

Een Minolta Dimage 5 camera gebruiken onder Linux



door Guido Socher ([homepage](#))

Over de auteur:

Guido houdt van Linux. Niet alleen omdat het interessant is te leren hoe een besturingssysteem werkt, maar ook vanwege de mensen die betrokken zijn bij het ontwerp ervan.

Vertaald naar het Nederlands door:
Guus Snijders
<ghs/at/linuxfocus.org>



Kort:

Niet lang geleden besloot ik een nieuwe digitale camera aan te schaffen. Zoals altijd keek ik eerst welke camera's er ondersteund werden door Linux. Ik koop namelijk nooit hardware die niet ondersteund wordt door Linux. Deze strategie raad ik iedereen aan, zelfs als je nog een ander besturingssysteem gebruikt. Het levert je meer vrijheid en flexibiliteit op. Je hoeft geen nieuwe hardware aan te schaffen wanneer je op een dag besluit met Linux te gaan werken. www.gphoto.org leverde het antwoord: Ja, de Minolta Dimage 5, een van de camera's die ik in gedachten had, werd volledig ondersteund.

Ik kocht de Dimage 5 en wilde de foto's van de camera overbrengen naar de computer maar, verassing: alle Linux sites die de Dimage 5 noemden, zeiden dat het zou werken met Linux, maar ik kon geen informatie vinden hoe je het aan de gang moest krijgen @!?!%

Op dat moment besloot ik een kort artikel te schrijven over Linux en de Minolta Dimage.

Introductie

De Dimage 5 camera werkt perfect onder Linux, maar om dit voor elkaar te krijgen, zul je het usb mass storage systeem moeten configureren. Meer specifiek, je zult de kernel opnieuw moeten compileren.

In dit zeer korte artikel ga ik alleen in op de configuratie en de installatie. Het bekijken, bewerken en opslaan van de digitale plaatjes komt hier niet aan de orde.

Het idee

De Dimage 5 en Dimage 7 camera's gebruiken het USB massa-opslag systeem. Dat betekent dat ze vanuit Linux worden aangesproken als een harde schijf. Je kunt de foto's kopiëren, verplaatsen en verwijderen met een willekeurige bestands beheerder (bijv. konqueror) of met shellopdrachten. Met andere woorden, je hebt niks bijzonders nodig om de foto's te verkrijgen. Je hoeft alleen Linux ervan te overtuigen met je camera te "praten".

Stap voor stap

Voor dit artikel heb ik Kernel 2.4.17 gebruikt, met andere 2.4.x kernels zal het waarschijnlijk ook wel lukken, maar niet met 2.2.x, aangezien het 'USB mass storage system' niet geschikt gemaakt (geport) is voor 2.2.

Bewerk het bestand unusual.devs.h in /usr/src/linux/drivers/usb/storage/ en voeg de volgende regels toe:

```
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 ),
```

Dat was het moeilijkste gedeelte! De rest is recht toe, recht aan. Ik heb de ontwikkelaar van het bestand op de hoogte gesteld van deze wijzigingen. Hopelijk zullen deze in toekomstige kernels worden opgenomen.

Nu is het tijd om de kernel te hercompileren, en te controleren of de juiste opties zijn geselecteerd, in aanvulling op de dingen die normaal gesproken nodig hebt voor jouw 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
```

Je kunt deze opties nakijken in het bestand /usr/src/linux/.config. Ik denk niet dat de CONFIG_USB_STORAGE_DATAFAB, CONFIG_USB_STORAGE_DPCM, CONFIG_USB_STORAGE_SDDR09 en CONFIG_USB_STORAGE_JUMPSHOT opties echt noodzakelijk zijn, maar dit zijn een paar Flash kaartlezers en als je een camera hebt kunnen die wel eens handig blijken.

SCSI ondersteuning is nodig omdat de usb-storage zich zal voordoen als een SCSI schijf.

Nu kun je de kernel compileren en installeren. Er is veel documentatie te vinden over het compileren van een kernel (de Kernel-HOWTO van www.linuxdoc.org of dit LinuxFocus artikel). Ik zal dan ook niet ingaan op de details. Wat je in principe zou moeten doen is het volgende:

```
#maak links aan in /usr/include voor:
scsi -> ../src/linux/include/scsi
asm -> ../src/linux/include/asm-i386
linux -> ../src/linux/include/linux
```

```
# configureer:
make xconfig
# compileer:
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
```

