



# Universal Driver 7.0.0 for Linux Installation Instructions

Revision A.0

May 2015

| Revision | Date      | Comment         |
|----------|-----------|-----------------|
| A.0      | 5/13/2015 | Initial release |
|          |           |                 |

**FOR TECHNICAL SUPPORT  
PLEASE CONTACT:**

[support@diamondsystems.com](mailto:support@diamondsystems.com)

© Copyright 2015  
Diamond Systems Corporation  
555 Ellis Street  
Mountain View, CA 94043 USA  
Tel 1-650-810-2500  
Fax 1-650-810-2525  
[www.diamondsystems.com](http://www.diamondsystems.com)

# CONTENTS

|   |          |
|---|----------|
| <b>1. Introduction.....</b>                           | <b>3</b> |
| <b>2. Prerequisites.....</b>                          | <b>3</b> |
| <b>3. Installation.....</b>                           | <b>4</b> |
| 3.1 Steps to Compile Universal Driver .....           | 4        |
| 3.2 Install Universal Driver .....                    | 6        |
| 3.3 Uninstall Universal Driver .....                  | 7        |
| <b>4. Troubleshooting Linux Driver Problems .....</b> | <b>8</b> |

## 1. INTRODUCTION

Universal Driver 7.0.0 for Linux includes a kernel module to support interrupt features and PCI / PCIe bus interface. For interrupt features and PCI devices (i.e., PC/104-Plus, PCI-104, PCIe/104, FeaturePak, and PCIe MiniCards), the kernel module must be compiled on the target system before installing the driver, since the Linux kernel version varies based on the Linux OS chosen by the user.

If interrupts are not intended for use, or if ISA interface board (PC/104) is used, there is no need to compile or install this kernel module. In this case, extract the driver files and link to the driver library (libdscud-7.0.0.a) with the application program.

## 2. PREREQUISITES

### Things to Do Before Installing

To build a Linux kernel module the system should have the kernel header source code (and applicable C/C++ compiler). Additionally, the kernel modules are built for a specific version of the kernel, so if the module is built in one version and the module is loaded under different version, then errors or unexpected results may occur. Most likely, Linux will reject the kernel module as being in an invalid module format if from an incompatible system.

### 3. INSTALLATION

#### 3.1 Steps to Compile Universal Driver

- Change to the directory where the Diamond Systems' Universal Driver is present:  
**cd /usr/local/dscud-7.0.0**
- Run **./compile.sh**

#### EXAMPLE:

```
linux:/usr/local/dscud # ./compile.sh
```

```
Diamond Systems Universal Driver Installer
```

```
-----  
This installer compiles a kernel module which supports interrupt driven data acquisition features of  
Diamond Systems products.
```

```
If the interrupt driven features of the board are not being used then there would be no need to run this  
installer.
```

```
Online documentation for this driver is available at http://docs.diamondsystems.com/dscud/
```

```
Hit CONTROL-C at any time to exit this installer.
```

```
[[HIT ENTER TO CONTINUE]]
```

#### Step One: Locate Kernel Source Code

```
-----  
The installer will now scan the system looking for Linux kernel source code. The /usr/src directory and  
the /lib/modules directory will be scanned.
```

```
Select a Linux kernel version below that will run on the TARGET system. Type the number next to the  
selection and hit ENTER.
```

```
0) Kernel 3.2.24 (/lib/modules/3.2.0-29-generic-pae/build)
```

```
1) Kernel 3.2.24 (/usr/src/linux-headers-3.2.0-29-generic-pae)
```

#### For example type 1

#### Step Two: Compile Kernel Module

```
-----  
The installer is now ready to compile the Linux kernel module. If errors occur see the README file in this  
directory as well as the online Diamond System's Universal Driver documentation for help.
```

```
This kernel module will only load under the exact Linux kernel version which you have installed in the  
directory /usr/src/linux-headers-3.2.0-29-generic-pae.
```

```
[[HIT ENTER TO CONTINUE]]
```

```
--> Compiling kernel module for your system
```

```
<--rm -f dscudkp.ko dscudkp.o dscudkp.mod.*
```

```
make-C/usr/src/linux-headers-3.2.0-29-generic-pae SUBDIRS=/home/ami/Desktop/delete/dscud-7.00  
modules
```

```
make [1]: Entering directory `/usr/src/linux-headers-3.2.0-29-generic-pae'
CC [M] /home/ami/Desktop/delete/dscud-7.00/dscudkp.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/ami/Desktop/delete/dscud-7.00/dscudkp.mod.o
LD [M] /home/ami/Desktop/delete/dscud-7.00/dscudkp.ko
make [1]: Leaving directory `/usr/src/linux-headers-3.2.0-29-generic-pae'
--> Installing module dscudkp.o in /lib/modules/misc <--
mkdir -p /lib/modules/misc
cp dscudkp.ko /lib/modules/misc/
```

### Step Three: Final Instructions

-----

The dscudkp kernel module has been installed in `/lib/modules/misc/`. The file must be copied to the same location on the target system.

