



Digi RealPort Driver Package for Linux

Version 1.9-40

RPM Part Number: 40002086_AA

INTRODUCTION

Digi RealPort Linux is a driver package for Digi's Ethernet-based Serial Products.

It is assumed that TCP/IP is running on the system and is properly configured.

It is also useful, though not necessary, for the Wish package to be installed, or some other TCL interpreter, in order to use the RealPort Manager Configuration tool.

It is currently supported on the following hardware platforms:

- Standard i386/i486 and Pentium PC (x86 32bit)
- x86 64bit

The following Linux distributions are supported

- Red Hat Enterprise Linux 2.1, 3, 4, 5.x, 6.x, 7.x, 8.x
- Red Hat Linux 7.2, 7.3, 8, 9
- Fedora Core 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 17, 18
- Mandriva Linux 8.2, 9.0, 9.1, 9.2, 10.0, 10.1, 2006, 2007.1, 2008
- SuSE 8.1, 8.2, 9.0, 9.1, 9.2, 9.3, 10.0, 10.1, 10.2, 11.1, 11.2, 15
- OpenSuSE 42.3, 15.1
- Debian 3.0.r1, 3.1, 4, 6, 9, 10
- Ubuntu 7.04, 7.10, 8.04, 8.10, 12.04, 14.04, 16.04, 18.04, 19.10

NOTE: Because of the rapid rate of releases from each respective Vendor, the tested/supported list above quickly becomes out of date.

This driver package has been tested and verified working for kernels up to and including version 2.4.37, 2.6.38, and on various 3.X, 4.X and 5.X kernels up through 5.5.

It is anticipated that this driver will work with Vendor kernel releases newer than the above but cannot be guaranteed because each respective Vendor can, and does, add their own various changes/patches to the stock kernel.org kernel.

These additional Vendor patches to the stock kernel.org Linux kernel can cause unforeseen incompatibilities with this driver.

Please reference the following number(s) when searching the Digi International web site (www.digi.com) for the latest software package: 40002086

SUPPORTED PRODUCTS

- Digi Connect Family
- Digi ConnectPort Family
- Digi One Family
- Digi CM Family
- Digi Passport Family
- Digi PortServer TS Family
- Digi RealPort is also supported on other Digi products, for a complete list of supported products, refer to the Digi website: www.digi.com/RealPort.

SUPPORTED LINUX KERNELS

- 2.4.x (UP and SMP)
- 2.6.x (UP and SMP)
- 3.x.x (UP and SMP)
- 4.x.x (UP and SMP)
- 5.x.x (UP and SMP)

Tested Linux Distributions

- Red Hat Enterprise Linux 2.1, 3, 4, 5.x, 6.x, 7.x, 8.x
- Red Hat Linux 7.2, 7.3, 8, 9
- Fedora Core 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17
- Mandriva Linux 8.2, 9.0, 9.1, 9.2, 10.0, 10.1, 2006, 2007.1, 2008
- SuSE 8.1, 8.2, 9.0, 9.1, 9.2, 9.3, 10.0, 10.1, 10.2, 11.1, 11.2
- Debian 3.0.r1, 3.1, 4, 6.0, 10
- Ubuntu 7.04, 7.10, 8.04, 8.10, 12.04, 18.04, 19.10

TECHNICAL SUPPORT

Get the help you need via our Technical Support team and online resources. Digi offers multiple support levels and professional services to meet your needs. All Digi customers have access to product documentation, firmware, drivers, knowledge base, and peer-to-peer support forums. Visit us at <https://www.digi.com/support> to find out more.

