

PCI-AC51 AND PCIE-AC51 USER'S GUIDE

Form 1459-170209—February 2017

OPTO 22
Automation made simple.

43044 Business Park Drive • Temecula • CA 92590-3614
Phone: 800-321-OPTO (6786) or 951-695-3000
Fax: 800-832-OPTO (6786) or 951-695-2712
www.opto22.com

Product Support Services

800-TEK-OPTO (835-6786) or 951-695-3080
Fax: 951-695-3017
Email: support@opto22.com
Web: support.opto22.com

PCI-AC51 and PCIe-AC51 User's Guide

Form 1459-170209—February 2017

Copyright © 2003–2017 Opto 22.

All rights reserved.

Printed in the United States of America.

The information in this manual has been checked carefully and is believed to be accurate; however, Opto 22 assumes no responsibility for possible inaccuracies or omissions. Specifications are subject to change without notice.

Opto 22 warrants all of its products to be free from defects in material or workmanship for 30 months from the manufacturing date code. This warranty is limited to the original cost of the unit only and does not cover installation, labor, or any other contingent costs. Opto 22 I/O modules and solid-state relays with date codes of 1/96 or newer are guaranteed for life. This lifetime warranty excludes reed relay, SNAP serial communication modules, SNAP PID modules, and modules that contain mechanical contacts or switches. Opto 22 does not warrant any product, components, or parts not manufactured by Opto 22; for these items, the warranty from the original manufacturer applies. Refer to Opto 22 form 1042 for complete warranty information.

Wired+Wireless controllers and brains are licensed under one or more of the following patents: U.S. Patent No(s). 5282222, RE37802, 6963617; Canadian Patent No. 2064975; European Patent No. 1142245; French Patent No. 1142245; British Patent No. 1142245; Japanese Patent No. 2002535925A; German Patent No. 60011224.

Opto 22 FactoryFloor, *groov*, Optomux, and Pamux are registered trademarks of Opto 22. Generation 4, *groov* Server, ioControl, ioDisplay, ioManager, ioProject, ioUtilities, *mistic*, Nvio, Nvio.net Web Portal, OptoConnect, OptoControl, OptoDataLink, OptoDisplay, OptoEMU, OptoEMU Sensor, OptoEMU Server, OptoOPCServer, OptoScript, OptoServer, OptoTerminal, OptoUtilities, PAC Control, PAC Display, PAC Manager, PAC Project, PAC Project Basic, PAC Project Professional, SNAP Ethernet I/O, SNAP I/O, SNAP OEM I/O, SNAP PAC System, SNAP Simple I/O, SNAP Ultimate I/O, and Wired+Wireless are trademarks of Opto 22.

ActiveX, JScript, Microsoft, MS-DOS, VBScript, Visual Basic, Visual C++, Windows, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries. Linux is a registered trademark of Linus Torvalds. ARCNET is a registered trademark of Datapoint Corporation. Modbus is a registered trademark of Schneider Electric, licensed to the Modbus Organization, Inc. Wiegand is a registered trademark of Sensor Engineering Corporation. Allen-Bradley, CompactLogix, ControlLogix, MicroLogix, SLC, and RSLogix are either registered trademarks or trademarks of Rockwell Automation. CIP and EtherNet/IP are trademarks of ODVA. Raspberry Pi is a trademark of the Raspberry Pi Foundation.

groov includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org>)

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Opto 22

Automation Made Simple.

Table of Contents

Chapter 1: Introduction	1
Pamux Systems SDK	1
Prerequisites	1
What’s in this Guide	2
A Note about Banks and Brains	2
Banks and the SNAP-B6	3
For Help	3
Chapter 2: Installing the Card	5
Installing the PCI-AC51	5
Installing the PCIe-AC51	6
Chapter 3: Installing and Using the SDK	9
Installing the SDK	10
Using the PamScan Utility	11
Developing Custom Applications	14
Developing the Application	16
Special Considerations	16
Building an Application and Driver Installer	17
Porting Legacy Applications	18
Converting VB Legacy Programs	18
Changing Hardware Access	19
Updating Data Sizes	19
Function Reference Help File	19
Understanding the Pamux Bus Reset Line	20
The Reset Line	20
Terminology	20
“LOW” and “HIGH”	20
Brain Reset Configuration: “Active HIGH” and “Active LOW”	21
Appendix A: Requirements, Specifications, & LEDs	23
System Requirements	23
Specifications	23

LEDs.....	24
PCI-AC51	24
PCIe-AC51	25

Appendix B: Converting Applications to a Newer Card27

Applications that Used the OptoPMUX.dll for the AC5	27
Converting Applications that Use Inp() and Outp()	27

Appendix C: PCI-AC51 and PCIe-AC51 Technical Reference29

Architecture.....	30
Adapter Card Identification	30
Adapter Card Base Address Configuration	30
BAR1 Register Table	30
Register Description	31
Powerup Conditions.....	32
Controlling the Reset Line	32
Controlling the Reset Line Using the Pamux Driver	32
Controlling the Reset Line When Not Using the Pamux Driver	32
Default Power-Up State of Reset Line	33
Reset Configuration of Brains When Using PamScan	33
Testing	33
DIP Switches (PCI-AC51) and Jumpers (PCIe-AC51).....	33
High and Low Latency Modes	34
Performing Pamux Operations	34
Issuing Pamux Reset	34
Reading the State Machine Version	35
Reading the Switch/Jumper Values	35
Polled PCI-AC51 and PCIe-AC51 Operation	35
Performing a Pamux Bank Write	35
Performing a Pamux Bank Read	36
Notes for the SNAP-B6 (Digital Aspect)	36

1: Introduction

The PCI-AC51 and PCIe-AC51 adapter cards bring industry-standard Pamux[®] protocol to computers that use a Peripheral Component Interconnect (PCI) or PCI Express (PCIe) local bus.

