



Published on Free Software Magazine (<http://www.freesoftwaremagazine.com>)

Mobile devices in GNU/Linux and GNOME

Use your hand-held device under GNOME

By Mitch Meyran

You have a computer (a laptop or a desktop). Since it's a machine you use often and don't tinker with much, it probably runs Ubuntu Linux. Or, maybe, another distribution (like Mandriva 2008). If it doesn't run GNU/Linux, I hope you're at least using BSD. If not, stop reading right now!

You also have a brand new digital camera, or a shiny new MP3 player. And you feel the dread: are those pure consumer oriented pieces of hardware compatible with my machine? Will I have to pay the Microsoft tax (and the required hardware upgrades) to get all my photos from my last holidays, or to listen to Beethoven's fifth sung a capella by lazy llamas? Read on.

Software requirements

What exactly do you need? If you're running a recent version of Ubuntu Linux (Feisty Fawn or Gutsy Gibbon), you're all set. If you're not (like me), you'll need the following:

- Linux Kernel 2.6.22, with HAL and hald running (pretty much standard)
- dbus 1.0.2 (also standard).

No, these are not bad words. Let's look at what they actually represent.

HAL

HALD (Hardware Abstraction Layer Daemon) reports kernel events; when some pieces of hardware are connected, it notifies another daemon, udev, which creates a device in /dev. The device creation event is also picked up by FreeDesktop.org's dbus, which then follows a set of rules (typically written by the distributor of your flavour of GNU/Linux, stored into e.g. /etc/dbus-1 as a set of text files). Depending on your device identification, it will start a set of actions to make it usable.

dbus

dbus usually works with your desktop manager, and is a main component of the Freedesktop.org's specifications: as such, GNOME, KDE and Xfce (at least) will usually use it to perform some sort of user interface actions. On one hand, dbus notifies the window manager that a new device has appeared; dbus will check with its sets of rules, and start whatever action is set for this action: typically, a webcam creates an icon on the desktop linked to Ekiga, a CD/DVD disk triggers a file system browse (or if said drive is empty, ask if you want to burn it), etc.

Let's look at the matter at hand: both MP3 players (or more accurately, mass-storage based digital players) and digital cameras have one thing in common: they are ultimately storage devices, and their main use is to carry or transfer data (the fact that the camera at least can generate its own data is pretty much irrelevant here).

Both MP3 players and digital cameras have one thing in common: they are ultimately storage devices

As such, they will usually broadcast themselves as storage devices: the kernel will pick up on that and, through `hald` and `udev`, load the corresponding drive controller driver (if the drive contains some media, it will also create the corresponding devices).

Most devices use an USB connection (or, as is the case for the iPod, an alternative IEEE1394/Firewire); what usually happens then, is that the kernel will:

- load the `usb_storage` module (if it hasn't already);
- create a device for the driver (probably `/dev/sda` if you still have everything in IDE, kick up the last letter a notch for every SCSI-like drive you have installed if not)
- detect if the drive contains a media—in such case it will create partition devices (like `/dev/sda1`, `/dev/sda2...`): usually one, but some (like the iPod) contain at least two partitions.

One sure way to know is to look at `dmesg`'s last lines: you'll see the kernel detect the device, then configure it. Use `dmesg | tail` to keep it legible, and that's it. When these devices appear, `dbus` picks up on the kernel event and transmits it to the desktop manager: GNOME will then, by default, ask the system through `dbus` to create a mount point in `/media`, and mount the drive with default settings. Please note that GNOME relies upon the generic `mount` command to auto-detect the drive's settings: you may need to specify some details, although it is very unlikely: most hand-held media use FAT or FAT32, and these simplistic file systems don't have lots of options (it would be another matter if some used, say, NTFS), so `mount` usually gets them mounted nice and tight (you may still have trouble with files containing accented characters: look at `vfat`'s character encoding mount options if that happens).

For the duration of the device's connection, its mount settings are stored in `/etc/mtab`. You're not supposed to edit it, it merely reflects what media are mounted, like `fstab` does for manual mounts.