CHANGE LOG

VERSION 1.9-40

ENHANCEMENTS

- Add support for 5.X kernels (tested up to 5.5) (RP-142)
- Fix support for gcc versions >4 (RP-142)
- Update the in-source openssl package to version 1.1.1d. (This is only used if openssl is not already installed on the Linux host when the driver is installed.) (RP-147, RP-151)

- dgrp_gui was originally written to run under X. Enhance it to also work under Wayland. See notes under "KNOWN LIMITATIONS" regarding a peculiarity. (RP-153)
- Deprecate support for obsolete devices (RP-146)
- Add Debian Source package release format. (RP-149)

BUG FIXES

- Fix problem with rpm builds on RHEL 7. (RP-126)

KNOWN LIMITATIONS

- To build the dgrp driver you must have first loaded the matching kernel source tree for your target kernel.
- To verify that you have matching kernel source, run `"/bin/uname -r"` and check to make sure the `/usr/src` directory has the corresponding `linux-<version>` directory.
- If you are running a prebuilt kernel from a packaged distribution, this typically involves loading the kernel sources from your distribution CD to your system, then using the following commands to do the following:

1. `cd /usr/src/linux-<your_version_of_the_kernel>`
2. `make mrproper` Clean up any old version files.
3. `make oldconfig` Make a configuration file to match your running kernel.
4. `make dep` Create the dependency and version files.

Now you can load and build the dgrp driver package.

- Occasionally, a Linux Vendor will ship a kernel that simply cannot be autodetected to add the various changes that might be required for that specific kernel. Because of this, there is an option that can be used for both the srpm and tgz to tell the driver package exactly what distribution you have.

Currently, the only needed and recognized option is for Red Hat AS/ES/WS 3 and Fedora

The flags are:

`REDHAT_AS_3 / REDHAT_ES_3 / REDHAT_WS_3`
`FEDORA`

Examples:

To tell the srpm that you have Red Hat AS 3, run this instead:

- `rpmbuild --rebuild --define DISTRO=REDHAT_AS_3 40002086_AA.src.rpm`

To tell the tgz that you have Red Hat AS 3, run this during the "configure" phase:

- `./configure DISTRO=REDHAT_AS_3`

- To install this package correctly under SuSE 8.0, a symlink from `/usr/lib/libncurses.so.2.1` to `/usr/lib/libncurses.so` needs to be created.

Simply run this command:

`ln -s /usr/lib/libncurses.so.2.1 /usr/lib/libncurses.so`

When the command completes, start the installation of the RealPort package.

- There is a known incompatibility/issues with the 2.6.26 kernel. Because of these incompatibility/issues, the **2.6.26 kernel is not supported with this driver.**
- There is a known incompatibility between our driver and the Red Hat 7.1 distribution which ships a custom modified 2.4 kernel. The problem manifests itself with device driver usage counts which never decrease, possible loss of use of some numbers of ports, and possible system instability. The root cause is in the modification of the kernel, which makes a change to an internal interface and introduces an incompatibility.
- There is a problem with the SBRK ioctl. When a BRK is sent no more data is transmitted until the port is closed.
- The ports on EM modules are observed to perform at a slightly reduced speed during throughput testing.
- The ditty PRINTER option is not yet supported.
- If this RealPort driver package detects that the system's OpenSSL layer is too old to be used reliably, the driver will compile its own more current version of OpenSSL. This compile could take a considerable amount of time, anywhere from 5 to 60 minutes extra. The extra time it takes depends upon how fast of system the driver is being compiled on.

DRIVER INSTALLATION

This driver is distributed in three formats: as source RPM and Debian packages, and as a compressed tar file of the source code.