- The PCI-AC51 is compatible with computers that feature a 5.0 or 3.3 volt PCI bus.
- The PCIe-AC51 is compatible with computers using the PCI Express bus.

These cards are an ideal choice when replacing an older PC with an ISA bus connected to an Opto 22 AC28 or PCI-AC28 adapter card. (The AC28 and PCI-AC28 adapter cards are not recommended for new designs. Instead, use the PCI-AC51 or the PCIe-AC51.)

With the PCI-AC51 or PCIe-AC51 adapter card, your computer can communicate with Opto 22 classic B4, B5, and B6 brain boards and with SNAP-B4 and SNAP-B6 brains.

- Each Pamux bus can access up to 32 remote brains.
- Each Pamux bus supports up to 512 points (equal to 64 banks).

For more information about banks and brains, see [page 2](#).

Pamux Systems SDK

For software developers building applications for the PCI-AC51 or PCIe-AC51, the Pamux Systems software development kit (SDK) is included free on the Opto 22 CD that comes with the cards. (You can also download the SDK from our website at www.opto22.com.)

The SDK contains sample applications with source code, Windows drivers, and documentation. In addition, the SDK provides drivers for migrating legacy applications to newer platforms.

The SDK also includes PamScan, a diagnostic tool for troubleshooting Pamux hardware. With PamScan, lets you read and write to analog and digital I/O, configure digital points as inputs or outputs, and configure analog points as specific point types.

For more about the Pamux Systems SDK, see [Chapter 3: Installing and Using the SDK](#).

Prerequisites

- This guide assumes that you are familiar with Opto 22 mounting racks and input/output modules.
- If you're planning to use the Pamux Systems SDK, this guide assumes that you are already familiar with Microsoft .NET and programming languages such as C#, Visual Basic, or Visual C++.

What's in this Guide

This guide assumes that you are familiar with Pamux and the brains, racks, and input/output modules used with Pamux. For more information about Pamux, see form 0726, [Pamux User's Guide](#).

If you are a software developer writing programs for the PCI-AC51 or PCIe-AC51 using the Pamux Systems SDK, this guide assumes that you are already familiar with Microsoft .NET and programming languages such as C#, Visual Basic, or Visual C++.

This guide includes the following chapters and appendices:

Chapter 1: Introduction introduces the card and provides Product Support information.

Chapter 2: Installing the Card describes how to install the adapter card.

Chapter 3: Installing and Using the SDK describes how to set up the SDK and use it to write custom applications for the PCI-AC51 and PCIe-AC51 adapter cards on Windows platforms.

Appendix A: Requirements, Specifications, & LEDs provides system requirements and specifications for the PCI-AC51 and PCIe-AC51 adapter cards. It also shows the location and function of the LEDs on the cards.

Appendix B: Converting Applications to a Newer Card describes how to convert legacy applications that used the AC28 and PCI-AC28 adapter cards.

Appendix C: PCI-AC51 and PCIe-AC51 Technical Reference provides technical information for authoring a device driver for an unsupported operating system or a Windows kernel-mode driver.

A Note about Banks and Brains

To use the PCI-AC51 and PCIe-AC51 with a Pamux bus—and certainly to write a program that interfaces the two devices—it helps to understand *banks*.

Each PCI-AC51 and PCIe-AC51 connects to one Pamux bus, and each Pamux bus consists of up to 512 I/O points. To easily address and reference so many points, Opto 22 brains and software logically group the I/O into banks, which are sets of eight consecutive points. Custom software applications that read a bank of points are typically more efficient than programs that read each point individually.

A Pamux bus with 512 I/O points consists of 64 banks, numbered 0 through 63. To access a point, you reference the bank number and then the point within the bank. (Remember that the first bank is number 0 and the first point in each bank is number 0.)

Technically, any point in bank 1 or greater can be referenced in multiple ways. For example, the sixth (physical) point in the second (logical) bank can be referenced in both of these ways:

- Bank 1, point 5
- Bank 0, point 13

If you are a programmer, it can help you in the future to clearly document in your code the conventions you used to reference points.

The following table shows a list of Opto 22 brains and the maximum number of points (and equivalent number of banks) they can access via a Pamux bus.

The Opto 22...	...can access	...which is the same as
B4 digital brain	32 digital points	4 banks
B5 digital brain	16 digital points	2 banks
B6 analog brain	16 analog points	2 banks
SNAP-B4 digital brain	32 digital points	4 banks

Banks and the SNAP-B6

When it comes to identifying points in banks, the SNAP-B6 high-speed analog/digital Pamux Protocol brain is a special case. The first eight modules on a 16-position B-series SNAP PAC rack can be either digital or analog. The last eight modules can only be analog. Each digital module contains four points (equal to one-half of a bank), and each analog module has either one or two points depending on the module. Consequently, the number of banks and the position of points that a SNAP-B6 can access is dependent entirely on the rack's configuration.

For Help

If you have problems installing or programming the PCI-AC51 or PCIe-AC51 adapter card and cannot find the help you need in this guide, contact Opto 22 Product Support.

