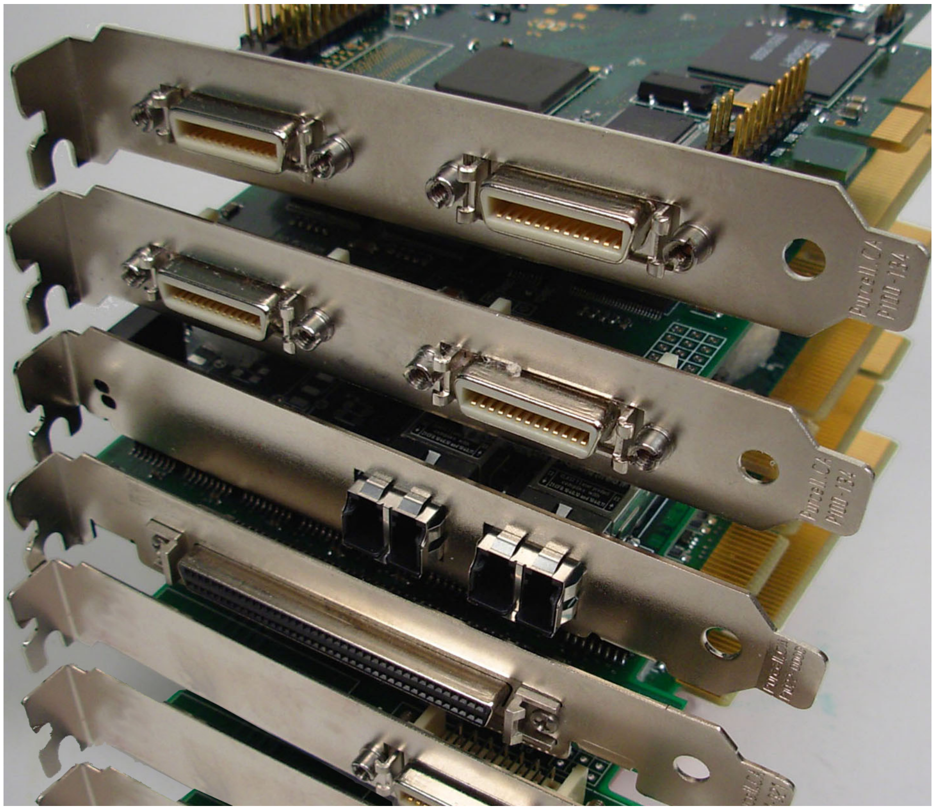
PCI DV Family

PCI Digital Video Boards



May 21, 2007
008-00966-13

a HEICO company

The information in this document is subject to change without notice and does not represent a commitment on the part of Engineering Design Team, Inc. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

Engineering Design Team, Inc. ("EDT"), makes no warranties, express or implied, including without limitation the implied warranties of merchantibility and fitness for a particular purpose, regarding the software described in this document ("the software"). EDT does not warrant, guarantee, or make any representations regarding the use or the results of the use of the software in terms of its correctness, accuracy, reliability, currentness, or otherwise. The entire risk as to the results and performance of the software is assumed by you. The exclusion of implied warranties is not permitted by some jurisdictions. The above exclusion may not apply to you.

In no event will EDT, its directors, officers, employees, or agents be liable to you for any consequential, incidental, or indirect damages (including damages for loss of business profits, business interruption, loss of business information, and the like) arising out of the use or inability to use the software even if EDT has been advised of the possibility of such damages. Because some jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitations may not apply to you. EDT's liability to you for actual damages for any cause whatsoever, and regardless of the form of the action (whether in contract, tort [including negligence], product liability or otherwise), will be limited to $50 (fifty U.S. dollars).

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, without the express written agreement of Engineering Design Team, Inc.

Copyright © Engineering Design Team, Inc. 1997–2007. All rights reserved.

EDT and Engineering Design Team are trademarks of Engineering Design Team, Inc.

Camera Link is a trademark of the Automated Imaging Association.

Kodak is a trademark of the Eastman Kodak Company.

Redlake and Megaplus are trademarks of Redlake MASD, Inc.

UNIX is a registered trademark of The Open Group.

Visual C++ and Windows are registered trademarks of Microsoft Corporation in the United States and other countries.

VxWorks is a registered trademark of Wind River Systems, Inc.

Xilinx is a registered trademark of Xilinx, Inc.

PCI DV C-Link FCC Compliance

This device complies with part 15 of the FCC Rules. Operation is subject to two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

CAUTION: Changes or modifications not expressly approved by Engineering Design Team, Inc. could void your warranty to operate this equipment.

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his or her own expense.

Contents

About the PCI Digital Video Products

The PCI DV is a single-slot PCI bus board that implements a high-speed direct-memory-access (DMA) channel between an external digital video camera and a host computer. The PCI DV family of products includes a number of different options for drivers and receivers, specialized connectors, bus form factors, fiber-optic extension systems, and products that conform to the Camera Link™ specification.

Common device drivers and APIs have been maintained across the product line, and except for a few operations that apply only to specific models, applications written for one PCI DV product work without modification on other products in the PCI DV family.

The PCI DV family of products consists of:

PCI DV	PCI DVK
PCI DV44	PCI DVa
PCI DV C-Link	PCI DV C-Link/SC
PMC DV C-Link	cPCI DV C-Link
PCIe DV C-Link	PCI DV FOX, and the RCX or RCX C-Link when connected to a PCI DV FOX

Throughout this guide, the term PCI DV is used generically to refer to all products in the PCI DV family, except where noted.

References

The following related publications may prove useful:

Manual	URL
Camera Configuration Guide	www.edt.com/manuals/PDV/camconfig.pdf
EDT DMA & Digital Video Software Library	www.edt.com/api (HTML)
EDT DMA & Digital Video Software Library	www.edt.com/manuals/misc/api.pdf (PDF)
PCI DV C-Link Hardware Addendum	www.edt.com/manuals/PDV/pcidv_cl_add.pdf.
PCI DV AIA Hardware Addendum	www.edt.com/manuals/PDV/aiag_add_cl2.pdf

System Requirements

The PCI DV is designed to run on various platforms and capture data from a variety of cameras. Due to the high speeds and large amounts of data involved, your system must meet certain minimum speed, memory and OS requirements.

Bandwidth

EDT digital video boards are high-speed DMA devices that depend on adequate PCI bus bandwidth for reliable operation. Because different cameras run at different speeds, the system PCI bus requirements vary depending on the camera in use. Therefore, choose and configure your system with PCI bandwidth requirements in mind.

EDT PCI DV, PCI DVK, and PCI DV44 boards (and their PMC and cPCI counterparts) are all 33 MHz boards with a maximum bandwidth of around 90 MB per second. PCI DVa, PCI DV C-Link, and PCI DV FOX are 66 MHz boards and require a 66 MHz bus to operate at full speed. Assuming that those boards are in a 66 MHz or faster PCI or PCI-X slot, they can sustain transfers of approximately 200 MB per second. If your camera exceeds these limits, then data loss — broken images or timeouts — will occur. Details are available in the Troubleshooting section, under Bandwidth Problems on page 22.

Installation

To install the PCI DV:

1. Install the Pdv driver software as specified on the software disk jacket.

NOTE Remove previous software releases before installing updates.

2. For those boards that require it, check board jumper settings. See Setting Control Signal Levels on page 4 for details.
3. Install the board into the computer according to your hardware manufacturer's instructions.
4. Cable the board to the camera. For details of which cables are required for which boards, see Cables on page 3.

Included Files

See the README for complete list of included files. The PCI DV driver package includes a capture and display application with a graphical user interface, as well as C source and executable for several command-line example, diagnostic, and utility programs. Applications mentioned in this manual include:

`pdvshow`	Capture and display application with a graphical user interface.
`take`	Captures images from the installed camera and (optionally) writes them to files. Also includes many diagnostic options.
`simple_take`	A simplified version of `take` to use as an example application, optimized to acquire and process data in parallel.
`simplest_take`	An even more simple example with no parallel optimization of data acquisition.

`simple_*`	Additional simple examples show other specific functionaility — see your distribution disk.
`serial_cmd`	Sends serial commands to an AIA serial camera in the appropriate format.
`edt_sysinfo`	Collects information about the board and the system in which it's installed, and writes the data to a file which you can then email to EDT Support for troubleshooting help. More information is available in edt_sysinfo on page 11.
`dvinfo`	Runs several image capture and other diagnostic tests and collects the results in a file which you can then email to EDT Support for troubleshooting help. More information is available in dvinfo on page 12.
`cl_speed`	A utility application for measuring the maximum bus bandwidth on Camera Link boards. More information is available in Testing on page 19.

For a more complete description of these applications, see Using the Example Applications on page 9. See the README file in the installation directory for a list, and descriptions, of all the examples and utilities.

Building Applications

PCI DV files are in the distribution directory `/opt/EDTpdv` (on UNIX® systems) or `C:\EDT\pdv` (on Windows® systems).

The package includes both executables and C source code for all of the examples, diagnostics and utilities. To rebuild any of these programs, you'll need development tools — a compiler and the `make` (UNIX) or `nmake` (Windows) application. To do so:

1. Run Pdv Utilities (Windows) or navigate to the installation directory in a terminal window (UNIX).
2. Enter:

```
make file
```

where *file* is the name of the example program you wish to build.

To build and install all the example programs, enter:

```
make
```

You can also build the sample programs by setting up a project in Windows Visual C++.

Cables

To connect your camera to the PCI DV, we offer a variety of cables. Part numbers and ordering information are available at www.edt.com/cables. Cables are available for the following boards:

PCI DV C-Link — The PCI DV C-Link is designed to be used with standard Camera Link cables. Cable assemblies may be provided with the camera. Cables and connectors are also available from EDT.