Prerequisites

Building the driver and utilities requires that certain host system packages be pre-installed. The precise list of required packages and their names varies somewhat between distributions and versions, but here are some guidelines:

1. C compiler and related tools. On Debian based systems (e.g Debian 10, Ubuntu 18), apt install the "build_essential" package. On RHEL/CentOS systems, 'yum groupinstall "Development Tools"'. On OpenSuSE, "zypper install -t pattern devel_basis").
2. Linux kernel headers. On Debian based systems, install "linux-kernel-headers" and "kernel-package" or, for just the current kernel: "linux-headers-\$(uname -r)". For RHEL/CentOS and OpenSuSE, install "kernel-devel".
3. Development support for "ncurses", and "Tcl/Tk". On Debian based systems these are packaged as "ncurses-dev" and "tk-dev"; on RHEL/CentOS and OpenSuSE as "ncurses-devel" and "tk-devel".
4. Use your host OS-supplied OpenSSL libraries. On Debian based systems these are packaged as "libssl-dev"; on RHEL/CentOS and OpenSuSE as "openssl-devel". If these are not installed when the driver is built, it will use its own copy of the OpenSSL libraries, but this will substantially increase the build time.
5. To build the driver from the Debian Source Package on Debian based systems. Install the "devscripts" and "debhelper" packages. (Note: on some distributions, the debhelper files are included in the "devscripts" package, others require separate installation of both packages.)

6. To build the driver from the Source RPM package on RPM based systems such as RHEL/CentOS. Install "rpm-build".

Building/Installing from the three different release file formats:

- If you install the source RPM version of the source, it is usually as easy as running the following:
 1. `rpmbuild --rebuild 40002086_AA.src.rpm`
 2. `cd ~/rpmbuild/RPMS/x86_64`
 3. `rpm -i dgrp-1.9-40.x86_64.rpm`
- If you install the compressed tar version of the source, it is usually as easy as running the following:
 1. `tar xvfz 40002086_AA.tgz`
 2. `cd ./dgrp-1.9`
 3. `./configure`
 4. `make all`
 5. `sudo make install`
 6. `sudo make postinstall`
- To install the Debian Source package
 1. Make and switch to a clean working directory, e.g:

```
mkdir work
cd work
```
 2. Unpack the Debian source package files for the driver:

```
tar -xvzf 40002086_AA_debiansourcepkg.tgz
```

to get:

```
README
dgrp_1.9.40.orig.tar.gz
dgrp_1.9.40-1.dsc
dgrp_1.9.40-1.debian.tar.xz
```
 3. Build the Debian binary package customized for your environment

```
dpkg-source -x dgrp_1.9.40-1.dsc
cd dgrp-1.9.40
debuild -b -uc -us
```

This should create `dgrp_1.9.40-1_amd64.deb` in the parent directory.
 4. Install the new Debian binary package

```
# Note the leading "../" to fully specify the file pathname
sudo apt install ../dgrp_1.9.40-1_amd64.deb
```

ADDITIONAL INFORMATION

This package includes several support utilities:

- `ditty` - an `stty` replacement.

- dinc - a cu/tip replacement.
- There is a package of configuration tools provided that are, by default, located in the /usr/bin/dgrp/config directory. There are scripts for using the command line to add and remove products, as well as the RealPort Manager. This is essentially a graphical front end for the scripts and a visual tool for monitoring individual serial ports.
- The software package includes two startup scripts in either the /etc/rc.d/init.d or /etc/init.d directories named "dgrp_daemon" and "dgrp_ditty". These scripts will be
 - Executed at startup automatically if the "chkconfig" tool is present on the system during the post installation phase of the driver install. Otherwise, these scripts must be manually added to the system startup.
 - To load and initialize the driver by hand, execute "dgrp_daemon start". To stop the driver daemons and unload the driver, execute "dgrp_daemon stop" followed by "rmmod dgrp".
- The following manpages are provided:
 - ditty-rp(1), drpd(8), dgrp(8), dgrp_cfg_node(8), dgrp_gui(8).

- Red Hat 7.1 -- Kernel Compatibility Issues

Some 2.4 kernel-based distributions (Red Hat 7.1 included) have a patch applied to them which modifies the behavior of Linux when an open of a serial port is canceled (for instance, if an application is waiting for the carrier signal and a user hits CTRL-C to kill the application)

With this behavior change, the device driver is unable to clean up its internal data structures and the sane functioning of the driver is compromised. The classic symptom of this problem is that the command "lsmod", which (among other things) will return a count of the applications using the device driver, will return a non-zero value even if all applications associated with the serial ports are killed.

