

LINUX CONNECTIONS

Connecting to USB cameras



Say cheese! Nick Veitch coaxes images from the next generation of USB digital cameras.

Many of the new range of USB cameras on the market do not conform to the traditional model of serial camera communications. In the past these devices have used their own serial protocols and custom commands to communicate with a host over a serial link. Many of these cameras are already ably supported by *gphoto*.

However, a growing number of USB devices (but not all!) do not use traditional communications. Instead these devices take photographs and store them on a Flash card or Smartcard, which is treated solely as a storage device. This device is then simply represented on the USB bus as a mass storage device – the host computer communicates with the flash card as a storage device.

This is a great boon to the ability of Linux to support these devices. Their appearance on the Universal Serial Bus means the usb-storage module, which also deals with USB CD/CDRW devices, hard drives and other storage devices, can simply address the cameras contents as though it were accessing a hard-drive. No complicated protocols or command sets, you just need to mount the device and you can copy the pictures off it.

USB support

The first thing to do is check that your kernel supports USB. Many older distributions used patched kernels to provide some USB support, but it is certainly much improved in later releases from 2.4 onwards.

Recent distros like SuSE 7.2, Mandrake 7.2 and later will come with a kernel supporting mass-storage devices. As this is a basic class-level support, rather than a particular device driver (like some of the Kodak DC series) it is most likely it is enabled and working in your kernel if you have a modern distribution.

It is also likely that you have the `usb` service running. This monitors the usb ports for devices that are removed or become available. Obviously, this is handy for something like a camera. Check that the `usb` daemon is running on your system (you'll need to be logged in as root to do this):

```
ps x | grep usb
```

The next thing you need is the `usb-mass-storage` module. Any usb enabled kernel will probably already have this compiled as a module. If your `usb` service wasn't running you may have to install it yourself though.

Connect up the camera via a usb port and enter:

```
modprobe usb-storage
```

This will start the `usb-storage` module. To check that the

device is connected and that the module has loaded properly, you can check out the `/proc` filesystem entries:

```
less /proc/bus/usb/devices
```

```
less /proc/modules
```

The `usb-storage` module will map the device using pseudo-scsi support, so it will actually appear as a SCSI device on your Linux box. If you have no SCSI interface, the device is likely to be `sda`, otherwise it could be `sdb` or something else depending on which distro you use, and what SCSI devices you have.

Now you need to create a mountpoint for the camera-drive:

```
mkdir /mnt/camera
```

should be fine.

Now we can test that the mount will actually work, by using the `mount` command manually:

```
mount -t vfat /dev/sda1 /mnt/camera
```

The `mount` command here is using the `vfat` filesystem. This tends to be the filesystem universally used on USB cameras which are supported as drives (for some reason).

Once the device is mounted, you should be able to use it just like any other drive (although it will be a little bit slower). The camera may use a special directory for storing the picture files, which varies from camera to camera. Try: `ls /mnt/camera` to see any directories on the device.

Mountlist entry


When you are sure that the device works, you'll probably want to set up a mountlist entry for it in your `/etc/fstab` file. If you have `supermount` or `automount` you'll probably want to set it up using that. Using your favourite text editor, open the `/etc/fstab` file and add a line:

```
/mnt/camera /mnt/camera/ supermount
dev=/dev/sda1,fs=vfat,defaults,noauto 0 0
```

or you could manually mount and unmount the drive with:

```
/dev/sda1 /mnt/camera vfat defaults 0 0
```

In use

You can now happily use your camera, view the contents of the drive or copy all the files off. One word of warning though. Although USB is designed to be hotpluggable, some of the USB device drivers for Linux are not 100% there yet. If you unplug the camera in the middle of copying or moving files, you can expect some data to get lost. Don't do it! Make sure any copying is completely finished before unplugging the camera! 

Is my camera supported?

Because of the reluctance on the part of manufacturers to supply cameras, we can only say for certain that this works with the HP Photosmart 315 camera (which we just happened to have lying around the office). It should, in theory, work for a whole range of cameras from different manufacturers, who support their devices in a similar way. If your camera is supported in Windows/Mac OS as a USB drive, chances are it should work under Linux.