PCI DVK, RCX — If you've ordered the PCI DVK or the RCX and intend to use the cable from the camera manufacturer, make sure you specify the correct cable. With Redlake™ Megaplus™ (formerly Kodak™) cameras, specify the RS-422 cable. It has Redlake-standard 68-pin connectors on both ends and no RS-232 breakout connector. EDT makes cables for other cameras.

RCX, RCX C-Link, PCI DV FOX
Fiber-optic cables are available from EDT in six-foot (CAB-FOC-6) and 200-foot

(CAB-FOC-200) lengths. For other lengths, order from a fiber cable company according to EDT specifications. Two that we've worked with are:

Borg Technologies	www.borg-tech.com
Stratos Optical Technologies	www.stratoslightwave.com

The RCX C-Link is designed to replace the standard Camera Link cable. A variety of adapters are available from EDT if you have an interference issue, such as a need for a 90-degree bend off the Camera Link connector. For an adapter, contact EDT.

PCI DV, PCI DVa	Cables for supported cameras are available from EDT. These cables are specific to each camera model; specify the camera manufacturer and model when you order. Custom cables are also available; contact EDT for more information.
PCI DV44	For older DVC 44-pin cameras only. Specify the DV44 cable, EDT part number 016-01515-01. It has a DVC-standard 44-pin connector.

The appropriate connector pinout for each cable can be found in PCI Pinouts or PMC Pinouts.

Setting Control Signal Levels

If you're using a Camera Link camera with a Camera Link board, the board is configured appropriately at the factory and will not have jumpers to set serial control signal levels. You do not need this section.

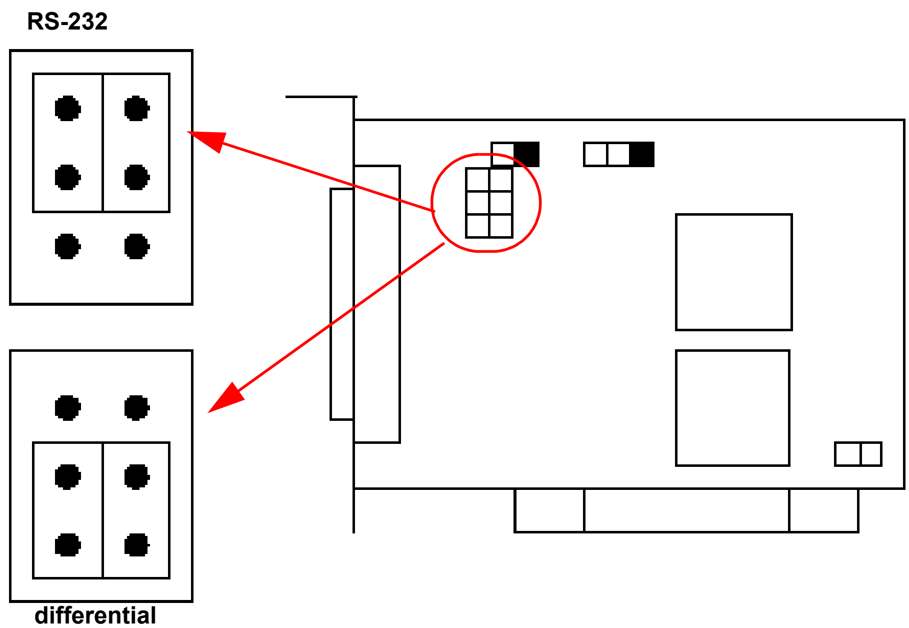
However, to provide serial control of the camera directly from the interface board, many AIA cameras have serial transmit and receive lines coming out of the digital interface connector. These lines have either differential (RS-422 or LVDS) or RS-232 signal levels. The PCI DVa and RCX LVDS and RS-422 products are configured at the factory for differential serial signal levels.

Many EDT cables have signal level converters built in; if you're using such a cable, leave the board at the default differential setting even if the camera outputs RS-232 serial. EDT cables that pass RS-232 straight through are labeled as such; when using those cables, set the jumpers for RS-232.

NOTE The PCI DV, PCI DVK, etc. and older RCI units (EDT part numbers 053-01082-00 and 053-01082-01) are fixed for differenial levels. EDT cables for these products have built-in RS-232 to differential converters; therefore, the default differential setting on the PCI DVa is the correct setting if you are using any cable built for the PCI DV or PCI DVK.

Figure 1 shows the appropriate jumper settings for the PCI DVa:

Figure 1. PCI DVa Serial Signal Level Jumper Settings



Setting the Camera Model

After installing the board and driver, you must configure it for the camera you wish to use. Sample configuration files are provided for various camera models.

NOTE If you're using a medium-mode camera, you must first reprogram the flash PROM on the the board by executing `pciload pdvcamlk_pir`. For details, see Table 1, "Arguments To pciload For Specific Boards" on page 26.

If you wish to use a camera model for which no configuration file is provided or to modify the directives of an existing configuration file, see the Camera Configuration Guide available at www.edt.com/manuals/PDV/camconfig.pdf.

On a Windows System

To configure the driver for a specific camera model using PdvShow:

1. Launch PdvShow by double-clicking the desktop icon.
2. The first time you run PdvShow, a dialog prompts you to choose your camera model. Choose from the list.
3. Click **OK**.

To reconfigure the driver for a different camera model or operating mode:

1. Launch PdvShow.
2. Execute **Camera > Setup**.
3. In the resulting dialog, choose the required model.
4. Click **OK**.

On a UNIX System

To configure the driver for a specific camera model:

1. Navigate to the installation directory `/opt/EDTpdv`.
2. At the prompt, enter:

   `camconfig`

3. In the resulting dialog, choose the required model.
4. Click OK.

To reconfigure the driver for a different camera model or operating mode, rerun `camconfig`.

Initialization Without PdvShow

Based on your choice of camera model and operating mode in the initial dialog, the application PdvShow initializes the board and camera by running a utility program called `initcam`. If you don’t wish to run PdvShow, you can run `initcam` yourself to initialize your board and camera. To do so, at a command prompt, enter:

```
initcam -f camera_config/file.cfg
```

where *`file`* is the appropriate camera configuration file for your camera model and operating mode.

The `initcam` utility optionally allows you to specify a unit and channel number, using the `-u` and `-c` flags, respectively. Thus, for example, if you have one PCI DV C-Link board connected to two base-mode cameras, addressed as channel 0 and 1, respectively, and you wish to initialize only the camera on channel 1, enter:

```
initcam -u 0 -c 1 -f camera_config/file.cfg
```

For more information on units and channels, see Addressing the Board and Camera on page 6.

For more information on camera initialization and configuration, see the Camera Configuration Guide available at www.edt.com/manuals/PDV/camconfig.pdf.

Addressing the Board and Camera

The number of data paths in a given PCI DV and host system depends on:

- the number of *units* — that is, the number of boards in the host;
- the number of *channels* per board — that is, the number of cameras connected to each board; and
- the number of taps in each camera.

The final item is outside of the scope of this manual; see the Camera Configuration Guide available at www.edt.com/manuals/PDV/camconfig.pdf. The other two — *units* and *channels* — are central.

Units

The number of PCI DV boards in a system is limited only by the number of available bus slots and the required bandwidth. (See System Requirements on page 2 for a discussion of bandwidth issues.)

When a system includes more than one board, each board is addressed by a unit number. Most frequently, this is an argument to the `-u` flag when invoking an application (such as `take`), or an argument passed into one of the Digital Video Library routines.

Boards are numbered starting with unit 0, the first board in the system.

Channels

A *channel* is a single data path from one camera, through one connector, to a PCI DV board, and from there output to one DMA channel of the host computer.

Some cameras have two taps, which their manufacturers may sometimes refer to as channels, in order to double their throughput. These taps are not what is meant, in this manual, by the term *channel.*

For example, the PCI DV C-Link has two connectors and can connect to two cameras. The PCI DV C-Link therefore has two channels, regardless of the number of taps possessed by either or both of the cameras connected to it. The PCI DV FOX can be ordered with one or two connectors, thereby offering one or two channels, again without reference to the individual characteristics of the cameras connected to it.

Channels are also numbered starting with zero. Channel 0 always uses the connector closest to the PCI Bus connector; the next connector accesses channel 1. If more connectors are present, the channels are numbered from the bottom up.

NOTE Although the software numbers channels starting with 0, the connectors on the board are labeled starting with 1. The connector labeled **1** is therefore channel 0, and the connector labeled **2** is channel 1.

Camera Link base, medium, and full modes implement different Camera Link specifications — base mode uses just one connector, while medium and full mode use two connectors and cables on a single camera.

PCI DV C-Link boards allow the two channels to be used for:

- two base-mode cameras, or
- one medium-mode camera with two connectors.

NOTE Due to bandwidth limitations, the PCI DV C-Link cannot support a full-mode camera.

When you use the PCI DV C-Link with two base-mode cameras, the software assigns each camera its own DMA channel, serial channel, and set of registers. It creates two devices, named `pdv0_0` and `pdv0_1` — each device having a unit number of 0 and a channel number of 0 or 1, respectively. Any application that seeks to access these cameras, whether one of our example applications or your own, must therefore specify a channel number.

For details about firmware to support the different modes, see Updating the Firmware on page 24.

Application Interface to Units and Channels

If you are using two base-mode cameras connected to one PCI DV C-Link board, open each device with the Digital Video Library routine `pdv_open_channel()`, which takes a channel number as an argument, instead of the more generic `pdv_open()`, which opens only a device connected to channel 0.