Unfortunately, it is impossible (from within the device driver) to determine which behavior is implemented in the running kernel. However, Digi now provides a workaround to allow customers with this problem to change the Digi behavior to be compatible with these "patched" kernels.

To enable the change which provides alternate behavior when a serial "open" call fails, execute the following after the device driver is loaded:

```
echo "alt_fail_open=1" > /proc/dgrp/info
```

To return to the standard behavior, execute:

```
echo "alt_fail_open=0" > /proc/dgrp/info
```

To examine the current state of this driver tuning variable:

```
cat /proc/dgrp/info
```

Unfortunately, this command must be executed each time the device driver is loaded (i.e. when the system is booted). To make this process easier for customers, the Digi standard driver startup tools will execute the command on the customer's behalf when it loads the driver if the following command is executed:

```
touch /usr/bin/dgrp/config/alt_fail_open
```

This command will create a file in your file system. The Digi tools, if they detect the existence of this file, will enable the "alt_fail_open" behavior on your behalf at every system boot.

- Running the RealPort (dgrp) driver on UEFI/SecureBoot systems.

If your system generates a "Required key not available" message when trying to load the RealPort (dgrp) driver, then you are probably running on a UEFI system with the SecureBoot option enabled. If you don't actually require the Secureboot functionality, the simplest solution is to boot into the UEFI setup menu (typically by pressing a special key like F2 when the system first boots) and disable SecureBoot, after which your locally compiled kernel modules such as the dgrp driver should load and run properly without further effort.

If you wish to keep SecureBoot enabled on your system, then you will need to sign kernel modules that you compile locally (such as the dgrp driver) with a Machine Owner Key (MOK). Details:

Step 1. Does your system generate a "Required key not available" error message when trying to load the dgrp driver?

If not, no further set up action is required on your system; the dgrp driver should load and operate normally. Go to Step 5.

Step 2. Have you previously created a Machine Owner Key (MOK) for your computer and do you still have access to its corresponding .der and .priv files?

If yes, skip to Step 3.

If no, create a new MOK with openssl.

For example:

```
openssl req -new -x509 -newkey rsa:2048 \  
-keyout MOK.priv -outform DER -out MOK.der \  
-nodes \  
-days 36500 \  
-subj "/CN=My_Own_MOK/"
```

This command will create two files -- MOK.der and MOK.priv -- which define your new machine owner key.

You can replace "36500" with the number of days in the future when you want the MOK to expire and "My_Own_MOK" with some identifying text that will help you recognize and distinguish this particular MOK from others when it is presented to you in a menu.

Step 3. Is your MOK already enrolled with your machine's UEFI system?

To see what MOKs are currently enrolled:

```
mokutil --list-enrolled
```

for an abbreviated version:

```
mokutil --list-enrolled | grep "Subject.*CN"
```

Is your MOK already enrolled with your UEFI system? If so, skip to Step 4.

If not, registering a new MOK with your UEFI and boot loader is a three part process:

- a. Mark the MOK with an import request

```
sudo mokutil --import MOK.der
```

This will request you to supply a password to be used to unlock the MOK during the next step.

You can confirm that the new MOK has been imported and is pending enrollment with:

```
sudo mokutil --list-new | grep "Subject.*CN"
```

- b. Then, on the next reboot you should be presented with the UEFI MOK Management screen (watch carefully since it may timeout if no action is taken) which will allow you to view and optionally enroll the pending MOK. View and enroll your new key (this will prompt for the password you created in step "a") to complete the MOK enrollment.

- c. Once the MOK has been accepted, the UEFI MOK Management screen will advise you that a subsequent reboot is required. This time the system should boot directly into Linux.

