

1 Overview

The nForce chipset contains a number of hardware devices that can be run under Linux. Along with a display processor, the chipset includes a network device (MAC), Audio hardware, an IDE controller, and an OHCI USB controller. Since much of this hardware uses industry standard interfaces, not all of them need custom drivers to run under Linux.

Please note: this is Beta documentation and device driver code. You may encounter bugs and problems with it. You may report bugs, problems, or answer questions using the email alias linux-nforce-bugs@nvidia.com.

2 Package Availability

This package can be found at the NVIDIA web site,

<http://www.nvidia.com/view.asp?PAGE=linux>

We strongly urge you to only use software obtained from this website or a trusted Linux distribution for your nForce hardware.

This package should work on most systems running a 2.4 kernel. It has been precompiled and tested on some of the more popular Linux distributions. Please note that the software is Beta quality, and hasn't been tested on all versions of the 2.4 kernel.

3 Licensing

The network driver provided by NVIDIA is subject to the NVIDIA software license; the license is available on the NVIDIA website, and is included in this package. By using this software, you are agreeing to the terms of the license. The rest of the software is provided under the GNU public license, which is also included in this package.

4 What the Package Contains

The packages contain the following items:

1. A README file
2. This Installation Guide
3. The GNU license
4. The NVIDIA license for the binary software
5. Makefiles with all, install, and uninstall targets

6. A patch for adding PCI ids to the i810 driver to make it recognize NVIDIA hardware
7. A patch for fixing a problem with the i810 driver when running NVIDIA hardware
8. A copy of the i810 driver in case it doesn't exist on the target machine
9. Source and binary files for building a network driver for NVIDIA hardware

5 What the Package Does

The packages will create and build a network driver and an audio driver into the appropriate locations for loadable modules. The audio driver is a patched version of the existing i810 driver; the network driver is from NVIDIA. The packages will also run `depmod` on the new module suite.

The binary packages will also update the modules configuration file, commenting out existing network, audio and usb entries, and adding entries for the new drivers. A backup file is created before this happens. If the binary package is uninstalled, it attempts to restore the original version of the file.

The source rpm and tar files don't try to modify the module configuration files. You should make sure the following lines are in the configuration file (it will be named `modules.conf` or `conf.modules`):

1. `alias eth0 mcpeth`
2. `alias sound-slot-1 i810_audio`
3. `alias usb-interface usb-ohci`

Neither the binary nor the source packages try to install the drivers into the currently running kernel. You can do this manually using `insmod` or `modprobe`. Upon reboot, the installation should happen automatically.

6 Installation

To install a binary rpms, you only need use `rpm` to install the package. This should install and configure the drivers. A source rpm requires building and installing the drivers yourself. The files in this package are organized into a build hierarchy to make this task easier.

6.1 Binary RPM Installation

In order to successfully install a binary RPM, the following must be true:

1. Your system is using Mandrake 8.1 or RedHat 7.2
2. You have not modified or installed a different kernel

If these points aren't true for your system, you should use the source rpm package, or the tarfile, then build and install the package by hand.

Become root and install using rpm:

```
example% cd <directory with the rpm file>
example% su
Password: *****
example# rpm -i nforce\textit{package name}-1.0-1.rpm
```

6.2 Source RPM Installation

The source rpm package contains source code, libraries, makefiles and documentation organized into three tar files. To install the source rpm, become root, install the rpm, build and install:

```
example% cd <directory with the rpm file>
example% su
Password: *****
example# rpm -i nforce<package name>-1.0-1.src.rpm
example# cd <directory with installed sources: maybe /usr/src/RPM/SOURCES>
example# tar -xvzf nvmaster.tgz
example# tar -xvzf nvaudio.tgz
example# tar -xvzf nvnet.tgz
example# cd nforce
example# make
example# make install
```

6.3 Tar File Installation

The source rpm package contains source code, libraries, makefiles and documentation organized into a single tar file.

To install the tar file, unpack it, build the sources and install them:

```
example% tar -xvzf nforce.tgz
example% cd nforce
example% make
example% su
Password: *****
example# make install
```

7 Removal

To remove the contents of one of the binary rpm files, use rpm to remove it:

```
example% cd <directory with rpm file>
example% su
Password: *****
example# rpm -e nforce<package name>-1.0-1.rpm
```

To remove the contents of one of the source rpm or tar file, go to the build directory, and run “make uninstall”:

```
example% cd <directory with unpacked build>
example% cd nforce
example% su
Password: *****
example# make uninstall
```