Routines that open either a unit, or a unit and a channel, return a pointer to the structure that represents the device in software; this pointer appears in our documentation as `pdv_p`. Many other routines accept that pointer and manipulate the structure. Because each represents a data path from camera to host memory, there is no difference between the pointer returned as a result of a request to open a unit, and that returned from a request to open both a unit and a channel; each is functionally equivalent to the other.

Image Capture and Display

The PCI DV comes with a general-purpose application that provides a graphical user interface for capturing and displaying images, called PdvShow.

PdvShow

Use PdvShow to capture an image from the installed camera (or simulated data), display it on the host computer monitor, and manipulate the displayed data in a few basic ways.

To start the application:

1. Double-click the PdvShow icon on your desktop, or enter `pdvshow` at a command line prompt.
2. The command line invocation allows you to specify options:

   `pdvshow -pdvn_c`

   where *n* is the unit number (useful if you have more than one PCI DV device) and *c* is the channel number for multichannel devices. For example:

   `pdvshow -pdv0_1`

... runs PdvShow using board 0, channel 1. This is useful, for example, if you have one PCI DV C-Link board connected to two base-mode cameras, and you wish PdvShow to access the camera on channel 1.

NOTE The command line is one of the properties of the icon. If you wish to use an icon to access a device other than the default unit 0, copy the icon, rename it, and change the shortcut properties to use the command line with the `-pdvn_c` option.)

For demonstration or debugging purposes, you can run this application when no board is installed in the system; the image window then shows a test pattern. To do so, at the command line, enter:

`initcam -u dmy0 -f` *`configuration file`*

`pdvshow -dmy0`

3. If you have not initialized the driver already, choose your camera (or the desired simulator) from the list displayed, and click **OK**.
4. If you have changed camera models since the last time the driver was initialized, the image window may show incorrect data. Choose **Camera > Setup** to select the new camera model.

Use the PdvShow toolbar and menus to access camera controls. For further instructions, see the online Help.

Using Camera Link Control Panels

Many manufacturers of Camera Link cameras supply a Windows-based graphical camera control application that can be used to change the camera's settings. These applications differ greatly from each other, but they all must have a way to specify a Camera Link-standard frame grabber DLL through which to send and receive camera control commands.

The PCI DV DLL is named `clseredt.dll`, installed in the folder `WINDOWS\system32`. Refer to your camera documentation for instructions on linking to a frame grabber DLL.

Most such applications need only be pointed at `WINDOWS\system32\clseredt.dll`, but in some cases it can be necessary to rename or copy the file to a different location. If this is the case, take care to ensure that you recopy any newer versions of this file to the appropriate location when you update the EDT driver package.

NOTE Depending on camera model and operating mode, PCI DV initialization may send commands to the camera to put it into the expected state; therefore, avoid running any third-party camera application while also running an EDT camera initialization operation such as `initcam` or `camconfig`, or the PdvShow camera setup dialog.

Using the Example Applications

A variety of example, utility, and diagnostic applications are provided. All can be run from the command line, and have UNIX-style options and arguments.

To help those developing PCI DV applications, C source code is provided for all the examples. For those just beginning, we recommend starting with the source for `simple_take` or `simplest_take`, as those are the easiest applications to understand.

The most commonly useful options to these programs are described below. Placeholders intended to be replaced by your own values appear in italics.

For a complete list of usage options, at the command line, enter the application name with the `-h` option to display the help message.

NOTE When running under VxWorks, be sure to enclose each argument in double quotes (").

take

`take` is a command-line application used to acquire images and (optionally) save them to files. Though it does not display the images, its many options make it a useful diagnostic tool. The source also shows how to change camera settings such as integration time; tune image acquisition in certain ways; and detect errors. Several of the most useful options are:

`-b` *filename*	The base name of the file in which to save the image, in Sun raster format (on UNIX systems) or Windows bitmap format (on Windows systems). If only one image is taken, this is the filename; otherwise a two-digit number is appended to each file, starting with 00. The appropriate suffix is automatically added.
`-c` *channel*	On multichannel boards, selects the channel to access — by default, zero.
`-f` *filename*	The name of the file in which to save the image, in raw format. The file includes only raw image data, with no formatting information.
`-l` *loopcount*	The number of consecutive pictures you wish to take. The default is one.
`-N` *numbufs*	The number of ring buffers — by default, one. A setting of four optimizes pipelining — one ring buffer currently acquiring data, one ready for data, one getting ready, and one backup.
`-u` *unit number*	The unit number, when multiple boards are installed in the host. The default is zero, indicating the first board.
`-v`	Turns on verbose mode. The default is off.

For example, to acquire one hundred images as fast as possible using four ring buffers, without saving them to files:

```
take -N 4 -l 100
```

Or, for example, if you have one PCI DV C-Link board connected to two base-mode cameras, and you wish `take` to use the camera on channel 1, enter:

```
take -u 0 -c 1 -N 4 -l 100
```

simple_take

`simple_take` shows how to use the PCI DV digital video library to acquire one or more images from a camera connected to the PCI DV and (optionally) save them to files. A suitable starting point for someone needing to add digital video acquisition to an application, this example shows how to use the ring buffer subroutines to pipeline image acquisition and subsequent processing, thus improving performance.

Several of the most useful options are:

`-b` *filename*	The base name of the file in which to save the image, in Sun raster format (on UNIX systems) or Windows bitmap format (on Windows systems). If only one image is taken, this is the filename; otherwise a two-digit number is appended to each file, starting with 00. The appropriate suffix is automatically added.
`-c` *channel*	On multichannel boards, selects the channel to access — by default, zero.
`-l` *loopcount*	The number of consecutive pictures you wish to take. The default is one.
`-N` *numbufs*	The number of ring buffers — by default, one. A setting of four optimizes pipelining — one ring buffer currently acquiring data, one ready for data, one getting ready, and one backup.
`-u` *unit number*	The unit number, when multiple boards are installed in the host. The default is zero, indicating the first board.

For example, to acquire four images as fast as possible using four ring buffers, saving each to files named `picture00.bmp` through `picture03.bmp` (or `.ras` on UNIX-based systems):

```
simple_take -N 4 -l 4 -b picture
```

simplest_take

As the name implies, `simplest_take` is the simplest example application. It sets up four ring buffers and acquires a single image, with no pipelining.

`simplest_take` accepts an optional argument of a file name to which to save the image. If no name is supplied, it reports a successful image acquisition, or any errors that occurred — useful for testing.

For example, to acquire an image and save it to a file named `pic.bmp`:

```
simplest_take -b pic.bmp
```

serial_cmd

The `serial_cmd` application sends serial commands to a camera through one of the PCI DV family of digital video boards, using calls to subroutines in the PCI DV digital video library.

By default, the application starts in command mode: the final argument to `serial_cmd` is the command to send to the camera. Delimit this command with single or double quotation marks — either works, as long as you're consistent. For example:

```
serial_cmd "MDE?"
```

If you omit the command argument, the application enters interactive mode, in which you can type one command per line. To quit the application, enter Control-C.

Several of the most useful options are:

`-c` *channel*	On multichannel boards, selects the channel to access — by default, zero.
`-u` *unit number*	The unit number, when multiple boards are installed in the host. The default is zero, indicating the first board.
`-x`	Treats the command argument as a hexadecimal number, which is sent to the camera without terminating nulls or carriage returns. The default is ASCII with a terminating carriage return added.

An example of command mode usage:

```
% serial_cmd "MDE?"      (Redlake "Query Mode" command)
MDE TR                   (camera response)
%
```

An example of interactive mode usage:

```
% serial_cmd
>MDE?                (Redlake "Query Mode" command)
MDE TR                   (camera response)
>
% serial_cmd -x      (for hexadecimal arguments)
> 03 06 02           (camera-dependent command)
> Control-C          (end the program)
```

To access a camera on channel 1 (for example, if you have two base-mode cameras connected to one PCI DV C-Link), enter:

```
% serial_cmd -u 0 -c 1 "MDE?" (Redlake "Query Mode" command)
MDE TR                        (camera response)
%
```

edt_sysinfo

The `edt_sysinfo` application gathers technical information about the EDT board and the system in which it's installed, then writes the results to the file `edt_sysinfo.out` in the current directory. This file can then be useful for diagnosing problems; it's an ASCII text file that you can view and include in email for technical support.

Run edt_sysinfo with the camera connected and powered on, if possible.

dvinfo

`dvinfo` gathers information specific to the PCI DV digital video board(s) installed, and runs tests; it then writes the results to the file `dvinfo.out` in the current directory. This file can then be useful for diagnosing problems; it's an ASCII text file that you can view and include in email for technical support.

Run `dvinfo` with the camera connected and powered on, if possible.

One useful option is:

`-u` *`unit number`* The unit number, when multiple boards are installed in the host. The default is zero, indicating the first board.

PCI DV Programming Interface

Programmers can interface to the PCI DV using either the PCI DV digital video library (`libpdv`) provided with the PCI DV software, or a combination of PCI DV library and the DMA library (`libedt`) calls.

The PCI DV digital video library provides a C language interface to the PCI DV; it also handles the necessary bookkeeping, error-recovery and camera shutter trigger and timing tasks. We therefore recommend using it for all PCI DV-specific programming; use the lower-level DMA library only where it provides needed functionality not provided in the digital video library. Routines in both libraries are documented online at www.edt.com/api.

Triggering

By default, most cameras power up in freerun mode, sending images continuously. For most camera models, EDT supplies configuration files that put the PCI DV board in a compatible mode in which it waits for the timing signals that indicate camera image acquisition: the frame valid, line valid, and data valid (pixel clock) signals and then receives the image data. (For details on configuring the PCI DV for various camera models, see Setting the Camera Model on page 5.)