Phone: 800-TEK-OPTO (800-835-6786)
toll-free in the U.S. and Canada)
951-695-3080
Monday through Friday,
7 a.m. to 5 p.m. Pacific Time

NOTE: Email messages and phone calls to Opto 22 Product Support are grouped together and answered in the order received.

Fax: 951-695-3017

Email: support@opto22.com

Opto 22 website: www.opto22.com

When calling for technical support, you can help us help you *faster* if you can provide the following information to the Product Support engineer:

- Opto 22 hardware part numbers or models that you're working with
- Software version
- Specific error messages you saw
- Version of your computer's operating system

2: Installing the Card

This chapter describes how to install the PCI-AC51 or PCIe-AC51 adapter card. You can use any available expansion slot on the computer.

You may add multiple adapter cards for convenience, but note that multiple cards do not increase Pamux throughput. The number of Pamux accesses per computer is constant.

In this chapter:

Installing the PCI-AC51	page 5
Installing the PCIe-AC51	page 6

Installing the PCI-AC51

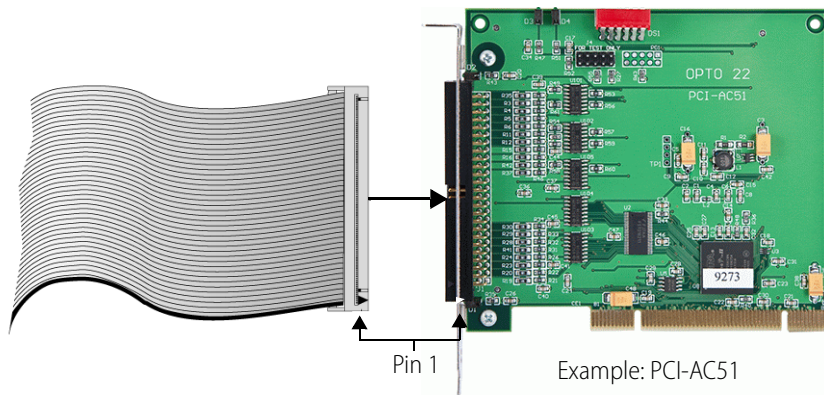
You can install the PCI-AC51 adapter card into any PCI expansion slot on your computer.

Note the power requirements listed in the Specifications table on [page 23](#).

CAUTION: Do not scratch this card or other cards in the computer, as scratching may irreversibly damage the card or other devices.

1. Turn off the computer. Remove the power cord and the computer's cover.
The power cord must be removed, as a spike may cause the computer to boot.
2. Before handling the card, discharge static electricity by touching the computer's metal chassis.
3. Install the card in the expansion slot. Verify that the card is properly seated in the motherboard socket. Secure the card to the chassis.

4. Attach the cable to the adapter card.



5. Connect the other end of the cable to the I/O mounting rack. See form 0726, the [Pamux User's Guide](#), for more information.
6. Reinstall the power cord. If you wish, leave the computer cover off temporarily to see the card's LEDs.
7. Turn on the computer. The first time you start the computer after you have installed the card, the BIOS automatically performs configuration.

If a "Found New Hardware" message appears, simply click Cancel. The Pamux Systems SDK setup program will install the device driver automatically.

If you have been using an AC28 adapter card for the ISA bus, you will need to make some changes to your legacy applications because of the new adapter card. See [Appendix B: Converting Applications to a Newer Card](#) for more information.

Installing the PCIe-AC51

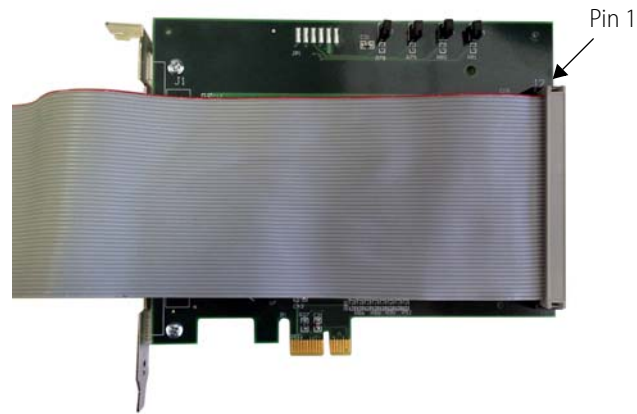
You can install the PCIe-AC51 adapter card into any PCIe expansion slot on your computer.

Note the power requirements listed in the Specifications table on [page 23](#).

CAUTION: Do not scratch this card or other cards in the computer, as scratching may irreversibly damage the card or other devices.

1. Turn off the computer. Remove the power cord and the computer's cover.
The power cord must be removed, as a spike may cause the computer to boot.
2. Before handling the card, discharge static electricity by touching the computer's metal chassis.
3. Starting with the connector end that allows the cable to lie flat, push the ribbon cable through one of the slot openings in the computer.

4. Run the cable across the top of the adapter card (so it covers the card) and attach the cable to the card, as shown.



5. Install the card in the expansion slot. Verify that the card is properly seated in the motherboard socket. Secure the card to the chassis.
6. Connect the other end of the cable to the I/O mounting rack. See form 0726, [Pamux User's Guide](#), for more information.
7. Reinstall the power cord. If you wish, leave the computer cover off temporarily to see the card's LEDs.
8. Turn on the computer. The BIOS automatically performs configuration the first time you boot up the computer with the card installed.