Some drives may appear directly as “whole”: my iRiver's storage partition for example, will directly show up as `/dev/sda` (and not `/dev/sda1`), which may make the rest of the system a bit confused if you retain an older set of `dbus` rules. Don't worry, in such cases a manual `mount` command will work very well:

```
mount -t vfat /dev/sda /media/iriver
```

The hardware

Now, I know this isn't supposed to be an article about hardware: for one thing, it's not “free”. Also, it's not because you use GNU/Linux that you're a tinkerer at heart. Be at ease though, I won't ask you to take your device apart (it would void its warranty!).

Oh, you would anyway? Read on, then.

Media players tested

iRiver iFP 190 series

The original, boxed product uses a proprietary storage schema (it may have changed since I bought it, maybe not): you need special software (Windows only) for it.

iRiver provides a UMS compatible firmware update which makes the player appear like a 128, 256 or 512 Mb USB key. It can playback WAV, WMA and MP3 formats (iRiver tried for Vorbis support, but they gave up due to hardware limitations; more recent models have Vorbis support though). Linux support is thus a UMS firmware update away; further updates can be done without Windows by simply copying the new firmware to the drive and ask it to update itself. It is powered by a single AA/LR6 standard battery, and connects through a slow USB 1.1 connection cable.

Most iRiver devices get regular, official, covered by warranty, firmware updates; look those up if you have one on the [iRiver website](#).

Apple iPod Mini 1st Gen

It comes with both an USB 2.0 and a Firewire connection (which can be used as the cable for the external charger), and contains a CompactFlash/Microdrive hard disk of 4 Gb. It requires specific software to upload the songs in a way that will have them recognized by the player. Replacing the non-standard internal battery and drive is tricky, but well documented on the Web – it costs much less than through Apple, for better results (switching the 4 Gb hard drive for a CompactFlash cartridge doubles battery autonomy; replacement batteries add an extra 25% battery life).

If your warranty isn't over yet, go through Apple for servicing though - as the above operation will void it.

I-BEAD 100 Digital Music Player

It contains soldered-in Flash RAM, and works as an USB key. Plug it in, browse it, use it. Biggest drawbacks: easily serviceable yet non-standard battery. USB 1.1 transfer speed. Not much else to say, your run of the mill media player.

Plug it in, browse it, use it. Not much else to say, your run of the mill media player

Cameras tested

Kodak EasyShare DX6340

This 3.1 Mpixels camera was well liked at the time it came out: on top of its optical zoom and microphone, its optical viewer allows its owner to switch off the external screen, making it last ten times longer on its twin AA/LR6 battery cells. It includes 16 Mb of internal RAM, as well as a standard SD slot. It connects to your PC with a USB cable. Since it started selling, a firmware update has appeared: apply it for a more stable experience.

Biggest drawback: it is not detected as an UMC drive, so you need to use an application that uses `libgphoto2`, like F-spot. But then, it can be excruciatingly slow and hang very often depending on your version of `libgphoto2`. If it does, update `libgphoto2` (version 2.4.0 works very, very, well).

Samsung Digimax 301

This 3.2 Mpixels camera is small, at the cost of an optical zoom. It is powered by a pair of AA/LR6 sized cells, and contains a SD card. It connects with a USB cable, is detected as an UMS drive (Full Speed USB, equivalent to USB 1.1) as well as a camera. GNOME knows its directory layouts, and will thus see it as a camera anyway.

HP Photosmart R707

This 5.1 Mpixels camera is a bit special as it is a dual mode camera; by default, it presents itself as a camera, but it can be configured to work as an USB drive. Note that this camera *must* use its latest firmware for safe operation: the battery charge manager doesn't work on firmware versions lower than 2.0.24, and must be updated (available on HP's website), otherwise you'll soon have a non-working battery, or worse, a blown-up one. Now, the new firmware is provided as a Windows .EXE file, but Wine can unpack it easily, and then you merely need to copy the resulting file to a freshly formatted SD card to proceed. Thus, Windows is not mandatory.

If you already fell victim to this problem, generic batteries model no. CS-NP60FU, 1050mAh, 3.7V work very well with it.