Once the Linux system is back up, confirm that your new MOK has been enrolled with:

```
mokutil --list-enrolled | grep "Subject.*CN"
```

Step 4. Sign the dgrp module with your machine owner key (MOK).

Assuming that your MOK.der and MOK.priv files in the current directory ("./."), sign the dgrp module using the "sign-file" script distributed with your kernel.

For example:

#Ubuntu 16 and similar systems

```
sudo /usr/src/linux-headers-$(uname -r)/scripts/sign-file \  
sha256 ./MOK.priv ./MOK.der $(modinfo -n dgrp)
```

#RHEL 7 and similar systems

```
sudo /usr/src/kernels/$(uname -r)/scripts/sign-file \  
sha256 ./MOK.priv ./MOK.der $(modinfo -n dgrp)
```

This looks for the "sign-file" script and the "dgrp.ko" module in the expected places for the currently running kernel (hence the "uname -r" and "modinfo -n" in the invocation).

You can confirm that the dgrp module has been signed with

```
tail -c 100 $(modinfo -n dgrp) | od -c
```

This should now report "Module signature appended".

Step 5. Proceed normally. You should now be able to load and run the dgrp driver and utilities such as "dgrp_gui" normally (without "Required key not available" error messages).

Known problems

As of this writing, CentOS 7.5 appears to have a shim bug that makes it fail to notice when new MOKs are pending enrollment, so the MOK Management Screen never appears. We can sign modules with existing MOKs, create new ones and schedule them for enrollment, but we cannot actually enroll new MOKs under CentOS 7.5 Other distributions including RHEL 7.5, Ubuntu 16.04 and OpenSuSE 42.3 work properly and, once a MOK is registered with one of them, all, including CentOS 7.5 can use it.

So, until the CentOS 7.5 bug is fixed, enrollment of new MOKs (step 3 above) must be performed on some other OS. Once a MOK is enrolled with the UEFI (a onetime process) it can be used to sign modules on any OS, including CentOS 7.5.

- Running the RealPort configuration GUI (dgrp_gui) under Wayland.

dgrp_gui is a Tk-base graphical user interface to the RealPort command line configuration utility, dgrp_cfg_node, originally written to run under X. Newer Linux distributions are moving towards replacing (at least as an option) X with Wayland (See www.wayland.freedesktop.org), which poses two problems for dgrp_gui.

1. Traditionally we expected dgrp_gui to be run with superuser privileges, invoked, for example as: "sudo dgrp_gui". Wayland requires that the main GUI application be run without elevated permissions; invoking only non-GUI portions (such as dgrp_gui's callouts to dgrp_cfg_node) at elevated permissions. So, as of the *_AA release, dgrp_gui explicitly does the dgrp_cfg_node callouts via "sudo". This means that it *will* run properly under both X and Wayland, but if the sudo credentials are not already cached (from some previous sudo use) when dgrp_gui is started, when it later comes time for the first dgrp_cfg_node invocation, the user will be password prompted. Once entered, the sudo password will normally be cached, so subsequent invocations will not prompt again.

But a potential cause for confusion is that the initial password prompt appears in the terminal window where dgrp_gui was invoked, rather than in the TK window, while the latter is probably the focus of the user's attention. So, if you start dgrp_gui without elevated privileges, be careful to look for the password prompt when it comes time for the first callout to dgrp_cfg_node.

Clumsy workaround: pre-cache your sudo credentials by running something else via sudo (e.g: "sudo ls") before starting dgrp_gui.