If a "Found New Hardware" message appears, simply click Cancel. The Pamux Systems SDK setup program will install the device driver automatically.

If you have been using an AC28 adapter card for the ISA bus, you will need to make some changes to your legacy applications because of the new adapter card. See [Appendix B: Converting Applications to a Newer Card](#) for more information.

3: Installing and Using the SDK

The Pamux Systems software development kit (SDK) helps developers build custom applications for the PCI-AC51 and PCIe-AC51.

The SDK is available on the CD that comes with the adapter card¹, and contains sample applications with source code, Windows drivers, and user documentation. It works with .NET platform languages including **C#**[®] and **VB.NET**[®], and supports both 64-bit and 32-bit versions of:

- Windows[®] 10 Professional
- Windows 8.1 Professional
- Windows 7 Professional²

NOTE: If you're using an unsupported operating system, you'll need to write your own driver. For details, see "Developing Custom Applications" on page 14 and Appendix C: PCI-AC51 and PCIe-AC51 Technical Reference.

To help you get started, the SDK includes a fully functioning Visual C# solution (.sln file) that interfaces with the PCI-AC51 and PCIe-AC51. You can use this solution as an example for building your own custom applications. When you install the SDK, the solution is copied to the C:\Program Files\Opto22\PC-PAMUX-SDK\Examples\PamScan folder.

For help troubleshooting your custom application, see "Error Codes" on page 16.

To update existing applications to run on Windows 10, Windows 8.1, or Windows 7, see "Porting Legacy Applications" on page 18.

In this chapter:

Installing the SDK	page 10
Using the PamScan Utility	page 11
Developing Custom Applications	page 14
Porting Legacy Applications.....	page 18
Function Reference Help File.....	page 19
Understanding the Pamux Bus Reset Line	page 20

¹If you don't have the CD, you can download the SDK from our website at www.opto22.com, or you can request the CD by calling Opto 22 at 800-321-6786 (toll-free in the U.S. and Canada) or 951-695-3000.

²For Windows 7 to properly identify the SDK's digital signatures and files, Microsoft Security Advisory 3123479 update (or higher) must be installed. To download the update, see <https://support.microsoft.com/en-us/kb/3123479>.

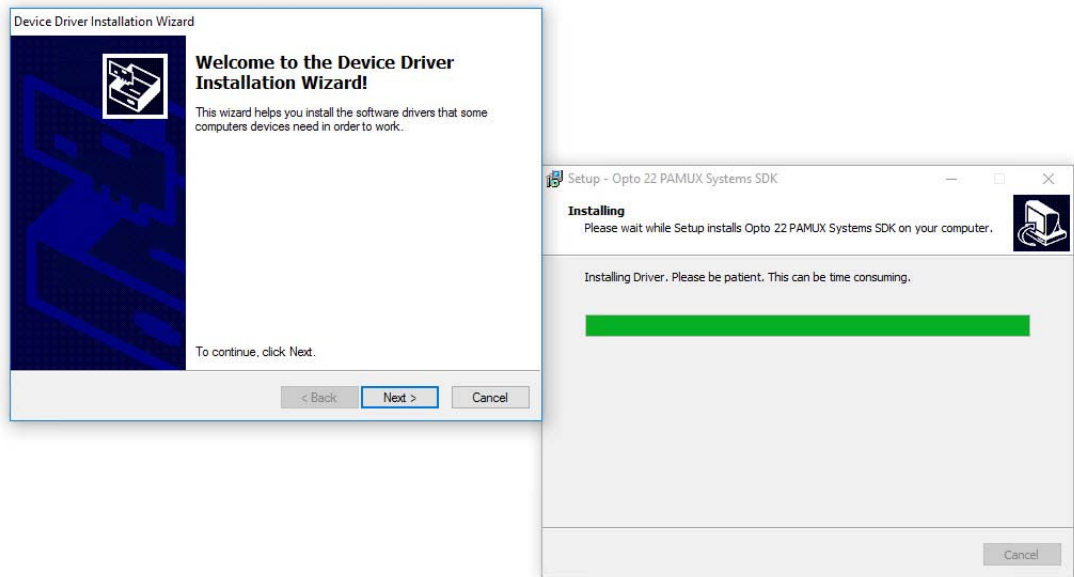
Installing the SDK

TIP: To ensure reliable control operations and prevent applications from closing unexpectedly, disable Sleep and Hibernation modes (by setting Windows Power Options to "Never") on computers that test or run applications built with the SDK. For more information, see <https://support.microsoft.com/en-us/help/13770/windows-shut-down-sleep-hibernate-your-pc>.

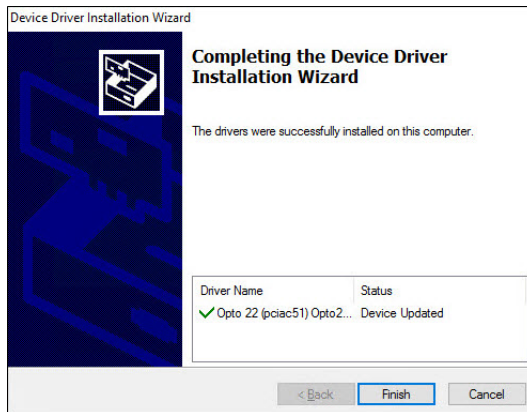
1. To start the Pamux Systems SDK setup, insert the CD in your CD-ROM drive (or, if you've downloaded the SDK from our website, double-click the installer).