However, some cameras have one or more external trigger modes, in which the camera waits for a signal from the PCI DV board or an external source (such as a photo diode) to trigger the shutter and begin image acquisition.

You can change trigger modes in your application or using configuration file directives. The discussion below describes the most common triggering modes and the configuration file directives, serial control commands, and software considerations that apply to them.

NOTE For details of the hardware registers involved, see the hardware manual for your specific board. For details of camera configuration directives, see the Camera Configuration Guide available at www.edt.com/manuals/PDV/camconfig.pdf.
For details of specific serial control commands, see your camera manual.

Continuous

Also known as *freerun*, the camera acquires images continuously without waiting for a trigger signal.

Configuration file directives	`MODE_CNTL_NORM: 00` (the default) `serial_init`: as needed to set freerun mode (usually not necessary because most cameras power up in freerun by default)

Triggered by the PCI DV

The camera waits for a trigger signal from the PCI DV board before acquiring an image.

Configuration file directives	`MODE_CNTL_NORM: 10` `serial_init`: as needed to put the camera in triggered mode

Pulse-width Triggered

Also known as controlled or level timing in some camera manuals, exposure duration is determined by the period of time that the EXPOSE line is held TRUE (high).

Configuration file directives	`MODE_CNTL_NORM: 10`
	`method_camera_shutter_timing: AIA_MCL`
	`serial_init`: as needed to put the camera in pulse-width mode

PDV library routine	`pdv_set_exposure`, millisecond units, range 0–25500

Some cameras have very fast shutters and can accept exposure times in microseconds. For such cameras, a different configuration file directive tells `pdv_set_exposure` to use units of microseconds instead of milliseconds:

Configuration file directives	`MODE_CNTL_NORM: 10`
	`method_camera_shutter_timing: AIA_MCL_100US`
	`serial_init`: as needed to put the camera in pulse-width mode

PDV library routine	`pdv_set_exposure`, microsecond units, range 0–25500

External Trigger Direct to Camera

This triggering mode sends a trigger signal directly from an external source to the camera, bypassing the PCI DV board completely. In this case, the camera and board can be configured as in free-running mode. However, unless an image arrives before the timeout period has expired, your application will time out while waiting for an image. You can avoid timeouts in either of two ways:

- If the maximum period of time between triggering signals is known, configure the `user_timeout` period in the software for a large enough value to avoid timeouts.
- if application blocking is acceptable, configure the `user_timeout` period in the software for an infinite period (`user_timeout=0`) to ensure that the application waits until an image arrives.

Configuration file directive	`MODE_CNTL_NORM: 00` `user_timeout` as needed to ensure that your application does not time out while waiting for an image
PDV library routine	`pdv_set_timeout`, millisecond units

External Trigger Pass-through

This triggering mode sends a trigger signal from an external device to the board, and from the board to the camera. A TTL signal is input to the board, which in turn sends an LVDS or RS-422 signal (depending on the board and its configuration) out to the camera trigger line, typically CC1.

Configuration file directive	`MODE_CNTL_NORM: A0` `user_timeout` as needed to ensure that your application does not time out while waiting for an image
PDV library routine	`pdv_set_timeout`, millisecond units

The same software timeout considerations apply as in External Trigger Direct to Camera, above.

External Triggering Pins

The pins to which to connect the trigger source are shown in Figure 2 through Figure 12. Trigger input 0 triggers camera 0, and trigger input 1 triggers camera 1. Fire the trigger by applying a TTL signal lasting at least 10 microseconds to these pins, which in turn send a signal of the appropriate level to the camera trigger line, typically CC1. The trigger cable can drive either pin high with respect to the other; no particular polarity is required.

The two pins of each trigger drive a Vishay SFH6206 optocoupler through a 130-ohm series resistor. This optocoupler is provided to allow coupling to electromechanical systems in which major ground spikes can occur when electrical devices such as motors, for example, turn on or off.

Figure 2. PCI DV C-Link External Trigger Pins

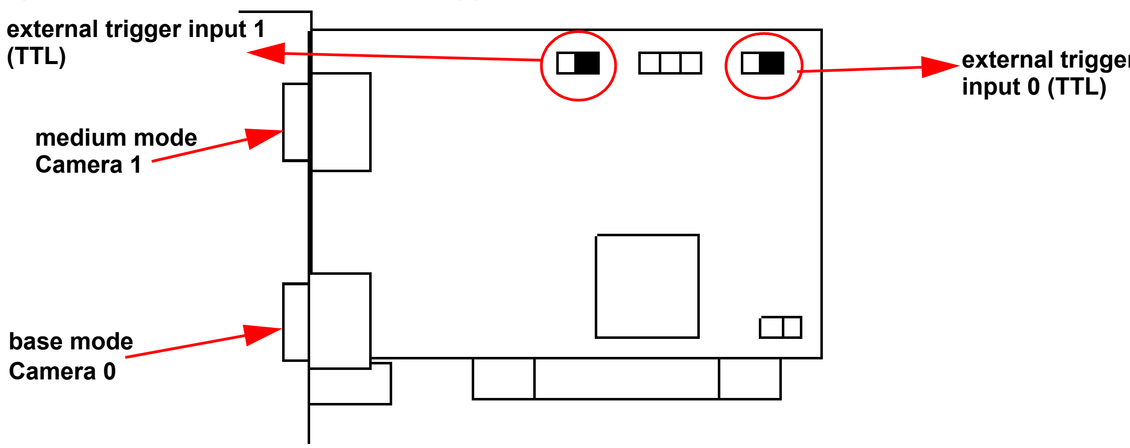
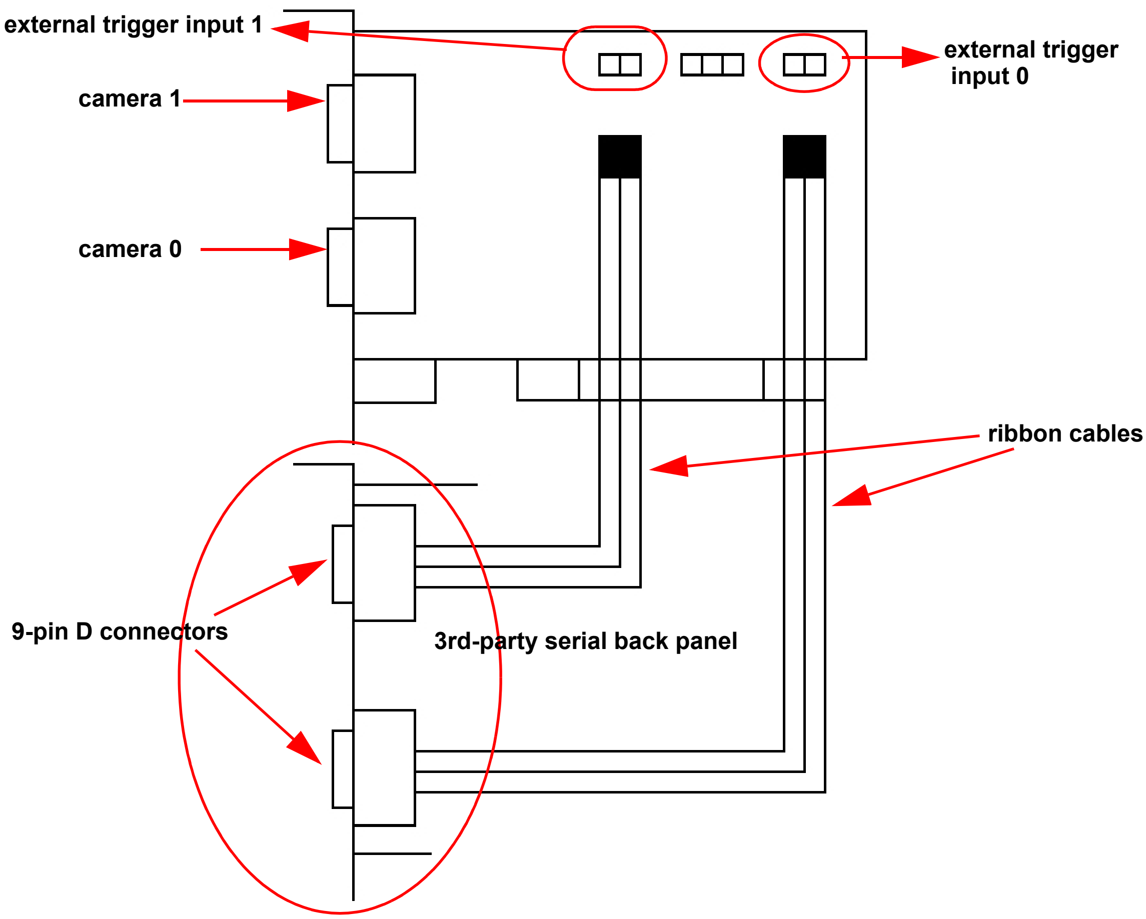


Figure 3 shows one of several ways to connect an external trigger signal to the trigger pins: a serial back panel with two 9-pin D-connectors and ribbon cables extending from them. You can use any two pins of the 9-pin connectors; just make sure that you're using the same two pins on each end. EDT does not provide these back panels, but they are commonly available through third-party vendors.

Figure 3. Connecting the PCI DV C-Link To an External Trigger Source



The PCIe DV C-Link has a ten-pin connector and a seven-pin Lemo connector for use as external triggers. External trigger input 0 uses pins 1 and 2, and external trigger input 1 uses pins 3 and 4, of either connector. Pins 7 and 10 of the ten-pin connector, and pin 7 of the Lemo, are ground. Other pins are reserved.

You can use both trigger inputs 0 and 1 simultaneously, but you cannot use both trigger inputs 0, or both trigger inputs 1, at the same time.