```
#bewerk /etc/lilo.conf
```

```
# draai lilo (of grub als je die gebruikt)
/sbin/lilo
```

```
# herstart de computer
```

De camera verbinden

Sluit de camera aan en start een usb verbinding in. Dit staat beschreven in de handleiding van de camera.

Controleer met het commando `/sbin/lsmmod` of de modules `usb-storage` en `usb-uhci` (of `usb-ohci`) zijn geladen. Zo niet, kun je ze laden met

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

Open het bestand `/proc/bus/usb/devices` en zoek de ingang voor de 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
```

Het usb proc bestandssysteem met bovengenoemde apparaten word meestal automatisch geladen, maar als dit niet het geval is, kun je het mounten met het volgende commando:

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

Voor degenen die zich afvragen wat de nummers op de driver regel in het apparaat bestand betekenen, een korte uitleg. De camera vertelt de computer welk protocol het spreekt:

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

De camera zal verschijnen als zijnde een SCSI schijf. Als je al een andere SCSI schijf hebt, zal je camera beschikbaar zijn via `/dev/sdb1`, anders is het `/dev/sda1`. Met `fdisk` kun je kijken waar de schijf 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
```

Voeg de volgende regel toe aan /etc/fstab:

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

Maak de directory /mnt/camera0 aan en maak het schrijfbaar voor iedereen:

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

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

Met deze setup is het voor iedere gebruiker mogelijk de camera te mounten en te unmounten, zonder eerst root te worden.

De foto's verkrijgen

We zijn klaar met installeren en configureren. Typ nu in:

```
mount /mnt/camera0
```

en je foto's zijn gewoon beschikbaar /mnt/camera0!

Waarschuwing: verbreek de verbinding van je camera nooit zonder het eerst te unmounten (umount /mnt/camera0). Het kan je computer laten vastlopen.

Ik heb een klein scriptje geschreven om alle foto's op de CompactFlash kaart in de camera binnen te halen. Dit script zal ook automatisch de camera mounten en unmounten. Ik heb het script cfimageget (download) gedoopt.

Sla dit bestand op als cfimageget en maak het uitvoerbaar met
chmod 755 cfimageget. Om alle foto's te kopieëren naar de huidige directory ("."), typ:

```
cfimageget .
```

Over de camera

Ben je ook van plan een Minolta Dimage aan te schaffen en ben je geïnteresseerd in mijn ervaringen met deze camera?

Kort gezegd vind ik dit een hele goede camera die op veel vlakken mijn verwachtingen overtrof. Hij werkt perfect met Linux eenmaal de configuratie en de kernelcompilatie achter de rug zijn.

De technische specificaties zijn te vinden op www.dimage.minolta.com.

De camera heeft echter, net als veel digitale camera's, een groot nadeel. Hij vreet batterijen! Ik heb een paar metingen verricht: Met alleen de elektronische 'view finder' aan, verbruikt hij 0,6 Ampere bij 6 Volt. Met de autofocus en het kleine beeld ingeschakeld consumeert hij gemiddeld zo'n 1 Ampere. Dit is 1 Ampere gemiddeld, met een piek van 3 ampere. Met 4 kleine AA NiMh batterijen (1600 mAh) kun je de camera 1 tot 1,5 uur gebruiken als je hem ingeschakeld laat en ook effectief gebruikt. Minolta beweert dat je zo'n 200 foto's zou kunnen maken met 1 set batterijen. Ik zelf vind dat je die 200 foto's dan wel heel snel moet maken, omdat anders je batterijen al leeg zijn. Dit komt omdat het weinig uitmaakt of je nu een foto maakt of niet. Wat telt is hoe lang de camera aan staat.

Referenties

- De linux usb site linux-usb.org
- De linux hotplug pagina linux-hotplug.sourceforge.net
hotplug is een nogal gecompliceerde set shell scripts om automatisch modules te laden zoals usb-storage. Ik geef de er de voorkeur aan om een modprobe op te nemen in /etc/rc.d/rc.local (Redhat/Mandrake) maar dat is iets dat je zelf moet weten.
- Een paar foto viewers:
gimp (www.gimp.org),
konqueror (www.kde.org),
gphoto (www.gphoto.org) De download functie van gphoto heb je niet nodig, maar je kunt het gebruiken voor het bladeren door directories en het genereren van html index pagina's,
gqview (gqview.sourceforge.net),
xv (www.trilon.com/xv/xv.html) Dit programma bestond reeds toen veel computers alleen nog maar text gebaseerde desktops hadden, maar het is nog steeds goed.
- Casio cameras en usb-storage: www.harald-schreiber.de

Site onderhouden door het LinuxFocus editors team © Guido Socher "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Vertaling info: en --> -- : Guido Socher (homepage) en --> nl: Guus Snijders < ghs/at/linuxfocus.org >
---	---