NOTE: If Windows Security dialog boxes appear, click the appropriate buttons (for example, Yes, Open, Run, or Install) to continue.

2. Follow the Setup Wizard's prompts to accept the License Agreement and select the location where the SDK will be installed. Note that if the SDK is already installed, the Setup Wizard prompts you to uninstall it before continuing.



3. On the Device Driver Installation Wizard screen, click Next to install the driver. After the driver is installed, this screen appears:



4. Click Finish to complete the driver installation, and then click Finish again to complete the SDK installation and launch PamScan.exe.


Using the PamScan Utility

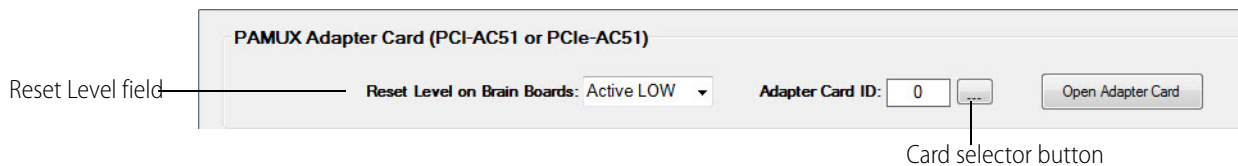
PamScan is a diagnostic utility that allows you to access any four banks of a Pamux bus. (For an explanation of banks, see [“A Note about Banks and Brains”](#) on page 2.)

Using PamScan, you can configure points as either input or output, and test your system by turning digital points on and off, displaying analog point values, and changing output values.

For software developers, PamScan can be used as an example for developing custom code. PamScan is written in Visual C# and the source code is located in the C:\Program Files\Opto22\PC-PAMUX-SDK\Examples\PamScan folder.


To use PamScan:

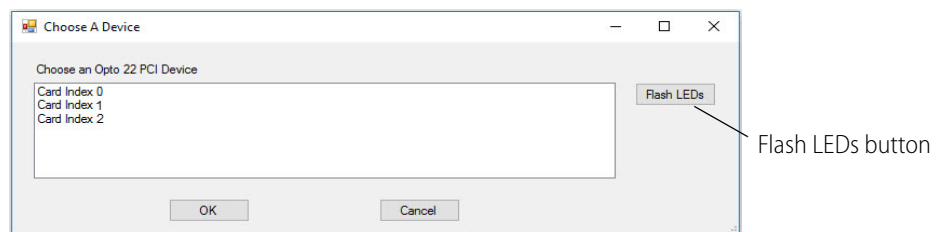
1. If PamScan is not already open, start it by pressing the Windows Start key , typing `pamscan.exe` and then pressing the Enter key.
2. On the PamScan main window, make sure the Reset Level field matches the reset level configured on the Pamux brain.



3. In the Adapter Card ID field, use the default value (0) if you have only one PCI-AC51 or PCIe-AC51 card installed.

If you have more than one card installed:

- a. Click the card selector button .
- b. In the Choose A Device dialog box, select a card.

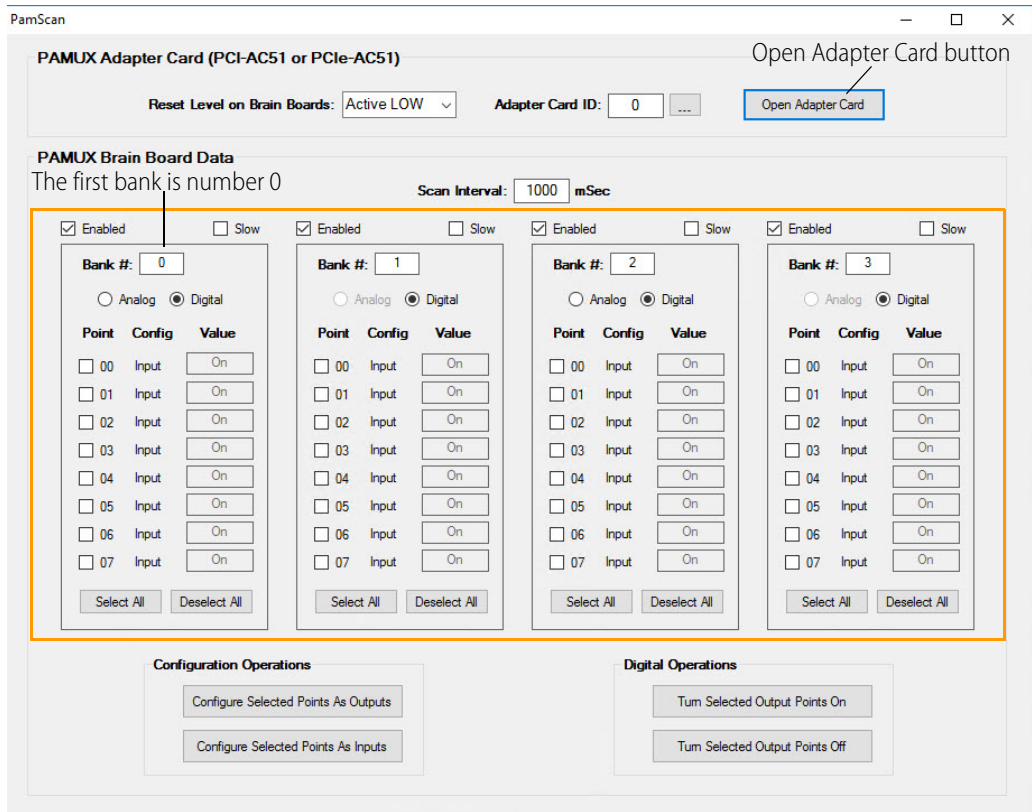


- c. (Optional) To identify which card you selected, you can flash its LEDs:
 - Open your computer’s case to see the installed adapter cards.
 - Click Flash LEDs.