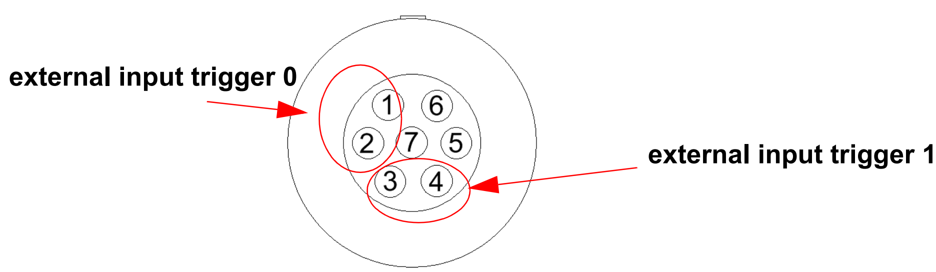


Figure 4. Lemo Pin Layout

Figure 5. PCIe DV C-Link External Trigger Pins

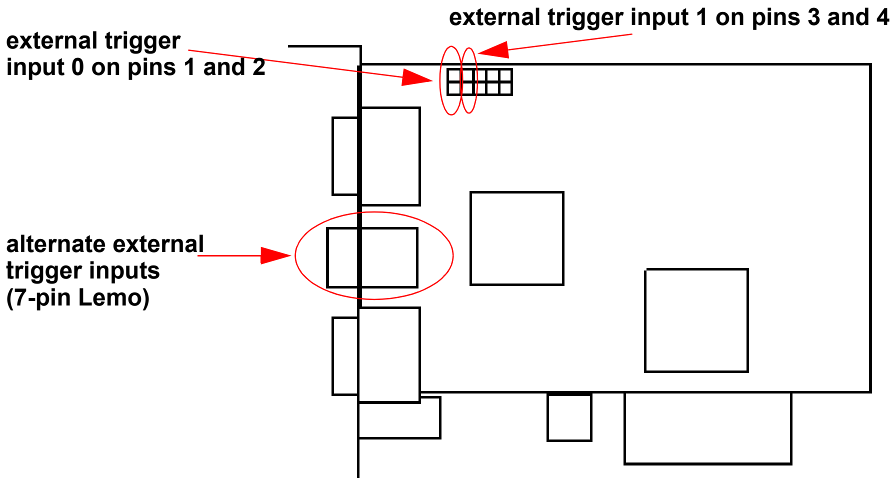


Figure 6 shows one of several ways to connect an external trigger signal to the trigger pins: a serial back panel with two 9-pin D-connectors and ribbon cables extending from them. You can use any two pins of the 9-pin connectors; just make sure that you're using the same two pins on each end. EDT does not provide these back panels, but they are available through third-party vendors.