2. In some Wayland implementations, dgrp_gui graphics don't display properly. It turns out that the root cause is that Wayland sometimes doesn't set the display dimensions properly. This is not just a dgrp_gui problem since it also causes reference Tk programs like <https://tkdocs.com/tutorial/firstexample.html> to fail in the same way. Workarounds:

- a. Select X11 display server instead of Wayland at login.
- b. Run `dgrp_gui` via `"ssh -X"` from a client running under X11.
- c. Force your `dgrp_gui` session to use the Wayland reference compositor: `"weston"`. From a terminal session run `"weston --modules=xwayland.so"`, then click on the "Terminal" icon in the upper left corner to get a command prompt, and then run `"dgrp_gui"` from there. This is "almost" a solution: the buttons display and work properly, but the inventory line displays in a small font and doesn't seem to scale up. Still, as a workaround, it is basically usable.
- d. Use the CLI tool, `dgrp_cfg_node`, instead of `dgrp_gui`. See the `dgrp_cfg_node` documentation (`"man dgrp_cfg_node"`) for details, but here are some examples:
 - i. Configure a 16 port RealPort device at 192.168.1.201 for standard (unencrypted) communication using a "TS" id:
`/usr/bin/dgrp/config/dgrp_cfg_node init -v -v \ -e never TS 192.168.1.201 16`
 - ii. To enable encryption on the network connection, use the "-e always" option: `/usr/bin/dgrp/config/dgrp_cfg_node init -v -v \ -e always TS 192.168.1.201 16`
 - iii. Once the device is configured, start the daemon with:
`/usr/bin/dgrp/config/dgrp_cfg_node -v -v start TS auto`
 - iv. Stop the daemon with: `/usr/bin/dgrp/config/dgrp_cfg_node -v -v stop TS`

HISTORY

VERSION 1.9-39

- Added IPv6 address driver support for terminal servers (such as the ConnectPort TS 16) that provide them. (RP-99)
- Added support for newer kernels through (at least) 4.16. (RP-110)
- Updated the in-source openssl package to version 1.0.2o. (This is only used if openssl is not installed on the Linux host when the driver is installed.) (RP-117)
- Added "Running the RealPort (dgrp) driver on UEFI/SecureBoot systems" section to release notes. (RP-116)
- Fix edge case in port closing code that could lead to a kernel lockup. (RP-113)

VERSION 1.9-38

- Updated the in-source openssl package to version 1.0.2l. (This is only used if openssl is not installed on the Linux host when the driver is installed.) (RP-86)
- Added support for 4.X kernels. Tested with Ubuntu 14 and 16's 4.4.0-81 kernel, Debian 9.0.0's 4.9.0-3 kernel, OpenSUSE 42.3's 4.4.73.1 kernel, and with a custom 4.11.4 kernel from kernel.org. (NPI-489)

- The 3.10.0-514.el7 kernel used in the new RHEL 7.3 release introduced another backport fix related to TTY_CLOSING that caused the RealPort driver build to fail. (RP-93)

VERSION 1.9-36

- Fix an installation problem on systems such as RHEL 7 where `/etc/sysconfig/network` exists but does not define a NETWORKING environment variable.
- Fix a driver build problem on newer kernels, which removed definitions for TTY_CLOSING (in v3.11) and TTY_HW_COOK_IN (in v3.19). Note that Red Hat backported at least some of these definition removals into RHEL 7 kernels starting sometime shortly after 3.10.0-229, so this driver fix is also needed for their newer 3.10 kernels.
- Add a description of the `rtstoggle` option to the ditty man page.
- Fix a timing vulnerability where an application that fast cycles many port open/closes could occasionally obstruct a close from completing properly, thus leading to a "locked" port.
- Fix a problem that could cause the `dpa.dgrp` utility program to crash with an arithmetic exception if it happened to catch a port during the early stages of an uncompleted open.
- Fix the default prefix so that builds from the `.tgz` distribution file will properly install man pages to `/usr/share/man` (rather than `/share/man`).

VERSION 1.9-35

- Previously there was a time window at the beginning of port opens during which "dgrp_cfg_node" could mistakenly conclude that the corresponding node was idle. If "dgrp_cfg_node uninit" were called during this window it could crash the kernel. A new detection mechanism has been added to cover these conditions.

