



Application Note #4425

Getting Started with Galil's C-language API for Linux 2.6

The GNU/Linux operating system provides an alternative to Microsoft Windows for developing host applications and graphical user interfaces for motion control applications. This article gives an introduction to Galil's open-source C-language communication library (dmclnx.a) for Linux. It provides a quick start for those developing a Linux application that communicates with a Galil controller and leads you from downloading from the website to running your first sample program. Some familiarity with C, Linux, and Galil is assumed.

The library can be used to communicate with all Galil Ethernet controllers: DMC-14x5/6, 21xx, 22x0, and 3xxx. It can also be used to communicate with the following Galil PCI controllers: DMC-18x0 and 18x6.

Getting your first Galil-enabled application running involves several steps:

- (1) download the Galil communication library from the web site
- (2) unzip it to a directory on the hard drive
- (3) Build the Galil PCI driver (galilpci.ko) if using DMC-18x0/6
- (4) Install the Galil PCI driver if using DMC-18x0/6
- (5) Build the Galil communication library (dmclnx.a)
- (6) Build the sample applications
- (7) Run the terminal to verify communication
- (8) Run a sample application
- (9) Now you can use the sample application as a basis for your application

This article discusses each of the steps above in detail.

CONTENTS

1	ARCHITECTURE	2
2	DOWNLOAD FROM THE WEB SITE	2
3	UNZIP	3
4	BUILD PCI DRIVER (DMC-18X0/6).....	3
5	INSTALL PCI DRIVER (DMC-18X0/6).....	4
6	BUILD COMMUNICATION LIBRARY (DMCLNX.A).....	4
7	BUILD SAMPLE PROGRAMS	4
8	RUN TERMINAL	5
9	RUN A SAMPLE PROGRAM	5

1 Architecture

Figure 1 shows the architecture of the communication library (dmclnx.a) containing the Galil API (application programming interface). See dmclnxlib/dmclnx.h for the full API documentation. Your application program makes calls to the API such as DMCOpen() and DMCCCommand(), which in turn make calls to the Linux operating system. If you are using a Galil Ethernet controller, dmclnx.a will make calls to the BSD Sockets API to access the Ethernet hardware. If you are using a Galil PCI controller (DMC-18x0/6), dmclnx.a will make calls to the Galil PCI driver (galilpci.ko), which is a module that is loaded into the Linux kernel and handles the hardware communication across the PCI bus. Note that setting up a PCI controller may require that you are logged in as root and that you install the kernel development package for your Linux distribution.

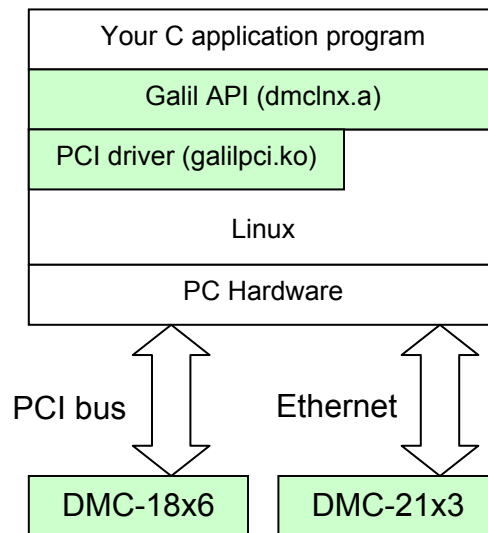


Figure 1. Architecture of Galil's communication library for Linux (Galil components are in green)

2 Download from the Web Site

The library can be downloaded from Galil's website here:

<http://www.galilmc.com/support/download.html#linux>.

This article discusses Linux version 2.6, but a similar download is available for Linux 2.4. When you are done, you should end up with a file called dmclnx2.6.tar.gz on your hard drive. This is a standard gzipped tar archive that needs to be unzipped.

3 Unzip

Unzip `dmclnx2.6.tar.gz` using the `tar` command as shown below. You will end up with a directory called “galil” whose structure is shown in figure 2. Note that there are three makefiles: one for the sample programs (`windows`, `simpeth`, `simppci`, and `term`), one for the PCI driver (`galilpci.ko`), and one for the library (`dmclnx.a`).

```
[root@localhost ~]# tar xvzf dmclnx2.6.tar.gz #unzip the archive
galil/
galil/simpeth.c
...
galil/term.c
[root@localhost ~]# cd galil #move into the new galil directory
[root@localhost galil]# ls #list the contents of this directory
dmclnxlib  makefile  PCIDriver  simpeth.c  simppci.c  term.c  windows.c
```