Back to the camera itself: not only can it work as a generic USB drive, gphoto2 knows how to handle it too. For some reason, it doesn't *always* work as an UMS device: switch it to 'camera' mode, and it'll happily work through gphoto2. So, it is pretty much trouble-free once its firmware is updated.

Apart from iPods, pretty much all devices benefit from a firmware update

As a general rule, always check if your device has an updated firmware, and use it if possible. Apart from iPods, pretty much all devices benefit from an update.

Playback your music

The easy way

Most models of MP3 players out there allow you to manage them easily: plug them in, put files in a format they support on them, unmount them, unplug them: done.

You can do that as you'd manually manage files, or use your favourite media player software to do that for you: Rhythmbox (for GNOME), Amarok (for KDE), Banshee, Audacious, XMMS (of course, in the latter case, usability is extremely limited: you'd be better off using a file manager)...

In some cases though, using these software is mandatory: iPods for example require their specially crafted database to be updated while a file is uploaded in order to play them back. If it isn't, then it won't play.

Of those, Rhythmbox (and Amarok) are recommended: they are geared towards handling connections of external devices, and are the default applications for GNOME and KDE.

Rhythmbox

Rhythmbox is an all-around music player based on gstreamer: it handles podcasts, webradios, local files in many formats and also external devices.

To use it, just plug your media player in, let Rhythmbox autodetect it, and then it's just a matter of dragging and dropping titles from your library (podcasts included) back and forth with your device. Once you're done, unmount the device, and that's it.

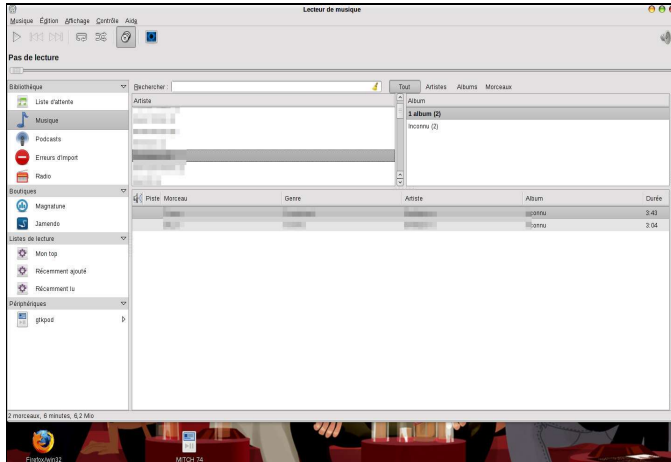


Figure 1: Rhythmbox right after importing my music collection

Rhythmbox's main window is divided into two main zones: the left task-bar conditions what you do with the application. At the top, the library is divided into the playlist (you set up what you actually listen), your local music, then podcasts, then error notifications (failed import, unrecognised file format, loss of sync... In my case, it couldn't playback WMA files, and just couldn't deal with Amiga modules) and web radios.

Following those, you have shops (here, Magnatunes and Jamendo), then your playlists and—if you have it connected—your mobile media player (here, it detected my iPod as `gtkpod`, but you can rename it).

Rhythmbox is an all-around music player based on gstreamer: it handles podcasts, webradios, local files in many formats and also external devices

Depending on the functionality you choose, the right-hand part of the window can have up to four subdivisions. The most complex one is for `Music`: it allows you to sort your music collection by author, album and song title. Upon importing music, Rhythmbox parses the file's tag (ID3 for MP3 files, embedded comments for Ogg, etc.) and sorts them into their relevant categories. It also allows fast sorting of your files, in case there are discrepancies in your ID3 tags.

Right beneath the program's main window in Figure 1, you can see the icon for the mounted iPod.

The hard way

To do it the "hard way, just copy your file to the player. You'll need to ensure by yourself that this model can handle the file you feed it, though:

- many small players can't handle high bit-rates: 192 Kbps is very often the upper limit, sometimes anything past 160 Kbps is too much. Remember the MPEG1-2 Layer 3 specifications don't cover

bit-rates above 320 Kbps;

- Some players can't handle variable bit-rates: while they are supposed to support variable bit-rate as part of the format (as VBR/ABR is nothing more than stretching the format specifications a bit), it is sometimes quite tricky to implement reliably on small pieces of hardware;
- MP3 is the most used format (only some old Sony devices don't understand it); however, some players can play back WMA, AAC and some can playback AC3, Vorbis and FLAC. But not all.