VERSION 1.9-34

- Fix driver build problem on systems (such as RHEL 7) that define `CONFIG_UIDGID_STRICT_TYPE_CHECKS`.

VERSION 1.9-33

- Added support for the now latest 3.X kernels (from 3.9.0 through 3.14).
- Use `/tmp/dgrp` (rather than `/tmp`) during installation to avoid the unintended side-effect of modifying the permissions on `/tmp`.
- Fix an initialization problem where first opens of transparent print devices (`/dev/pr*`) could crash 3.7.0 and newer kernels.

VERSION 1.9-31

- Added support for 3.X kernels through 3.8.
- Changed the dgrp udev rules:
 - a. Use SYMLINK rather than NAME since the latter causes problems on newer systems.
 - b. Add TAG="systemd" so dgrp devices will be recognized on systemd-based distributions (e.g: to start getty's).
- Updated configure scripts to add support for newer distributions.

VERSION 1.9-30

- Added support for kernels up to 2.6.38

VERSION 1.9-20

- Added support for kernels up to 2.6.28
- Added support for many new versions of distributions.
- Added support for new style "init" scripts in SuSE 11.1
- Fixed very rare data corruption issue when under very high loads and multiple PortServers and SMP.

VERSION 1.9-17

- Added support for Red Hat Enterprise 5,
- Added support for Fedora Core 5, 6 and 7.
- Fixed problem with registering our devices with sysfs in 2.6.18. Fix compile WARNINGS under 2.6.18.
- Fix a problem with the RealPort UDEV script when using new versions of UDEV.
- Fix problem on network disconnect/reconnects where the port would always bring up DTR/RTS, even in cases where the port had both signals down before the network disconnect.

VERSION 1.9-6

- Added support for DPA for RealPort. The utility is installed as /usr/bin/dpa.dgrp
- Added support for SYSFS in the 2.6 kernels.

VERSION 1.9-4

- Fixed problem with setting baud rates higher than 38400 when using the driver on the IBM pSeries (PPC64) platform.

VERSION 1.9-1

- Fixed driver not working under 2.6.10 kernels and higher.
- Fixed problem with adding similar node names to the driver.

VERSION 1.8-1

- Added RTS Toggle support to the driver.
- Added support for the new Alan Cox TTY layer changes in the kernel.
- Added support for Fedora Core 3, Mandriva Linux 10.1
- Added -n option back into the daemon.
- Added more race condition checking in the tty_close routine and the input routine.
- Fixed driver not working under 2.6.8 kernels and higher.

VERSION 1.7-1

- Added support for x86 64bit.
- Added support for Fedora Core 1, Core 2.
- Updated SSL version to 0.9.7d Removed support for Linux 2.2 kernels.
- Fixed problem with installing from the tar.gz image.
- Fixed problem with major() and minor() symbols not being found on very early versions of the 2.4 kernels.
- Fixed problem on setting the following "o flag" settings: nl<0/1> cr<0/1/2/3> tab<0/1/2/3> bs<0/1> vt<0/1> ff<0/1>
- Fixed man page of ditty, now correctly describes [-]DTR.
- Fixed dgrp_cfg_node to correctly set device permissions when using the -m flag. Fixed dgrp_cfg_node to correctly set owner and group when both options are selected. Fixed dgrp_cfg_node to correctly set WAN line speed.
- Fixed reporting DTR inconsistency when opening and closing the tty ports very rapidly. Reworked locking in dgrp_tty_write to fix possible deadlock. Fixed possible race condition in tty_close and dgrp_input.
- Added fix to resolve possible false "RealPort protocol error" detection.
- Fixed bug on port close. Port might not have gotten drained before the close.

VERSION 1.6-2

- Added support for RealPort with encryption.
- Added support for Red Hat 9, Suse 8.2, Mandriva Linux 9.1.
- Added support in ditty for the "startin" and "startout" options.
- dgipserv - Added support to set unassigned Ethernet address, support for setting the tftp server IP address, support for setting virtual ports and virtual ports timeouts, and support for setting optimize to throughput or latency.