The LEDs of the selected card will flash five times. If the LEDs don’t flash on the card you are looking at, you may have selected the wrong card. Try selecting a different card from the list, and then click Flash LEDs again to make sure you’ve selected the right card.
- d. When you’re done, click OK to close the dialog box.

- To view the banks on the card, click Open Adapter Card.

When you click Open Adapter Card, the banks on the selected card are displayed.

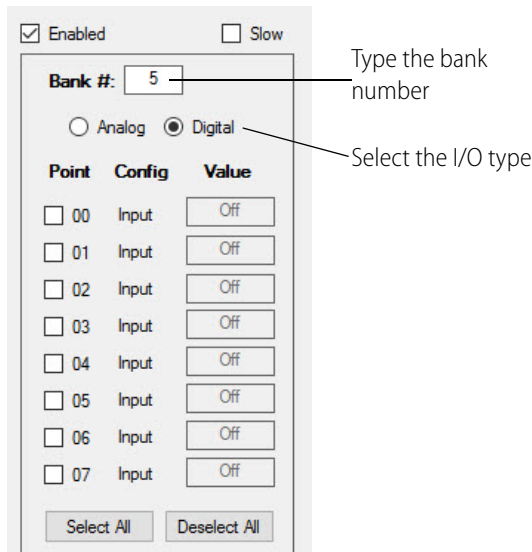


Use this portion of the screen to view the banks as you test them.

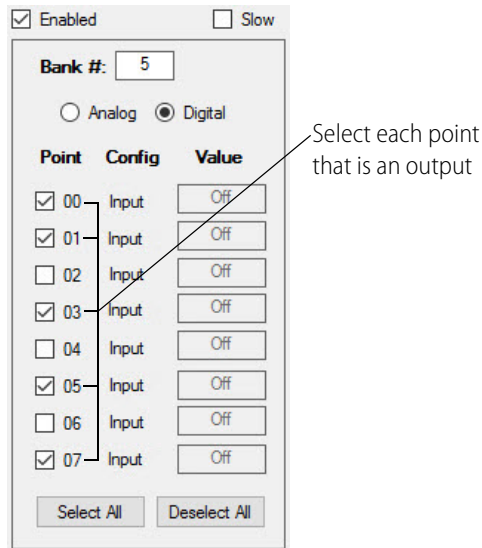
Each bank has eight I/O points and a unique bank number.

For example, the banks shown here are numbered 0, 1, 2, and 3 on the Pamux bus.

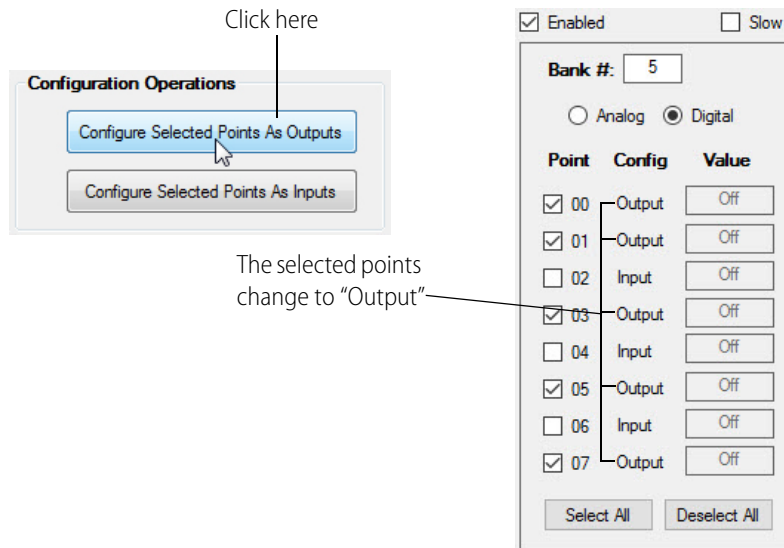
- For the card you've selected, identify a bank that has I/O installed.
- Then, in one of PamScan's Bank # fields, type the bank's number, and select the I/O type (Analog or Digital).
- In this image, 5 is the bank number and its points are digital I/O.