This method doesn't work with players that require a special way to access them, like all Apple iPod models.

You'll need to ensure by yourself that this model can handle the file you feed it

The Geek way

Some of these players are far more powerful than their manufacturers let you believe: several projects exist to make them run more than just a playback engine.

For you iPod owners, there is iPod Linux: take the μ Linux kernel, and adapt it to the iPod. You can then run a bunch of bona fide GNU/Linux applications (including a music player) on your iPod. Note that this way allows you to playback file formats the original firmware can't. It has some drawbacks though: as it is a full operating system, it is quite demanding on the hardware, and will suck the battery dry faster than the original firmware.

It doesn't work on the most recent models either, due to an encrypted boot sector. Older models may run fine, but I'll advise you to stop updating your iPod's firmware: Apple keeps making things more difficult, and won't release iTunes for Linux any time soon.

Another way is to use RockBox: that one is a real replacement firmware for many hand-held music players (not just the iPod). As such, it is more fit for everyday use than iPod Linux, and it comes with several of its advantages: much better customization of the hardware, several applications have been ported to this platform, to the point it can be considered a small computer. Yes, Doom runs on my iPod mini, with Rockbox; if you ever wondered what Doom would have looked like on a Nintendo Gameboy (the original), this is it.

Please note: if, for one reason or another, you are afraid of destroying the firmware, making the device unbootable, read as much as you can about restoring factory state from the Rockbox website, or from the manufacturer's - most iPods for example, include a hard-wired restore mode that will wipe the disk clean, connect with iTunes and ask it to reinitialize the device with the latest official firmware. Instructions, testimonies, How-To available at Rockbox's.

Doom runs on my iPod mini, which qualifies it as a computer

It is quite easy to install (detailed instructions for your player are available on the Rockbox website) and works very well. It is also constantly updated.

Both of these options will allow you to play back more formats (Vorbis, AC3, FLAC, etc.) than before, and you won't need an extra application to transfer files to your iPod any more.

I tried taking pictures of my iPod Mini 1st generation running Id Software's Doom shareware, but it was far too small to be useful.

Get to your pictures

Use the F-spot, Luke

When you connect your camera to your computer, GNOME asks you if you want to import your photos (if you always want to do that, check the option marked 'always do this action' the first time GNOME asks). It will then open F-spot, which downloads the thumbnail previews, and ask you what images you want to copy to your computer. At this stage, you can select the images you want to take, and what tag you will give them. The next stage, is the actual downloading of the images: be patient.

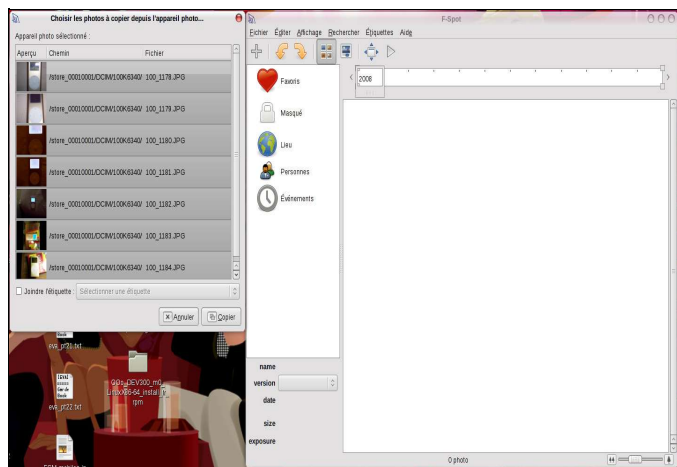


Figure 2: the import dialogue right along F-spot's main window. If you really strain your eyes on the full-size image, you'll see that the imported photos are of the iPod running Doom. Don't hurt yourself.