Figure 6. Connecting the PCIe DV C-Link To an External Trigger Source

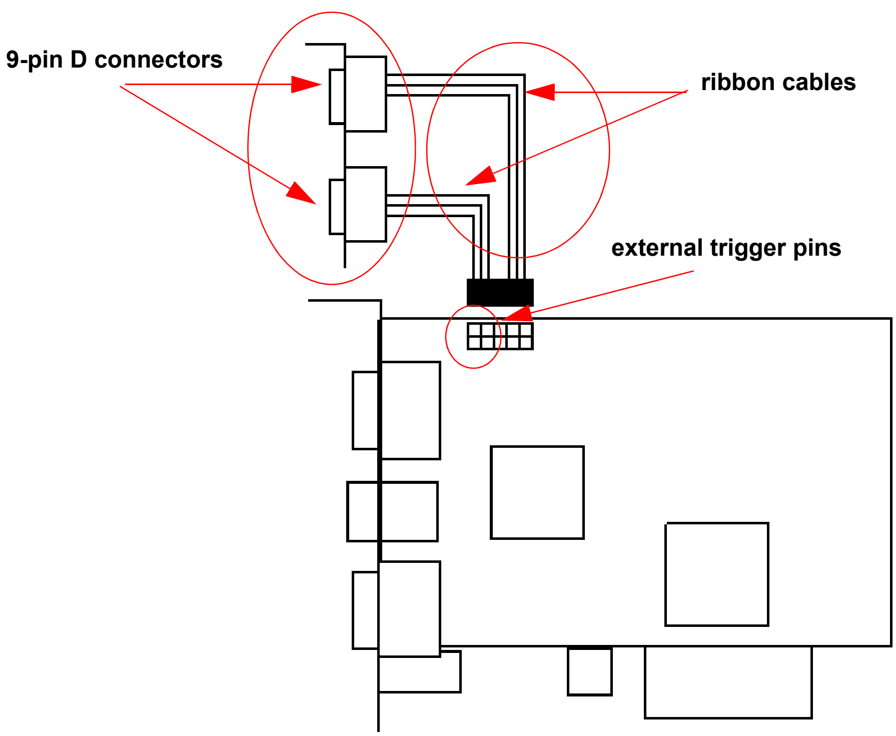


Figure 7. PMC DV C-Link External Trigger Pin

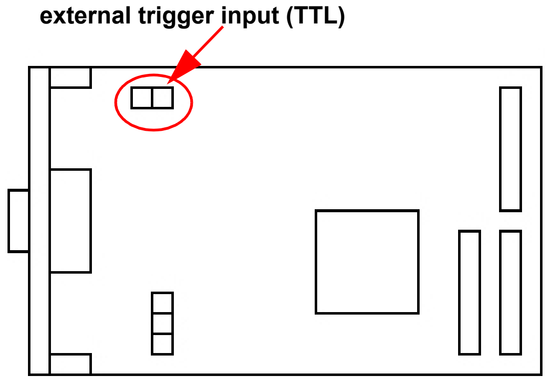


Figure 8. cPCI DV C-Link External Trigger Pins

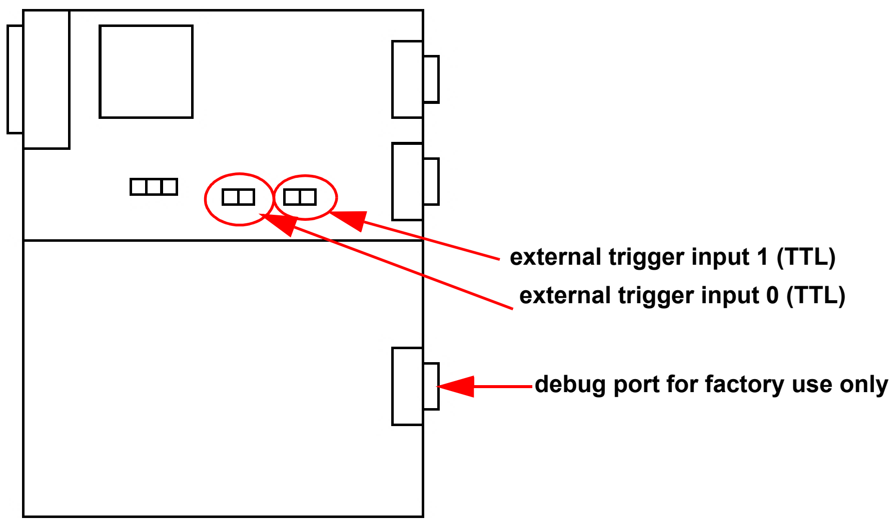


Figure 9. PCI DV FOX External Trigger Pins

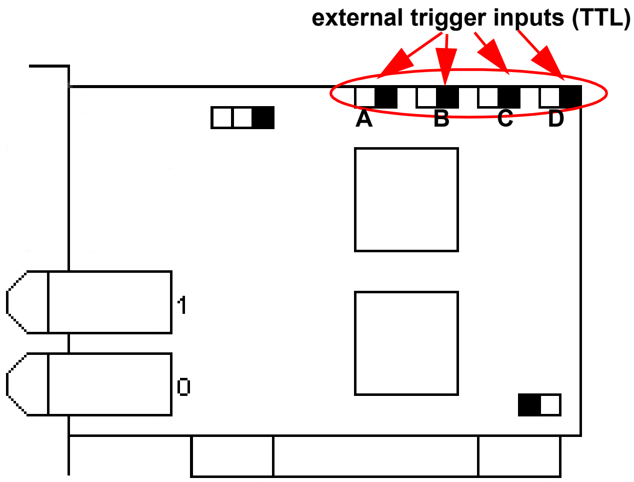


Figure 10. PCI DVa External Trigger Pin

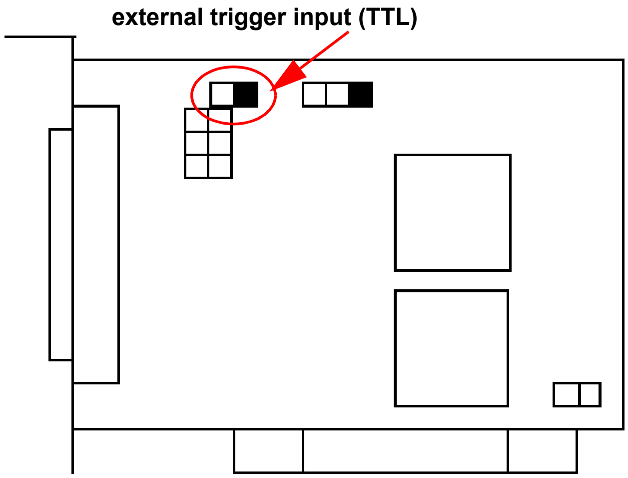


Figure 11. PCI DVK External Trigger Pin

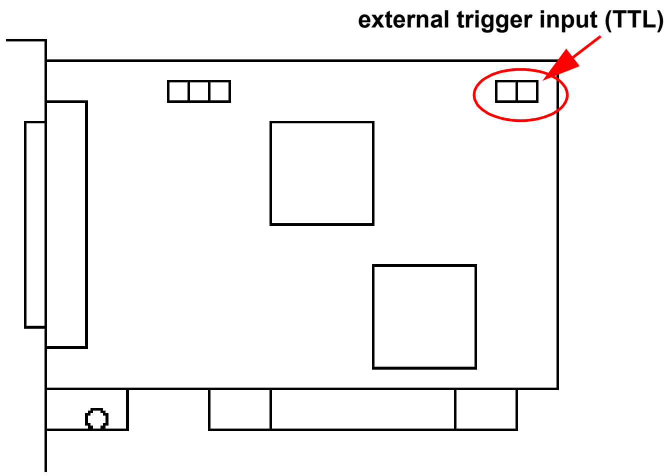
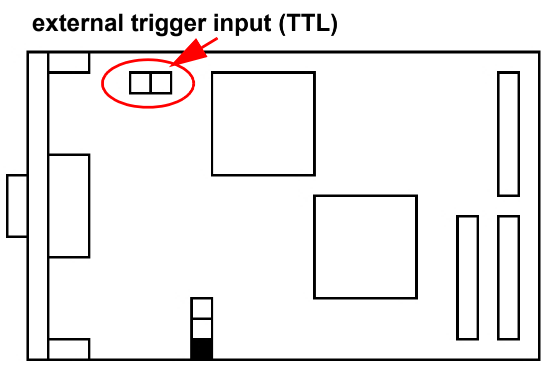


Figure 12. PMC DVK External Trigger Pin



Testing

In addition to the example applications PdvShow and `take` (see Using the Example Applications on page 9), several of the PCI DV family of boards offer special simulation capabilities for more in-depth testing.

PCI DV C-Link Simulation

The PCI DV C-Link includes a simulated channel 2, allowing you to generate sample data without a camera attached to the board; it is therefore useful for testing the PCI DV C-Link hardware. The Channel 2 simulator uses a simple counter to generate 16-bit pixel data; pixels start black and grow progressively lighter until they reach white.

Data is generated as fast as the bus speed allows: for a 33 MHz bus with no contention, 90–120 MB per second; for a 66 MHz bus with no contention, 190–220 MB per second.

To use the simulator with PdvShow:

1. Start the application with:

   `pdvshow -pdv0_2`

2. Choose the **Camera Setup** menu item.
3. Select the desired configuration.

To use the simulator with your own application:

1. *Initialize the board to use channel 2.*
   Use the initialization application `initcam`, specifying channel 2 and the camera configuration file for your camera. From a command prompt, enter:

   initcam -f camera_config/*yourCamera*.cfg -c 2

   ...replacing the italicized text with the configuration file name appropriate for your camera and application. This initializes the board registers and driver with the appropriate values for your camera.

NOTE For general testing, use one of the `generic`*xx*`_cl.cfg` files provided with your board.

2. *Specify channel 2 to your application.*
   Open the device with a call to `pdv_open_channel`, passing it `NULL` for the device, the board unit number, and 2 for the channel. For example, for an application opening a board with a unit number of 0:

   ```
   PdvDev *pdv_p = pdv_open_channel(NULL, 0, 2);
   ```

   The pointer returned from this call points to the simulated data; any image acquisition calls, such as `pdv_wait_image`, that pass this pointer will access the simulated data.

Because data is generated as fast as possible, you can also use the Channel 2 simulator to measure the maximum possible bus bandwidth, using the utility application `cl_speed`. This utility takes the unit number as an argument and begins Channel 2 simulation on the specified board; it then reports the data speeds achieved.

PCI DV and PCI DVK Simulation

The PCI DV and PCI DVK boards offer two specialized configuration files allowing you to generate sample data at a fixed rate:

`camsim.cfg`	provides simulated 1024 x 1024 x 8 camera data at 5 MHz.
`camfst.cfg`	provides simulated 1024 x 1024 x 8 camera data at 20 MHz.

You can generate this sample data using the image capture and display application PdvShow, the configuration application `camconfig`, or the initialization application `initcam`.

To do so using PdvShow:

1. Choose the **Camera Setup** menu item.
2. From the resulting list, choose **EDT Simulator 8 bit 5 MHZ** or **EDT Generic Simulator 20 MHz**.
3. Click **OK**.

To do so using `camconfig`:

1. In the initial dialog, choose **EDT Simulator 8 bit 5 MHZ** or **EDT Generic Simulator 20 MHz**.
2. Click **OK**.

To use `initcam`, from a command prompt, enter:

```
initcam -f camera_config/camsim.cfg
```

or:

```
initcam -f camera_config/camfst.cfg
```

PCIe DV C-Link Simulation

When loaded with a specialized FPGA configuration file, certain revisions of the PCIe DV C-Link can operate as a simulator. If you need this functionality, contact EDT.

Troubleshooting

The following sections offer suggestions if you encounter any difficulties. To submit a bug report or ask for further technical assistance, see our Contact page.

A number of diagnostic utilities are provided with the PCI DV device driver package. The most useful of these are briefly described in Using the Example Applications on page 9.

When contacting EDT for technical support, provide the output from the applications that are most relevant, as well as the output from the utilities `edt_sysinfo` and `dvinfo`.

Diagnostic Programs

The `edt_sysinfo` program gathers generic board and system information and writes the results to the file `edt_sysinfo.out`. For more information, see edt_sysinfo on page 11.

The `dvinfo` program gathers information specific to the PCI DV digital video board(s) installed, and runs tests; it then writes the results to the file `dvinfo.out` in the current directory. For more information, see dvinfo on page 12.

If you are having trouble, run these applications before contacting us. If possible, send the output files along with your query, or have them available when you call.

Problems Installing Software

Check that you have the latest release of the software. For instructions on downloading the latest release from our web or ftp site, see our Software page.

Remove previous software releases before installing updates.

On Windows systems, reboot the computer after installing the software if you have more than one board in the host. You are prompted to do so when it's necessary.

Also on Windows systems, if a board isn't recognized after installing the software, use the **Device Manager** located under the **Hardware** tab of the **System Properties** control panel to uninstall the unknown EDT device. Then choose **Scan for Hardware Changes** from the **Action** menu.

Do not extract and install files by hand and circumvent the installation process. The installation procedure updates system files and creates links as well as creating new files.

Similarly, do not remove files manually. Instead, use `pkgrm` (for Solaris systems), or the **Add or Remove Programs** facility (on Windows systems). On Linux systems:

- If *package*.run was used to install the package, remove it with ./uninstall.sh.
- If the package was installed using rpm --install *package*, remove it with rpm -erase *package*.

Corrupted Images, Timeouts, or Slow Acquisition

Corrupted images, repeated timeouts, or slow acquisition rates usually indicate that the bus is too slow or the driver is misconfigured. To correct this:

- Check the camera configuration file in effect to make sure it's the correct one for the camera model and operating mode in use.
- When updating to a new device driver, always recompile and relink applications that use EDT software libraries.
- Ensure that the PCI DV board is not sharing the bus with other devices that reduce the available bus bandwidth.
- Determine the bandwidth required by the camera in use, then ensure that both the board and the host can sustain the required throughput. For faster cameras or multiple camera installations, choose a host that has a single or multiple 66 MHz or faster PCI or PCI X buses. See Bandwidth Problems below for details.

Bandwidth Problems

EDT PCI DV, PCI DVK, and PCI DV44 boards (and their PMC and cPCI counterparts) are all 33 MHz boards with a maximum bandwidth of around 90 MB per second. PCI DVa, PCI DV C-Link, and PCI DV FOX are 66 MHz boards; assuming that the board is in a 66 MHz or faster PCI or PCI-X slot, these boards can sustain transfers of approximately 200 MB per second. If your camera exceeds these limits, then data loss — broken images or timeouts — will occur.

To avoid data loss, first determine the bandwidth required by the camera in use, then ensure that both the board and the host can sustain the required throughput. Bandwidth requirements depend on the camera's pixel clock speed, the number of bytes per pixel, the number of taps, and the number of channels. To calculate bandwidth requirements:

```
pixel clock speed * taps * bytes/pixel * channels = total bandwidth
```

NOTE It is the pixel clock speed that affects bandwidth requirements, not the frame rate. If data loss is a problem, increasing the delay between frames will not help.

Cameras that output 10–16 bits per pixel require two bytes per pixel of bandwidth. Therefore, a camera that has a 40 MHz pixel clock, two taps, and 12 bits per pixel requires system bandwidth of:

```
40 * 2 * 2 = 160 MB per second
```

You can ease this constraint by running the camera in a less demanding mode — for example, you can often run two-tap cameras in single-tap mode, or 12-bit cameras in 8-bit mode.

If you're using a multichannel board with two or more cameras, this also can exceed bandwidth limits. For slower cameras, the bus bandwidth may be sufficient, but for two fast cameras, it may not be.

Also, ensure that the PCI DV board is not sharing the bus with other devices that reduce the available bus bandwidth. PCI video boards or other image acquisition boards — including more than one EDT board on the same bus — all reduce bandwidth, sometimes considerably.

Also, one 33 MHz board on a system will slow a 66 MHz bus down to 33 MHz for all devices. To avoid such problems, isolate such boards on separate, independent PCI buses, or choose a different bus (such as AGP or PCI Express Video).

Because most desktop systems come with single 33 MHz buses, they are usually suitable only for single cameras that output less than 90 MB per second. For faster cameras or multiple camera installations, choose a server or workstation system that has single or multiple 66 MHz or faster PCI or PCI X buses.