6. For *output* modules, select each point that is an output...

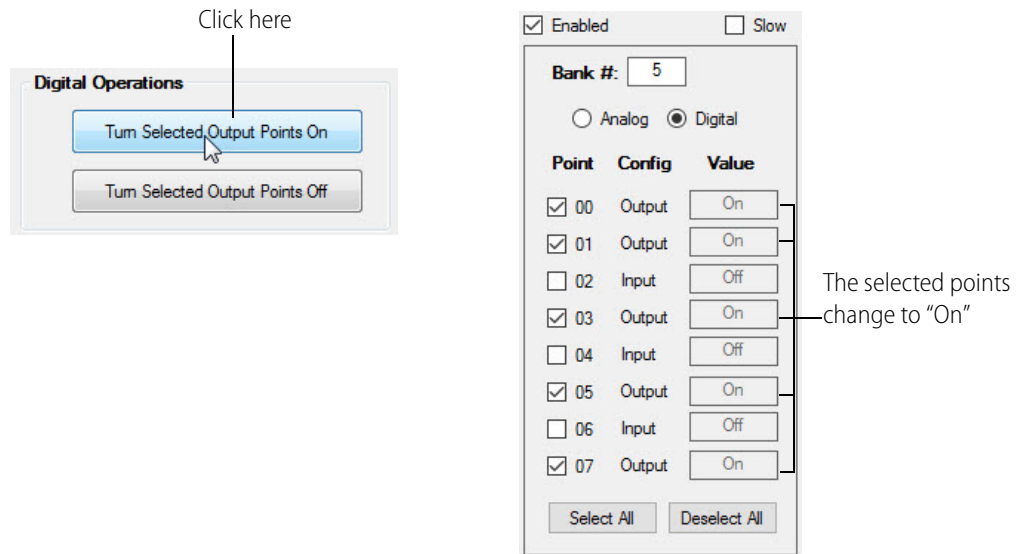


...and then click Configure Selected Points As Outputs.
The selected points are now configured as output points.



NOTE: When a Pamux brain powers up, all points are automatically configured as inputs. (However, if you accidentally configured an input point as an output, you will have to reconfigure it as an input.)

- To test the state of an output, click Turn Selected Output Points On. Notice that the selected output points change from “Off” to “On.”

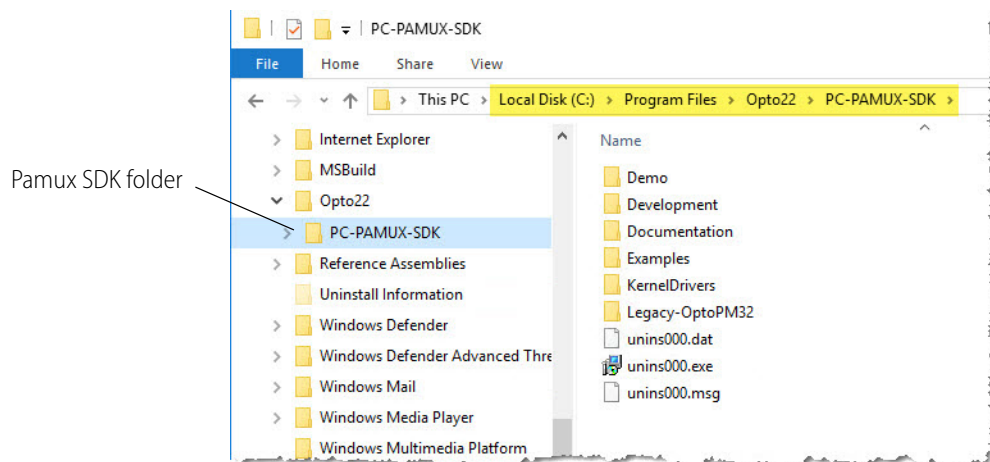


- Continue to experiment by selecting points, configuring points as inputs or outputs, and turning points on and off until you have finished testing.

Developing Custom Applications

NOTE: To upgrade existing applications to work with Windows 10, Windows 8.1, and Windows 7 operating systems, see [page 18](#).

When you install the SDK, new folders are created in the C:\Program Files\Opto22\PC-PAMUX-SDK directory. These folders contain the sample code, documentation, demo utility, and, most importantly, software stacks and kernel driver installers that you need for building custom applications.



The SDK contains several .NET managed software stacks. Applications built with any of the software stacks will work with both the PCI-AC51 and PCIe-AC51 adapter cards.

Although the files in each stack have the same filenames, each stack is different. When you create a custom application, you must **use the stack that matches your development environment and the target platform.**

If you're developing on a computer with this kind of operating system:	...and your .NET solution is configured to run on this platform:		
	Any CPU	x86	x64
32-bit	Use a 32-bit SDK software stack.	Use a 32-bit SDK software stack.	Not applicable.*
64-bit	Use a 64-bit SDK software stack.	Use a 32-bit SDK software stack.	Use a 64-bit SDK software stack.

*This option isn't applicable, because if you're coding on a 32-bit machine, you won't be able to run or debug code targeted to a 64-bit platform.

Software stacks are located in the C:\Program Files\Opto22\PC-PAMUX-SDK\Development directory, and folder names are designed to help you find the stack you need.

For example, if:

- Your development computer is a 64-bit machine
- You're using .NET 4.0
- Your .NET solution is configured for a 32-bit platform

...then use the software stack located in C:\Program Files\Opto22\PC-PAMUX-SDK\Development**32Bit\TargetedToNETFramework4AndAbove.**

1. *Building an app for a 32-bit platform?*
Open the **32Bit** folder

2. *Developing with .NET 4.0 or higher?*
Choose a stack from the **TargetedToNETFramework4AndAbove** folder

3. This is the stack to use

Developing the Application

TIP: To ensure reliable control operations and prevent applications from closing unexpectedly, disable Sleep and Hibernation modes (by setting Windows Power Options to “Never”) on computers that test or run applications built with the SDK. For more information, see <https://support.microsoft.com/en-us/help/13770/windows-shut-down-sleep-hibernate-your-pc>.

For a functional reference of the classes and methods in the SDK, see [page 18](#).