Next up, ordering your pictures. You can tag them on import, but also later on, then edit them through the GIMP (F-Spot doesn't provide editing functionalities past image orientation and colour adjustment, which are merely EXIF settings edition, not actual image editing)

Gphoto2 is both a library and a command line utility

Gphoto2 is both a library and a command line utility. The former is used by applications such as F-spot (for GNOME) or DigiKam (for KDE) to access some cameras that don't use UMS. It can access both serial port and USB connected cameras. The list of supported models can be seen with `gphoto2 --list-cameras`.

But the list is long, so you'd be better off filtering the list with `gphoto2 --list-cameras | grep <your camera model/brand>`.

Some are marked as TEST, others as EXPERIMENTAL. If yours is, too, try updating to a more recent version from <http://www.gphoto.org>, or through your distribution's updates (backports, upgrade your distribution...).

Switch to manual

However, you may want to proceed faster than, or you're just too 'old school' for F-spot: do not fret, as copying files from an UMS enabled camera is no more difficult than doing so from a USB key. You can open Nautilus, Thunar, Konqueror, Dolphin, or whatever, and just copy the images from one place (the camera) to another.

Mobile devices in GNU/Linux and GNOME

You can be even more rad, and do that through a terminal window with `cp` or `mv`, or with Midnight Commander, emacs, or... Whatever rocks your boat.

If, however, your camera doesn't work as an UMS device, things can get tricky. If your camera is supported by `gphoto2`, try the following commands:

```
gphoto2 --auto-detect
```

This may return:

```
$ gphoto2 --auto-detect
Model                               Port
-----
Kodak DX6340                        usb:
Kodak DX6340                        usb:001,012
```

Other useful commands are:

`-n`: gives the number of files on the device

`--storage-info`: describes what your camera has for storage (internal RAM, SD card...)

`-p <range>`: get images inside specified range

`-P`: get all files on the camera

Use `gphoto2 --help` for a more comprehensive list of options, or `man gphoto2` for the whole resource. `Gphoto2` also provides a shell, which may prove more useful than passing options all the time.

Gphoto2 also provides a shell, which may prove more useful than passing options all the time

If you just can't deal with `gphoto2` or it doesn't work for you, get an SD card reader, and re-read the part about manual file copy.

However, be extra careful: some cameras take liberties with the FAT partition format, and may panic if you feed them SD cards they didn't format themselves: thus, avoid formatting your SD card with your distribution's partition manager: use the camera's built-in functionality instead if you want to wipe out your card. Moreover, while extracting the SD card may allow you access your pictures with the camera powered off, I recommend you don't do that often: those cards (and their contacts inside the camera) are not made to be removed and plugged back in all the time.

Conclusion

I hope I tried enough different devices (sorry, I couldn't get my hands on a COM port connected camera) to give a comprehensive article. Still, a repeated word of warning: if your device doesn't work properly, first check its firmware (get the latest version, provided it doesn't add 'functionalities' like an encrypted boot sector), then your library versions (`hald`, `udev`, `dbus`, `gphoto2`, `libgpod`...), and before you despair, make sure that you don't have a moody USB controller.

Once you're up and running, enjoy—Free tools are powerful enough to fit any requirement you may have about them, and many times leave competition in the dust.

I mean, it's not Apple that turned my 10-hours battery life iPod into a 20 hours running game console...

Bibliography

I pretty much wrote this article through trial and error; still, you may want to look at the following resources:

- the `gphoto2` man page (`man gphoto2`) and [website](#)
- make your iPod your own with [Rockbox](#) and [iPodLinux](#)
- [dbus at Freedesktop.org](#)
- For `udev` and `hald`, have a look at their man pages (`man 8 udev`, `man 8 hald`).

Biography

[Mitch Meyran](#) (/user/17379" title="View user profile.): Mitch is a professional computer abuser. He started on this peculiar kink when he got a 8088-based PC, and soon found out that tinkering with computers is almost as fun as using them - provided the OS lets you do it.

Copyright information

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU General Public License, Version 2 or any later version published by the Free Software Foundation. A copy of the license is available at <http://www.gnu.org/copyleft/gpl.html>.

Source URL:

http://www.freesoftwaremagazine.com/articles/mobile_devices_linux_gnome
