivers/usb/storage/` und fügen Sie die beiden folgenden Einträge hinzu:

```
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 ),
```

Dies war der schlimmste Teil! Der Rest ist schnell erledigt. Ich habe den Maintainer dieser Datei über meine Einträge informiert. Es steht zu hoffen, dass diese Zeilen in Zukunft schon im Kernel implementiert sein werden.

Nun rekompilieren Sie den Kernel. Stellen Sie dabei sicher, dass Sie die folgenden Einstellungen aktiviert haben zusätzlich zu den sonstigen Optionen, die Sie normalerweise für den Betrieb ihres Rechners benötigen.

```
# 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
```

Sie finden diese Optionen in der Datei /usr/src/linux/.config. Ich bin nicht sicher, ob Sie die CONFIG\_USB\_STORAGE\_DATAFAB, CONFIG\_USB\_STORAGE\_DPCM, CONFIG\_USB\_STORAGE\_SDDR09 und CONFIG\_USB\_STORAGE\_JUMPSHOT Optionen wirklich benötigen, diese stellen Devices für Flash-Card-Leser dar, aber es könnte eine lohnende Sache sein, wenn Sie eine solche Kamera besitzen.

Der SCSI Support wird benötigt, da sich das USB Storage System dem User gegenüber als eine SCSI Festplatte identifiziert.

Kompilieren und installieren Sie nun den neuen Kernel. Es gibt eine Menge an Dokumentation, wie man dieses tut z.B. im Kernel-HOWTO auf [www.linuxdoc.org](http://www.linuxdoc.org) oder [hier](#) bei uns auf Linuxfocus. Von daher werde ich nicht näher darauf eingehen. Generell müssen Sie folgendes tun:

```
#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:

Vielen Dank an Seth W. Klein für seinen Hinweis, daß es nicht gut ist symlinks aus /usr/include zu setzen. Eine detaillierte Erklärung findet sich in [dieser Mail von Linus](#).

## Anschluss der Kamera

Schliessen Sie die Kamera per USB Kabel an den Rechner an, wie dieses geht wird im Handbuch der Dimage genau erklärt.

Überprüfen Sie nun mit dem Befehl /sbin/lsmmod, ob die Module usb-storage und usb-uhci (oder usb-ohci) geladen sind. Falls das nicht der Fall ist, können Sie sie so nachladen:

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

Öffnen Sie die Datei /proc/bus/usb/devices und prüfen Sie, ob ein Eintrag für die Kamera vorhanden ist:

```
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
```

Das USB proc Dateisystem mit den oben genannten Einträgen wird normalerweise automatisch gemounted, falls dies aber bei Ihnen nicht der Fall sein sollte, dann können Sie es mit dem folgenden Befehl nachholen:

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

Für diejenigen, die sich fragen, was die ganzen Zahlen in der Treiberzeile bedeuten, folgt hier die Auflösung: Die Kamera beschreibt dem Computer welche Protokolle sie beherrscht.

```
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)
```

Die Kamera erscheint als eine SCSI Festplatte. Wenn Sie bereits eine SCSI Festplatte angeschlossen haben, dann identifiziert sich die Kamera mit /dev/sdb1 ansonsten ist es /dev/sda1. Mit fdisk können Sie herausfinden, wo sich ihre "Festplatte" befindet:

```
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

Fügen Sie die folgende Zeile in Ihre /etc/fstab:

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

Erzeugen Sie ein Verzeichnis /mnt/camera0 und setzen Sie die Zugriffsrechte so, dass jeder dort schreiben kann:

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

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

Diese Konfiguration sorgt dafür, dass jeder User auf dem Rechner die Kamera mounten und unmounten kann, ohne vorher zum root Account wechseln zu müssen.

## Abholung der Bilder

Wir sind nun fertig mit dem eigentlichen Setup und der nötigen Konfiguration. Tippen Sie bitte jetzt

```
mount /mnt/camera0
```

und holen dann die geschossenen Bilder aus dem Verzeichnisbaum unterhalb von /mnt/camera0.

Ein Wort der Warnung: Trennen Sie niemals die Kamera vom Computer, ohne vorher das Gerät zu unmounten (umount /mnt/camera0). Es könnte sonst sein, dass sich der ganze Rechner dabei aufhängt.

