1 Contents

2 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. Many of the components of the nForce chipset use industry standard interfaces; thus, not all components require custom drivers to be used on Linux.

Bug reports and installation questions may be directed to:
linux-nforce-bugs@nvidia.com.

Please be sure read the TroubleShooting section first, and to report all relevant details, such as Linux Disribution, Kernel version, binary or source RPM, the fact that you checked the SBIOS configuration, the contents of /etc/modules.conf, and a listing of any error messges you find (you can run dmesg, or look at /var/log/messages as root). If you are having operational problems, please be as specific as possible, so we can try to reproduce the problem, and determine if we have a fix for it.

3 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.

4 Minimum Requirements

At the current time, the nForce drivers require a 2.4 series kernel.

5 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.

6 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 network driver 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 older i810 drivers 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

7 What the Package Does

This package will build a network driver and an audio driver, then place them in the appropriate locations for loadable kernel modules. For older distributions, a patched version of the existing i810 audio driver is included; latter versions of the i810 audio driver support the NVIDIA hardware directly; in those cases, no audio driver is provided. The network driver is from NVIDIA.

The binary packages will also update the modules configuration file, commenting out existing network, audio and usb entries, and add entries for the new drivers. If the package has been built for a newer Linux distribution, the modules configuration file will be set to use the i810 driver as the designated audio driver. A backup file is created before any changes are made to the configuration file. If the binary package is uninstalled, the package attempts to restore the original version of the file from this backup.

The 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 /etc/modules.conf on most current distributions):

- alias eth0 nvnet
- alias sound-slot-1 i810_audio
- alias usb-interface usb-ohci

Neither the binary nor the source packages will load the kernel drivers during the installation. You can do this manually using *insmod* or *modprobe.* (USAGE: 'insmod modulename' or 'modprobe modulename') Upon reboot, the kernel modules should insert themselves automatically.

8 Installation

Nvidia has provided precompiled binary RPMs for systems running Red Hat 7.2 and Mandrake 8.1. You should select the RPM appropriate to your current kernel version. You can check what kernel your machine is running by checking the output of 'uname -r'.

If you are using a different version of Red Hat or Mandrake, a modified kernel, or a different rpm-based distribution, you should use the SRPM to build an RPM appropriate for your system. If you are using the SuSE RPM, see NOTE: SuSE INSTALLATIONS, below.

If you are using a non-RPM based system, or if you prefer not to use RPM, you should download and install from the source tarballs.

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.

You might want to check the BIOS configuration on your system to ensure that the audio and networking devices will be detected. See the System BIOS Configuration description in the Troubleshooting section.

8.0.1 NOTE: SuSE INSTALLATIONS

Upgrading the SuSE Kernel: If you are using the SuSE Linux distribution, note that the binary RPM for SuSE only supports SuSE versions using the upgraded kernel. This note contains directions from SuSE on upgrading to the new kernel for SuSE versions 7.1, 7.2 and 7.3:

Get the upgraded kernel from one of the following sites (depending on SuSE distribution):

```
ftp://ftp.suse.com/pub/suse/i386/update/7.1/kernel/2.4.16 (for SuSE 7.1)
ftp://ftp.suse.com/pub/suse/i386/update/7.2/kernel/2.4.16 (for SuSE 7.2)
ftp://ftp.suse.com/pub/suse/i386/update/7.3/kernel/2.4.16 (for SuSE 7.3)
```

For update instructions, see SuSE's SBD article, located at

```
http://sdb.suse.de/en/sdb/html/ftpkernel.html
```

Alternative SuSE Installations: The following instructions should be used before proceeding with a source RPM or tar file installation. You will need to use one of these installation methods if you want to install nForce drivers without upgrading your kernel. Please note that these installation methods won't work for SuSE 7.1 or 7.2.

Install with YaST(2) the following d (development) series packages

- make
- gcc
- binutils
- glibc-devel
- kernel-source

Use the following commands to set up your development environment

```
example% cd <directory with the rpm file>
example% su
Password: ******
example# pushd /usr/src/linux
example# make cloneconfig
example# make dep
example# popd
```

You can now proceed with the tar file or source RPM installation steps below, omitting the steps to become root.

8.1 Binary RPM Installation

The binary installation procedure involves downloading the binary RPM file appropriate to your distribution and installing it. If you are using the SuSE distribution, see NOTE: SuSE INSTALLATIONS, above.

Become root and install using rpm:

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

8.2 Source RPM Installation

The Source RPM will create an RPM appropriate for your system. If you are using the SuSE RPM, see NOTE: SuSE INSTALLATIONS, above. To install:

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

At this point, the SRPM will be recompiled. When this command has completed, look for a line that says:

```
Wrote: /usr/src/{RPM root dir}/RPMS/i386/nforce{package name}-1.0-1.rpm
```

This will provide you with the name and location of the new RPM. You should then install this RPM per the instructions in the Binary RPM Installation section.

8.3 Tar File Installation

The tar file 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. If you are using the SuSE RPM, see NOTE: SuSE INSTALLATIONS, above:

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

9 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
```

10 TroubleShooting

10.1 System BIOS Configuration

Since the audio and network drivers work on the nForce chipset, these devices along with other motherboard devices are controlled by the System BIOS. If the installed drivers don't recognize hardware on your system, the problem may be your System BIOS's plug and play configuration. If your system BIOS expects the operating system to configure hardware devices, and your Linux kernel doesn't support ACPI-style configuration, you'll need to change your BIOS settings.

To view or change an Award-style system BIOS, reboot the machine, and press the Delete key. When you get a configuration screen, select "PnP/PCI Configurations" in that screen if "PNP OS Installed []" Has "Yes" choosen, change the selection to "No".

If your BIOS is Phoenix-style, use the F2 key instead of Delete, and scroll through the menus to find OS Type, and chose "Other". The exact details of System BIOS configuration vary with BIOS vendor, so the screens may not be exactly the same.

Warning: You should write down the original configuration of the BIOS before changing it. Changing your system BIOS configuration may adversely affect the operation of the system, and even make it unbootable.