The `install.sh` script will load the kernel module so that it can be used by the driver. This script must be run each time the Linux system boots. See the README file for help with this.

Driver compilation complete.

- In the example above, the `dscudkp.ko` was compiled to be used on the system that has Linux-3.2.24. If User has completed all the steps in development PC, copy `/lib/modules/misc/dscudkp.ko` and `/usr/local/dscud-7.00` directory to target board using identical respective locations.
- Once copying the files to target system is done, change directory to the `dscud-7.00` Folder and run the command to install the driver.

## 3.2 Install Universal Driver

- Change the directory  
**cd /usr/local/dscud-7.00**
- Run `./install.sh`

This script file loads the Universal Driver kernel module and creates file as `/dev/dscud`.  
To confirm that the driver loaded successfully run the following command

**\$ lsmod | grep dscudkp**

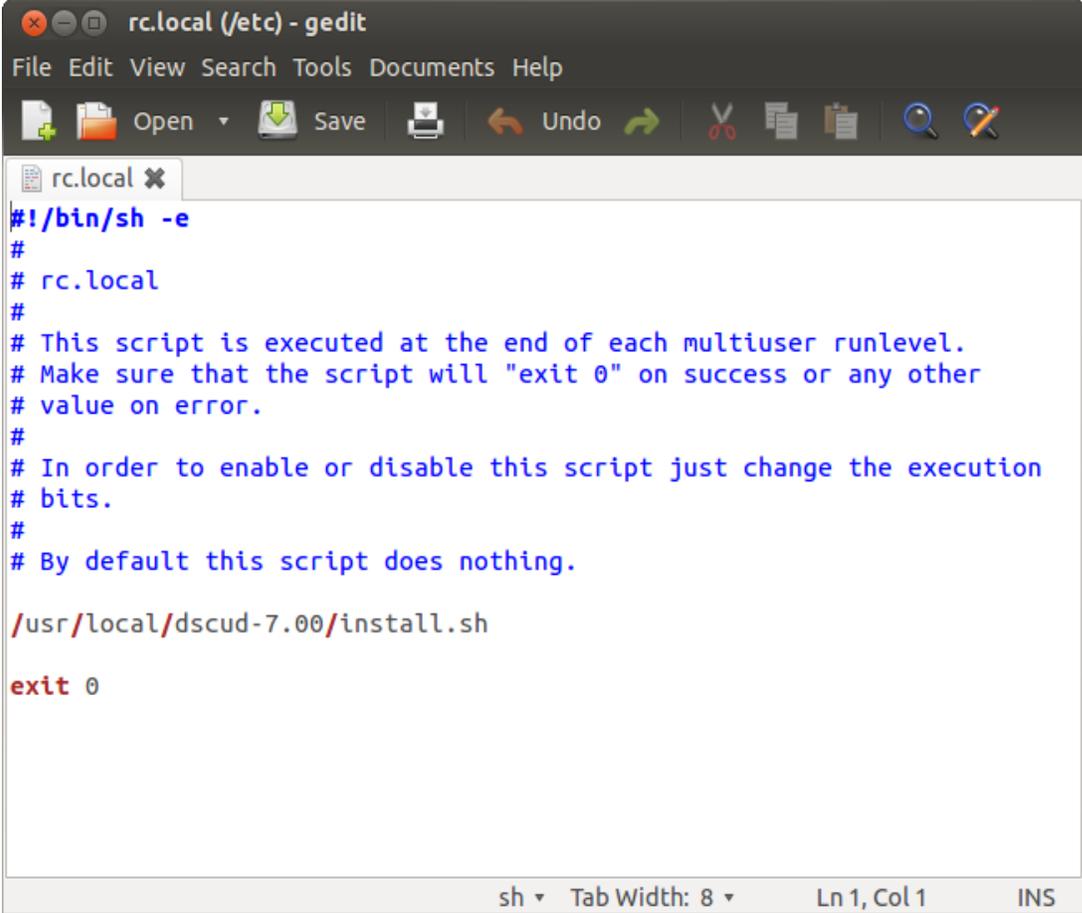
which will show the driver name filtered from the list of loaded drivers.

