



DIAMOND SYSTEMS CORPORATION

Universal Driver Build Instructions for Linux.

Created By: Jonthan L. Montana

Introduction :

This document describes the build process for building a version of UD (Universal Driver) on any Linux operating system version supporting Linux kernel Linux-2.4.x.x – Linux-2.6.x.x. This document also lists the instructions on how to build a demo/test application to use the new UD driver built with the instructions.

Requirements:

- Linux-2.4.x.x – Linux-2.6.x.x source located in /usr/src (i.e., /usr/src/Linux-2.6.x.x)

<http://www.kernel.org/pub/linux/kernel/>

- Development tools: gcc, make, stdlib.h, etc.

<http://gcc.gnu.org/>

<http://www.gnu.org/software/make/>

- UD-6.01-Linux-2.6.23.tar.bz2 (The latest version is DSCUD-6.01 so we will use it in the examples below.)

```
dscud-6.01/dscudkp.ko
dscud-6.01/Makefile
dscud-6.01/dscud.h
dscud-6.01/dscudkp.h
dscud-6.01/libdscud-6.00.a
dscud-6.01/install.sh
dscud-6.01/uninstall.sh
dscud-6.01/load.sh
dscud-6.01/README
```

Universal Driver Build and Installation Procedures:

Step 1 : Copy DSCUD-6.00.x.tar.gz to current working directory on Linux development platform:

```
# cp --a UD-6.01-Linux-2.6.23.tar.bz2.'
```

```
# tar -jxvf UD-6.01-Linux-2.6.23.tar.bz2.
```

```
# mkdir -p /usr/local/ - if not already created
```



DIAMOND SYSTEMS CORPORATION

```
# mv dscud-6.01 /usr/local
```

```
# cp -a /usr/local/dscud-6.01/dscudkp.ko /lib/modules/misc
```

Step 2 : In this example, dscudkp.ko was compiled to be used on the system that has Linux-2.6.23. To use dscudkp.ko, it must be used on a system that has the Linux kernel compiled from the source that was used to create dscudkp.ko. To verify that your system has the correct Linux kernel version do '**# uname -a**' and verify the Linux kernel version that was used to create universal driver listed in the README.

Step 3 : Reboot the system or run '**# /usr/local/dscud-6.01/load.sh**' to load the dscudkp.ko. After powering-on the target board and booting-up OS with UD-6.01 support, make sure that dscudkp.ko is loaded and linked to the kernel.

EXAMPLE:

```
linux-target:/root # lsmod
Module          Size      Used by
Dscudkp         13028      0
```

We are now finished installing and setting up universal driver support on our target system.

NOTE: The load.sh script is not needed for 2.6.x.x kernels by default. The kernel should load the kernel objects on boot-up. If there is no module installed after verifying it with 'lsmod', then use 'load.sh'.

Compiling Universal Driver Demo/Test Application:

This section will show how to create a very basic demo/test application to use UD API support. You can create this on the target platform that contains the UD support or the development platform mentioned above in "**Dscud Build and Installation Procedures**" or on any Linux platform as long as you copy the dscud.h and libdscud.a into the locations mentioned above. If those locations don't exist, then they will need to be created.

EXAMPLE:

Below is an example of using the UD driver API to perform AD Auto calibration on the Helios board.

Create a Makefile with the following rules and specifications:

```
LIB=-L/usr/local/dscud-6.01 -L/opt/dscud-6.01 -ldscud-6.01 -lpthread -lm
INC=-I/usr/local/dscud-6.01 -I/opt/dscud-6.01
```



DIAMOND SYSTEMS CORPORATION

```
DEMOS=\
DSCADTest \
```

```
all: $(DEMOS)
```

```
clean:
    rm -f $(DEMOS)
```

```
$(DEMOS):
    ( gcc -g -o $@ $@.c $(LIB) $(INC) )
```

Create DSCADTest.c in the current working directory containing the newly made Makefile.

```
DSCADTest.c:
```

```
#include "dscud.h"
```

```
// Include any other system declaratives that are needed for the program.
```

```
.
.
.
```

```
int main( int argc, char **argv)
```

```
{
```

```
    //=====
    // I. DRIVER INITIALIZATION
    //
    //   Initializes the DSCUD library.
    //
    //=====
```

```
    if( dsclnit( DSC_VERSION ) != DE_NONE )
    {
        dscGetLastError(&errorParams);
        fprintf( stderr, "dsclnit error: %s %s\n", "0x09", "DSC Init Failed!");
        return 1;
    }
```

```
    //=====
    // II. BOARD INITIALIZATION
    //
    //   Initialize the Helios board. This function passes the various
    //   hardware parameters to the driver and resets the hardware.
    //
    //=====
```



DIAMOND SYSTEMS CORPORATION

```
printf( "\nHELIOS BOARD INITIALIZATION:\n" );

dsccb.base_address = 0x300;
dsccb.int_level = 5;

if(dscInitBoard(DSC_HELIOS, &dsccb, &board)!= DE_NONE)
{
    // Error strings
    return 1;
}

//=====
// III. AUTOCAL SETTINGS INITIALIZATION
//
//      Initialize the structure containing the AD autocal settings.
//      Set dscautocal.adrange for the MODE to calibrate. Set boot range
//      for the mode the board to be in upon power up.
//
//=====

/* PRE-FILLED EXAMPLE
dscautocal.adrange = 0xFF;
dscautocal.boot_adrange = 0;
*/

do {
    printf( "\n\nAUTOCAL SETTINGS INITIALIZATION:\n" );
    printf( "Enter the range to calibrate (0,1,2,3,5,6,7) or 255 to calibrate all ranges: " );
    scanf( "%d", &intBuff);
    dscautocal.adrange = (BYTE) intBuff;

    printf( "Enter the AD range to boot up in (0-7): " );
    scanf( "%d", &intBuff);
    dscautocal.boot_adrange = (BYTE) intBuff;
    getchar();

//=====
// IV. AUTO-CALIBRATION
//
//      Perform the actual AD auto-calibration of the board.
//
//=====

printf( "\n\nAUTO-CALIBRATION:\n" );
```



DIAMOND SYSTEMS CORPORATION

```
printf( "Starting AD auto-calibration...this will take a few seconds\n" );
```

```
if( dscADAutoCal( board, &dscAutocal ) != DE_NONE )
{
    dscGetLastError(&errorParams);
    fprintf( stderr, "dscADAutoCal error: %s %s\n", dscGetErrorString(errorParams.ErrCode),
            errorParams.errstring );
    return 1;
}

Return 0;
}
```

Compile this file by typing 'make' in the current working directory that contains the Makefile and DSCADTest.c. This will create a binary called DSCADTest which should be set to executable. To be sure it's executable type 'chmod +x DSCADTest'. Now copy this file onto your target device that contains the Universal Driver support as mentioned above in "**UD Build and Installation Procedures**" and run it './DSCADtest'.