- Fixed "configure" script problem where modversion.h wasn't being found in the proper location.
- Fixed include file problem where the driver would not compile correctly under some newer versions of the 2.4 kernel.
- Fixed problem with an incorrect assert message being generated.

VERSION 1.5-0

- Added transparent print support. Please read the man page for "dgrp", under section "Transparent Print" for more information.
- Changed package install to run a "configure" script first before trying to compile the driver. The script should catch most of the problems that the compile might have beforehand, and give a much better description of the problem than a cryptic compiler error message.
- Added Support for Red hat 7.3, Suse 8, Mandriva Linux 8.2, and Caldera OpenLinux 3.1.1 (server and workstation).
- Fixed off-by-one error in the tbuf parsing. This bug, although very rare, could have caused the kernel to panic, or cause random kernel memory corruption.

VERSION 1.4-0

- Improved support for Red Hat 7.1 by offering an architecture to workaround a problematic kernel patch shipped with that distribution.
- Added support for EL-8, EL-16 and EL-32.
- Added the "dgelreset" application, which allows one to reset an EtherLite regardless of whether it is running RealPort compatible firmware, and without having to specify a MAC address.
- Added "MODULE LICENSE" information to the device driver, in order to comply with anti-tainting policies in newer 2.4 Linux kernels.
- Modified driver to prevent port "hangup" in the event that the daemon dies.
- Added TCFLSH handling to "ditty-rp"... previously, the "flushin", "flushout", and "flush" options didn't work.

VERSION 1.3-0

- Added full support for Digi's EtherLite products which are running Digi RealPort compatible firmware, including the EtherLite 2, EtherLite 160, EtherLite 162, and the EtherLite 80.
- Fixed a kernel panic if a port is open, `dgrp_daemon stop` is executed, and a program then goes to write to the still open port. Now write returns EIO in that condition.
- Fixed a select(2) problem in kernel 2.2.15 and later of 2.2.x kernels which could cause programs like telnet or ssh to appear to hang until a key is pressed.

- Fixed a data loss problem which would manifest itself when exactly four kilobytes of data were transmitted after a port was opened.
- Added `/lib/modules/`uname -r`/kernel/drivers/char` to the list of locations that the kit will test for existence when trying to find an appropriate location to install the driver module, in response to testing in native 2.4 kernel-based distributions.

VERSION 1.2-2

- Increased support for Digi's ethernet-based serial connectivity family; including Digi One, EtherLite, and PortServer products.
- Added support for arbitrary integer baud rates via the "ditty-rp" utility.
- Detailed information about the type and version of configured remote devices was made available via the `/proc/dgrp/nodeinfo` file.
- A memory corruption problem which would cause system instability was eliminated.
- Unexpected pauses and loss in port traffic as a result of a network buffer overflow were resolved.

VERSION 1.1-7

- Added 2.4 kernel support.
- Removed references to invalid email addresses. Driver now handles
- Red Hat 7.0 (kgcc vs. gcc). The GUI tool now displays the DCD signal when viewing ports.
- RealPort ports now match other Linux serial ports for default settings, as well as making all port settings "sticky". As a result of this change, CLOCAL is now on by default. Configuration scripts remove the `/dev` files if a PortServer is unconfigured.
- Only physical line drops are detected as a "hangup" condition... the driver would previously treat a change from CLOCAL to -CLOCAL as a carrier transition.
- Startup scripts are installed in `/etc/init.d` if `/etc/rc.d/init.d` does not exist. RPM installation will not "error out" if "chkconfig" (a Red Hat tool) does not exist on the system.
- A message was added to the RPM installation to communicate to users whether the installation scripts were added or not.
- On newer systems, RPM would inadvertently strip the driver module of its symbols, causing it to fail. The module is no longer stripped.

VERSION 1.0-0

Initial Release