- Installing and setting up Diamond System's Universal Driver on the target system is done.

### Steps to install Universal Driver at boot up time

To load the Universal Driver (`dscudkp.ko`) at OS boot time, the `install.sh` file must be executed by the OS, in our example `/usr/local/dscud-7.00` directory. In order to run this script automatically by OS, It can call the script from `/etc/rc.local` which is the correct place for custom scripts to run at boot time. Hence add the script file path in `/etc/rc.local` file as follows

- Become root user  
`$ su root`
- Open `rc.local` file and add script file path  
`$ gedit /etc/rc.local`



```
rc.local (/etc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
rc.local x
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

/usr/local/dscud-7.00/install.sh

exit 0
sh Tab Width: 8 Ln 1, Col 1 INS
```

Figure 1: rc.local Script

### 3.3 Uninstall Universal Driver

Run the following command to uninstall the driver

```
$. /uninstall.sh
```

## 4. TROUBLESHOOTING LINUX DRIVER PROBLEMS

Here are some troubleshooting tips with problems in Linux.

- The driver must be run as the root user. If you would rather not do this to avoid potential security risks, you can use the `setuid ()` call to switch to a different user after calling `dsclnit ()`. The file permissions for `/dev/dscud` should be changed so that alternate user has read/write access to it.
- Check to make sure the `/dev/dscud` device exists. This device file is created by the `install.sh` script. It is required for communication with the `dscudkp.ko` kernel module. If it does not exist, run the `install.sh` script provided with the driver to create it.
- Run the `lsmod` command to make sure the `dscudkp` kernel module is loaded (there are often many modules loaded in Linux; using command ``lsmod | grep dscud`` will single that one out). If it is not listed, either the `install.sh` script was not run, or an error occurred. Run the `load.sh` script manually to check for errors.
- If the module loader complains of "unresolved symbols" this indicates that the module was built for a different version of Linux than you are running. Make sure that `/usr/src/linux` on the system where the driver was built matches the version of the kernel you are running on the target system. It is also a good idea to check the boot-loader settings (ex.: GRUB) to make sure the kernel version being booted is the same as the kernel header source being used to compile.
- If the module loader complains of "kernel-module version mismatch" this also indicates that the kernel source that was downloaded in `/usr/src/linux` on your development system does not match the version on the target system.
- If the `install.sh` script complains of "missing or wrong headers" this indicates that the full kernel headers was not found in `/usr/src/linux`. Make sure Kernel headers are present in the `/usr/src/linux` directory, if not download from Linux community website and install.
- To download the kernel source/header files for your Linux kernel, it is recommended to see if your package installer/organizer software has a pre-made package with your kernel in mind already, and download it/install it from the CD/DVD. If not, then it may be necessary to download it from [www.kernel.org](http://www.kernel.org) ; use command `'uname -a'` to find out which version you are running.
- To find help compiling kernel modules, contact the official help and forums for your specific Linux distribution first; or it may also be useful to try <http://www.linuxforums.org> , or for example: <http://ubuntuforums.org/>
- The most common kernel compilation problem is that in the future some pre-compiler directive is changed (example: `INTERRUPT_HANDLER_NAME_12`), where two names are merged, or one name is replaced with another:

```
INTERRUPT_HANDLER_NAME_12 & INTERRUPT_HANDLER_NAME_11 →  
INTERRUPT_HANDLER_NAME_MERGED
```

- The current release of the driver is compatible with Linux kernel versions 2.6.23, 3.2.0 and perhaps higher. The Standard Board Support Package (BSP) for DSC products as of the release date of this driver is Ubuntu 12.04 (LTS), using kernel version 3.2.0. We recommend to use this Linux distribution if possible for ease of use with Universal Driver.
- As there is a plethora of different Linux distributions and flavors available, we cannot provide detailed instructions for each one. Effort has been made to accommodate the general cases described here. If you have any questions with regard how to use the Universal Driver 7.0.0 for Linux using the board APIs, such as for data acquisition or other I/O, please feel free to contact Diamond Systems' Technical Support at: [support@diamondsystems.com](mailto:support@diamondsystems.com)