The PCIe DV C-Link is a four-lane PCI Express board that, because of bridging issues between PCI and PCI Express, is a 66 MHz board with a maximum bandwidth of approximately 360 MB per second. It must be set into a four-lane or greater PCIe Bus slot.

Problems Acquiring Images Using Our Applications

Make sure the camera device is on and is receiving power.

Check all interface cable connections. Turn off the camera, unplug the cable at both ends, reattach it, and then turn the camera back on.

Try a different cable, if available.

Try a different camera, if available.

Make sure the PCI DV software has been installed, and the driver is running and recognizes the board. To do so, at the command line in the EDT installation directory, enter:

```
pciload
```

If you see no information about a specific EDT board, this indicates a problem occurred during driver installation. If so, one at a time, try:

- cycling system power,
- moving the board to a different slot, and
- uninstalling and reinstalling the driver.

Make sure the board is installed in a PCI bus that meets its speed requirement. For best performance, install EDT boards in a 66 Mhz or faster PCI or PCI-X bus not shared with any other devices. For further information, see Bandwidth on page 2.

Problems With Your Applications

Application or system failures that occur when running user applications that use the EDT software libraries are most commonly the result of a mismatch between driver and library versions. When updating to a new device driver, always recompile and relink applications that use EDT software libraries.

Use the PDVDEBUG and EDTDEBUG environment variables to enable debug console output from the EDT software libraries. Both libraries are documented online at www.edt.com/api.

Applications that use the PCI DV Digital Video Library can set the PDVDEBUG environment variable to 1 or 2; a value of 1 turns on call trace information from most (though not all) PDV library routines; a value of 2 enables more verbose trace information. A value of 0 turns off debug output.

Similarly, applications that use the EDT DMA Library can set the EDTDEBUG environment variable to 1 or 2 for less or more verbose debug output from the DMA software library.

The EDT Message Handler Library provides generalized error- and message-handling for EDT software libraries and can be helpful for debugging your programs. See the EDT Message Handler Library in the EDT DMA Library for specific routines and usage.

Before calling for technical support, reproduce the problem with one of our applications, such as `take`, `simple_take`, or `PdvShow`, if possible. If you cannot reproduce the problem with an EDT application, compare your code with ours to see if you can spot the difficulty.

A simple example program that demonstrates the problem is also helpful.

Problems with Threads

The PCI DV driver and software libraries use threads. Make sure all PDV acquisition calls (for example, `pdv_start_image, pdv_wait_image, pdv_multibuf`) are in the same thread.

Some third-party software packages may not be thread-safe. If this causes problems, contact your third-party vendor to determine if a thread-safe version is available.

Updating the Firmware

When upgrading to a new device driver, or switching to a FPGA configuration file with special functionality, you may also need to reprogram the PCI interface flash PROM using `pciload`.

The following procedure applies to standard firmware only. If you are running a custom firmware file and need to update it, first get a custom firmware configuation file from EDT.

NOTE The presence of a newer version of the firmware with a new driver doesn't necessarily mean that the firmware must be updated; if a package contains a mandatory upgrade, it is prominently stated in the README file.

On UNIX systems, `pciload` is an application in the installation directory `/opt/EDTpdv`.

On Windows systems, double-click the Pdv Utilities icon to bring up a command shell in the installation directory `\EDT\Pdv`.

To see currently installed and recognized EDT boards and software, enter:

```
pciload
```

The program outputs the date and revision number of the firmware in the PROM.

To compare the PCI firmware in the package with the one already loaded on the board, enter:

```
pciload verify
```

The program compares the firmware in the PROM against the firmware file in the installation directory. If they match, there's no need to upgrade the firmware. If they differ, you'll see error messages. This does not necessarily indicate a problem; if your application is operating correctly, you may not need to upgrade the firmware.

If you wish to update the standard firmware, enter:

```
pciload update
```

To upgrade or switch to a custom firmware file, enter:

pciload *firmware_file*

...replacing the italicized argument with the filename of the desired upgraded firmware, up to the string "`_xv.bit`" (where *x* is a number, usually 3 or 5). For example, to load medium mode firmware `pdvcamlk_pir_3v.bit` and `pdvcamlk_pir_5v.bit` into the PCI DV C-Link, enter:

```
pciload pdvcamlk_pir
```

The board reloads the firmware from the PROM only during power-up, so after running `pciload`, the old firmware is in the PCI Xilinx® until the system has power-cycled.

NOTE Updating the firmware requires cycling power, not simply rebooting.

For a list of all `pciload` options, enter:

```
pciload help
```

Firmware on remote products such as the RCX or RCX C-Link may sometimes require an upgrade. The process is similar to that described above.

To update the RCX or RCX C-Link firmware using a PCI DV FOX:

1. Make sure the fiber optic cable is connected and the unit is powered on.
2. Check the README file on the distribution disk that came with your board for the name of the current firmware file.
3. Enter:

   ```
   rcxload -u unit number pathname.rcx
   ```

   ... replacing the path to the new firmware file name for *pathname*.

Corrupted Firmware

EDT boards are shipped with a protected and a programmable flash PROM boot sector. By default, the boards are shipped to enable reprogramming of the flash PROM in the field; however, most boards have jumpers you can reset to reboot from the protected sector, in case the firmware has somehow become corrupted. This enables you to get the system running again; you can then replace the jumper in its original position.

Most often, firmware corruption is the result of an interrupted load process, or an unanticipated interaction with the host computer. Therefore, the firmware file itself is most often uncorrupted; following the procedure below is sufficient to solve the problem. However, if the actual firmware file itself has become corrupted, first contact EDT for the current firmware you'll need to replace it; then follow this procedure.

To reboot from the protected sector:

1. If necessary, move the new firmware file to the directory in which you have installed the EDT driver software.
2. Power down the host and board.
3. To avoid later confusion, remove any other EDT boards from the host.
4. On the EDT board with corrupted firmware, move the jumper from its programmable to its protected setting, as indicated by the figure for the board you have. See Figures 13 through 20.
5. Turn power back on to the host and board.
6. Navigate to the directory in which you have installed the EDT driver software.
7. At the command prompt, enter:

   ```
   pciload
   ```

   The program outputs the date and revision number of the firmware in the PROM — in this case, the date and revision number shipping as of your purchase date. If no errors are reported, you have successfully booted from the protected sector.

8. Move the jumper back to its original position.
9. Enter:

   `pciload` *`firmware`*

   ...replacing firmware with the correct name, as indicated in Table 1.

   If the feedback shows no errors, the new firmware has been successfully installed, although it is not yet running.
10. Power the host and board down again.
11. Turn power back on to the system. The new firmware is now running.

Table 1. Arguments To `pciload` For Specific Boards

Board Name	`pciload` Command
PCI DVa	`pciload dva1`
PCI DVK	`pciload pdvk`
PMC DVK	`pciload pdvk`
PCI DV FOX	`pciload dvtlk4`
PCI DV C-Link (Rev. 52 or higher)	`pciload pdvcamlk2` (base mode) or `pdvcamlk_pir` (medium mode)
PCI DV C-Link (Rev. 51 or lower)	`pciload pdvcamlk` (base mode) or `pdvcamlk_pir` (medium mode)
PCIe DV C-Link	`pciload pedvcamlk`
PCIe DV C-Link as simulator	`pciload pedvclsim`
PMC DV C-Link	`pciload pdvcamlk` (base mode) or `pdvcamlk_pir` (medium mode)
cPCI DV C-Link	`pciload pdvcamlk` (base mode) or `pdvcamlk_pir` (medium mode)