Ich habe ein kleines Script geschrieben, das automatisch alle Bilder von der CompactFlash Karte holt. Das Script mounted und unmounted dabei auch gleich die Kamera. Auf diese Weise laufen Sie nicht Gefahr, einmal die Kamera vom Rechner zu trennen, ohne vorher das Device ungemounted zu haben. Ich habe das Script cfimageget ([download](#)) genannt.

Sichern Sie die Datei als cfimageget und machen es danach ausführbar mit

```
chmod 755 cfimageget.
```

Um alle Bilder in das aktuelle Verzeichniss zu kopieren, brauchen sie dann nur noch den Befehl einzugeben

```
cfimageget .
```

## Die Kamera selber..

Vielleicht sind Sie hier hergekommen, weil Sie sich für eine Minolta Dimage interessieren und sich für meine Erfahrungen mit der Kamera interessieren.

Generell würde ich sagen, dass die Kamera wirklich gut ist und in vielen Belangen hat sie meine Erwartungen übertroffen, nachdem man sie mit den beschriebenen Mitteln zur Zusammenarbeit mit Linux überreden konnte.

Die technischen Spezifikationen für die Kamera finden Sie auf [www.dimage.minolta.com](http://www.dimage.minolta.com).

Die Kamera leidet, wie viele andere Digitalkameras auch, allerdings unter einem grossen Problem: Sie frisst Batterien. Ich habe dazu ein paar Messungen vorgenommen: Mit eingeschaltetem elektronischen Sucher verbraucht die Kamera ungefähr 0.6 Ampere bei 6 Volt. Wenn sich allerdings der Autofokus einschaltet und der kleine Monitor auch eingeschaltet ist, dann steigt der Verbrauch im Mittel auf 1 Ampere, bei Spitzenlasten von bis zu 3 Ampere. Bei 4 kleinen AA NiMh Akkus mit 1600 mAh kann die Kamera nur knapp eine bis eineinhalb Stunden arbeiten, ehe die Akkus getauscht werden müssen. Minolta gibt an, dass man mit einem Batterieset ungefähr 200 Fotos schießen kann. Ich persönlich denke allerdings, dass man diese Bilder sehr schnell schießen muss. Dabei ist es nicht wichtig, ob man ein Bild schießt oder nicht, wichtig ist der Zeitraum, über den die Kamera aktiviert ist.

## Referenzen

- Die Linux USB Seite [linux-usb.org](http://linux-usb.org)
- Die Linux Hotplug Seite [linux-hotplug.sourceforge.net](http://linux-hotplug.sourceforge.net)  
Hotplug ist allerdings ein reichlich kompliziertes Set von Shell Scripten, um automatisch solche Module wie usb-storage zu laden. Ich persönlich befürworte eine einfache Lösung wie ein modprobe Eintrag in /etc/rc.d/rc.local (Redhat/Mandrake), aber das liegt ganz an Ihnen.
- Eine Auswahl von Bildbetrachtungsprogrammen:  
gimp ([www.gimp.org](http://www.gimp.org)),  
konqueror ([www.kde.org](http://www.kde.org)),  
gphoto ([www.gphoto.org](http://www.gphoto.org)) Man braucht hierbei nicht die Download-Funktion von gphoto einsetzen, aber man kann es dazu gebrauchen durch die Verzeichnisse zu browsen oder man benützt den Html-Generator,  
gqview ([gqview.sourceforge.net](http://gqview.sourceforge.net)),  
xv ([www.trilon.com/xv/xv.html](http://www.trilon.com/xv/xv.html)) Dieses Programm existierte bereits, als viele Computer nur eine Textkonsole hatten. Es ist allerdings noch immer gut zu gebrauchen.
- Informationen zu Casio Kameras und usb-storage: [www.harald-schreiber.de](http://www.harald-schreiber.de)

---

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Guido Socher "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a></p>	<p>Translation information: en --&gt; -- : Guido Socher (<a href="#">homepage</a>) en --&gt; de: Marcus Franke &lt;Marcus.Franke/at/gmx.net&gt;</p>
--	---