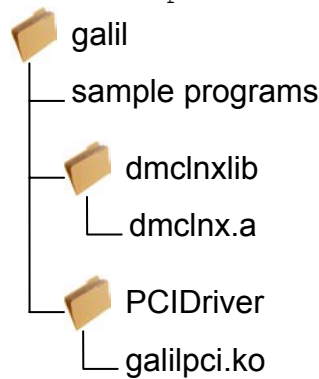


Figure 2. Directory structure of `dmclnx2.6.tar`

4 Build PCI Driver (DMC-18x0/6)

You may skip the following two sections if you are using an Ethernet controller. Next move into the `PCIDriver` directory and build the PCI driver with `make`. You MUST have the kernel development package installed to do this (on my Fedora Core 5 system, the package `kernel-devel - 2.6.15-1.2054_FC5-i686` is installed at `/usr/src/kernels/2.6.15-1.2054_FC5-i686`, which matches the output of `uname -r`). You should end up with a file called `galilpci.ko` (the PCI driver). Note that there are sample programs in this directory that access the PCI driver directly (bypassing `dmclnx.a`).

```
[root@localhost galil]# cd PCIDriver #move to the PCI driver directory
[root@localhost PCIDriver]# ls #list the contents of this directory
CheckSerialNumber.c  datarecord.c  galilpci.c  log.txt  main.c  Makefile
pciioctl.h  term.c
[root@localhost PCIDriver]# make #build the PCI driver
make -C /lib/modules/2.6.15-1.2054_FC5/build SUBDIRS=/root/galil/PCIDriver
modules
make[1]: Entering directory `/usr/src/kernels/2.6.15-1.2054_FC5-i686'
  CC [M]  /root/galil/PCIDriver/galilpci.o
  Building modules, stage 2.
  MODPOST
  CC      /root/galil/PCIDriver/galilpci.mod.o
  LD [M]  /root/galil/PCIDriver/galilpci.ko
make[1]: Leaving directory `/usr/src/kernels/2.6.15-1.2054_FC5-i686'
```

5 Install PCI Driver (DMC-18x0/6)

Once galilpci.ko is built, you need to install it into the running kernel with insmod (install module). You can verify it was loaded with lsmod and check the /dev directory for a device file to appear for each DMC-18x0/6 you installed into the PC. To use insmod and lsmod, you MUST install modutils (the module-init-tools package in Fedora, installed to /sbin). You can remove the module with rmmod galilpci.

```
[root@localhost PCIDriver]# insmod galilpci.ko #install the driver
[root@localhost PCIDriver]# lsmod #an entry for galilpci should show up here
Module                Size  Used by
galilpci              9824  0
...
jbd                   53077  1 ext3
[root@localhost PCIDriver]# ls /dev/galil* #there should be one file per DMC-
18x0/6
/dev/galilpci0 /dev/galilpci1 /dev/galilpci2
[root@localhost PCIDriver]# cd .. #move up one directory
```

6 Build Communication Library (dmclnx.a)

Next build the Galil communication library. Just run make in the dmclnxlib directory and you should end up with a file called dmclnx.a.

```
[root@localhost galil]# cd dmclnxlib #move to the com library directory
[root@localhost dmclnxlib]# make #build the communication library (dmclnx.a)
gcc -c -g dmcbin.c -o dmcbin.o
...
gcc -c -g dmcsup.c -o dmcsup.o
ar r dmclnx.a dmcbin.o dmccvs.o dmcdma.o dmceth.o dmcfmw.o dmcint.o dmcisa.o
dmclnx.o dmcpci.o dmcser.o dmcsup.o
ar: creating dmclnx.a
[root@localhost dmclnxlib]# cd .. #move up a directory
```

7 Build Sample Programs

Next build the sample applications by running make in the galil directory. Four programs will be built:

- (1) windows.c: shows how to write cross-platform code that will compile on both Windows (with dmc32.dll) and Linux (with dmclnx.a)
- (2) simppci.c: example for PCI controllers
- (3) simpeth.c: example for Ethernet controllers
- (4) term.c: terminal for both PCI and Ethernet controllers

```
[root@localhost galil]# make #build the sample programs
gcc -g -Wall -DLINUX windows.c dmclnxlib/dmclnx.a -Idmclnxlib -owindows
gcc -g -Wall simppci.c dmclnxlib/dmclnx.a -Idmclnxlib -osimppci
gcc -g -Wall simpeth.c dmclnxlib/dmclnx.a -Idmclnxlib -osimpeth
gcc -g -Wall term.c dmclnxlib/dmclnx.a -Idmclnxlib -oterm
```

8 Run Terminal

Next run the terminal to verify communication with the controller. The example below is for an Ethernet controller (select 1 if you have a PCI controller) and prints the controller's model and serial numbers. You should see a colon (:) prompt after hitting the Enter key. Note that the Ethernet controller must have a pre-assigned IP address (see <http://www.galilmc.com/support/appnotes/software/note4422.pdf>).

```
[root@localhost galil]# ./term #run the terminal on an Ethernet controller
Galil Terminal Program for Linux Ver. 1.1.0
```

```
Galil Motion Control Inc. 7/2000
```

```
Please specify which type of controller you are using
```

```
1 for PCI controllers
2 for ETHERNET controllers
```

```
2
```

```
Please specify IP address:
```

```
10.0.0.67
```

```
Connected. Ctrl-C to exit.
```

```
:^R^V
```

```
DMC2182 Rev 1.0o
```

```
:MG_BN
```

```
5532.0000
```

```
:
```

```
Thank you for using Galil!
```

9 Run a Sample Program

Next run the sample application for your controller (simpeth for Ethernet controllers and simppci for PCI controllers). After this, you are ready to start modifying simpeth.c or simppci.c and turn it into your motion application program!

```
[root@localhost galil]# ./simpeth 10.0.0.67 #run the Ethernet sample
Simple Ethernet communications example for Galil Motion Controllers
```

```
Connected to DMC2182 Rev 1.0o
```

```
:
```

```
The serial number is 5532.0000
```

```
:
```

```
The value of the inputs is 255
```

```
The value of the X axis motor position is 4000
```

```
The value of the X axis commanded position is 8000
```

```
The value of the X axis position error is 4000
```

```
The value of the X axis commanded position using QR is 8000
```

```
Done
```