Figure 13. PCI DVa Boot PROM Jumper Settings

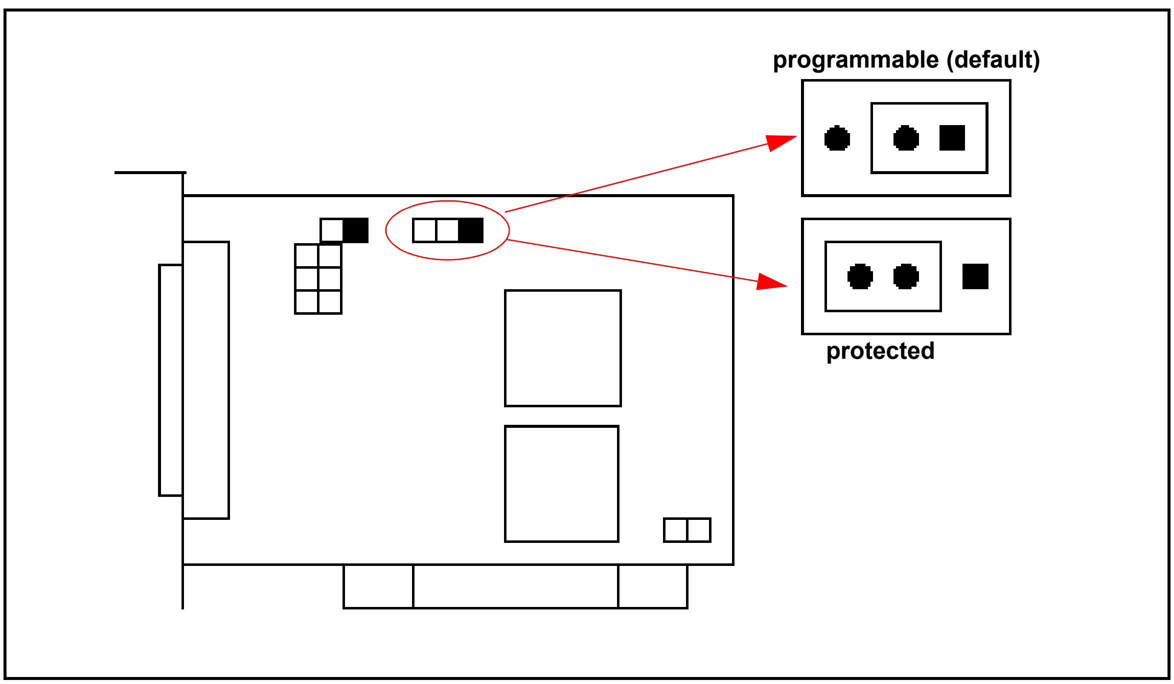


Figure 14. PCI DVK Boot PROM Jumper Settings

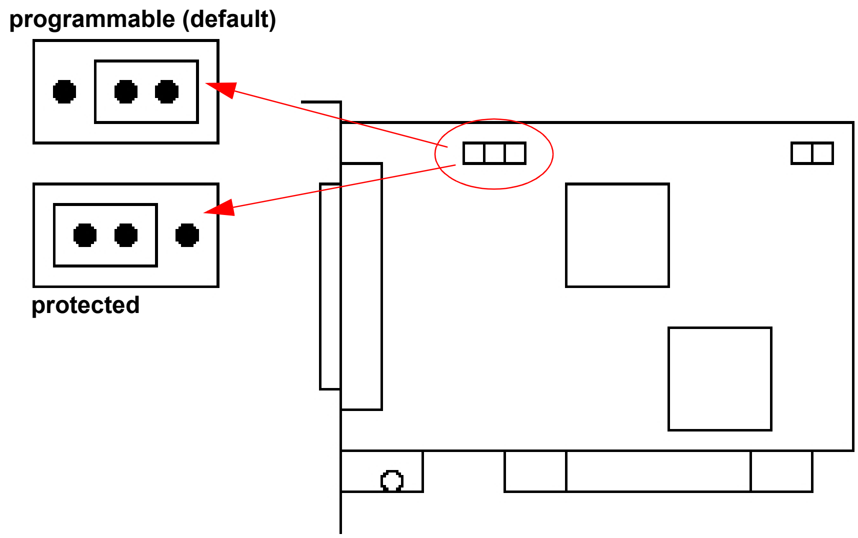


Figure 15. PMC DVK Boot PROM Jumper Settings

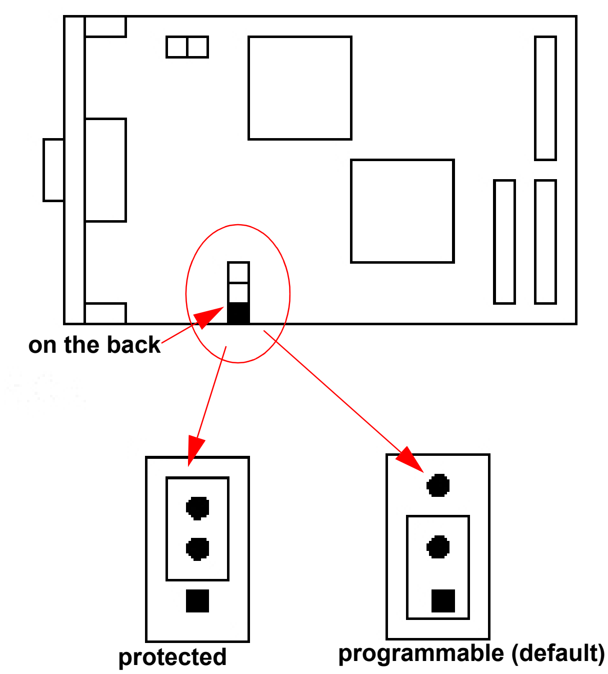


Figure 16. PCI DV FOX Boot PROM Jumper Settings

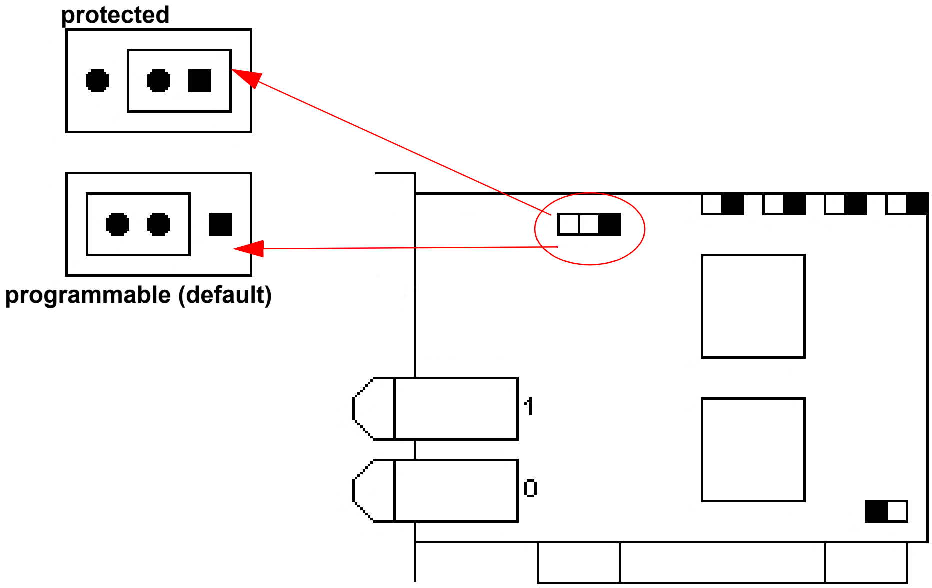


Figure 17. PCI DV C-Link Boot PROM Jumper Settings

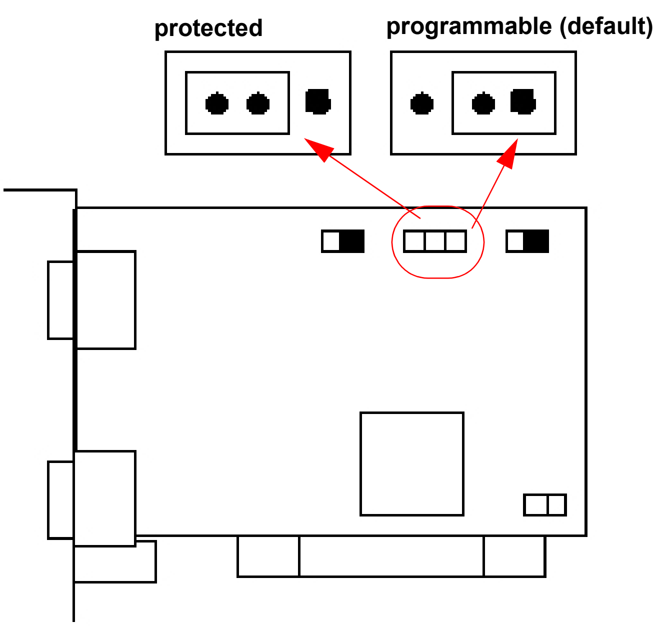


Figure 18. PCIe DV C-Link Boot PROM Jumper Settings

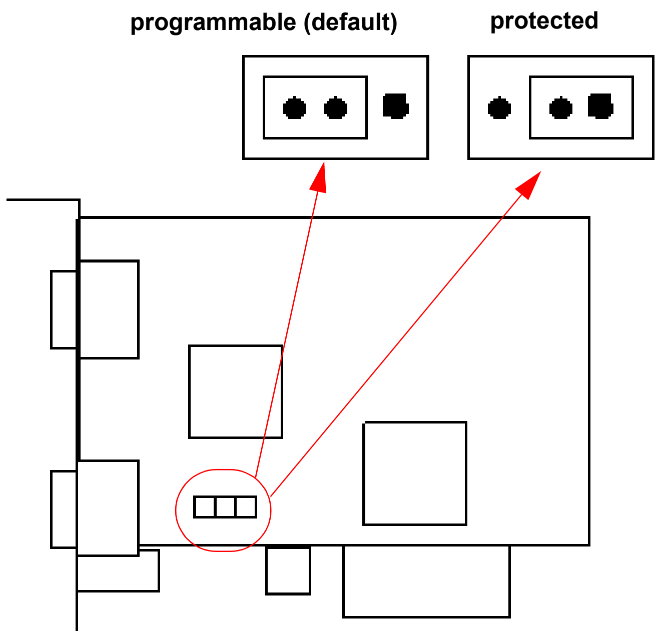


Figure 19. PMC DV C-Link Boot PROM Jumper Settings

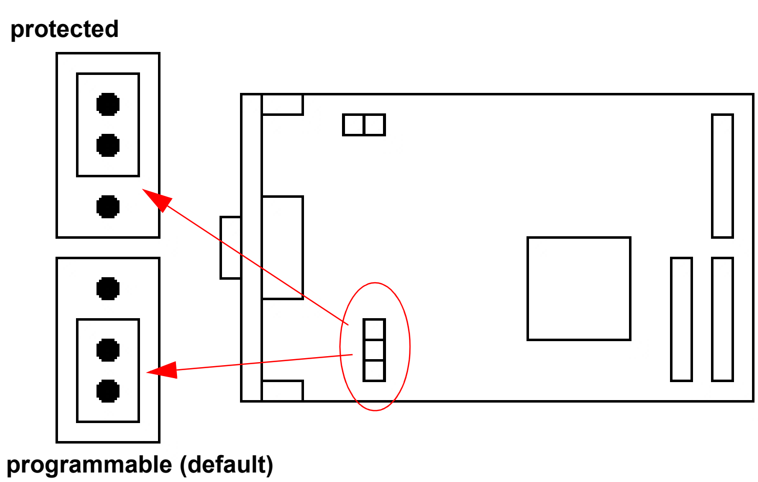


Figure 20. cPCI DV C-Link Boot PROM Jumper Settings