To use the SDK in a .NET application:

1. Copy the appropriate software stack to the folder where the application’s executable (.exe) file resides.¹
2. Add the PCIAC51.dll from the chosen software stack as a reference to your solution.
To identify which software stack to use, see [page 15](#).
3. Add this .NET using statement to your solution:
`using Opto22.Adapters.Pamux`
4. Instantiate the PCIAC51_CardList object by accessing the public property TheCardList.
Example: `public PCIAC51_CardList pciac51cards;
pci51cards = PCIAC51_CardList.TheCardList;`
5. Call GetCard() on the PCIAC51_CardList object.
Example: `pci51cards.GetCard(0,
PCIAC51_Card.eResetLevel.BrainSetForResetActiveLow, True);`
6. Call WriteDigitalConfig() and WriteAnalogConfig() to configure input and output points.
7. Create a loop that reads or writes points.
8. When the loop is complete, call ReleaseCard() to close the card properly.

Special Considerations

When developing custom applications, here are some special considerations to keep in mind:

Error Codes. If one of the function calls returns false, it’s an indication that an error or warning has occurred. To inspect the message, use the **lastError** property on either the PCIAC51_CardList object or the PCIAC51_Card object, depending on what function was called.

Error and warning messages are also logged in the Application area of the Windows Event Log. The messages are grouped under the application name, Opto 22 PCI-AC51.

Multi-threaded Applications. The Pamux Systems SDK permits only a single handle to a card. If your application uses multiple threads, you should implement a mutex on the handle to avoid thread collision.

Also, multiple applications cannot access the adapter card simultaneously. If you want multiple applications to access the card, you’ll need to use or build a program to control and synchronize access.

¹ In Visual Studio®, the bin folder is the default location for a solution’s .exe files. For details on how .NET applications locate DLLs, see the Microsoft Dev Center, [https://msdn.microsoft.com/en-us/library/windows/desktop/ms682586\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms682586(v=vs.85).aspx)

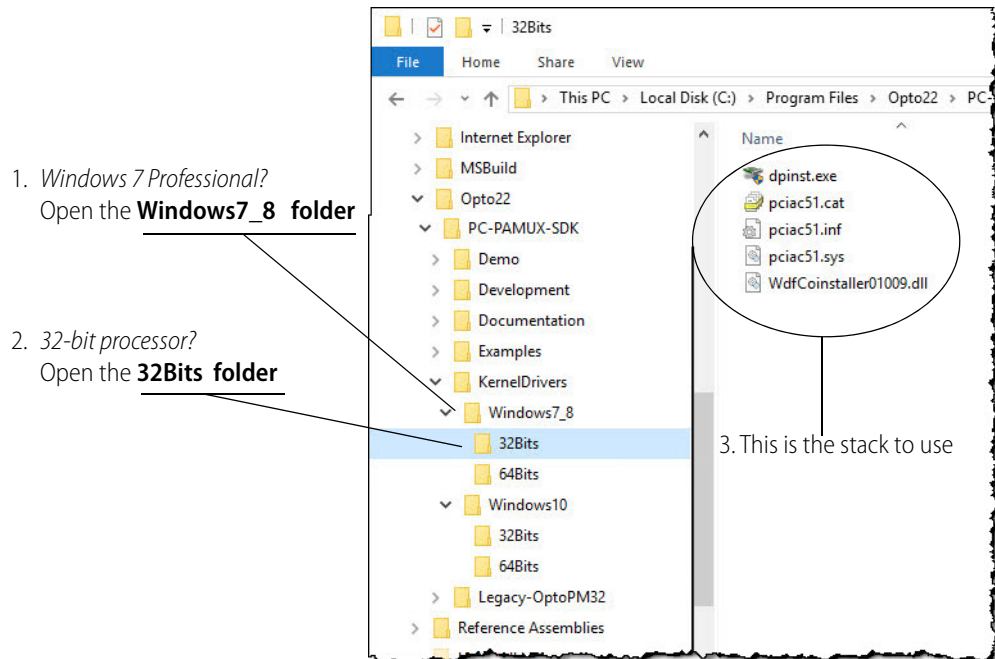
PCI Express Slot Width. While Opto 22’s PCI-Express adapter cards are a 1x form factor, they may be installed in wider PCI-Express sockets, like 4x, 8x, and 16x. For expansion slot availability, see the user’s manuals for your computer or motherboard.

Building an Application and Driver Installer

The SDK’s adapter card installer places both 32-bit and 64-bit driver and library components in the application installation directory. This allows you to build and distribute your application to specific platforms.

To install the adapter card kernel mode driver, the SDK includes 32-bit and 64-bit versions of Microsoft’s Driver Package Installer (dpinst.exe).

If your operating system is:	...and your processor is:	...Use the Driver Package Installer in this folder:
Windows 7 Professional or Windows 8.1 Professional	32-bit	C:\Program Files\Opto22\PC-PAMUX-SDK\KernelDrivers\Windows7_8\32Bits
Windows 10 Professional	32-bit	C:\Program Files\Opto22\PC-PAMUX-SDK\KernelDrivers\Windows10\32Bits
Windows 7 Professional or Windows 8.1 Professional	64-bit	C:\Program Files\Opto22\PC-PAMUX-SDK\KernelDrivers\Windows7_8\64Bits
Windows 10 Professional	64-bit	C:\Program Files\Opto22\PC-PAMUX-SDK\KernelDrivers\Windows10\64Bits



To install the driver, run dpinst.exe in the folder that contains the driver’s installation components.

For more information about the Driver Package Installer, see Microsoft’s Hardware Dev Center website at <https://developer.microsoft.com/en-us/windows/hardware>.

Opto 22 encourages OEMs that distribute software to wrap their applications and the necessary DLLs in an installer.

Porting Legacy Applications

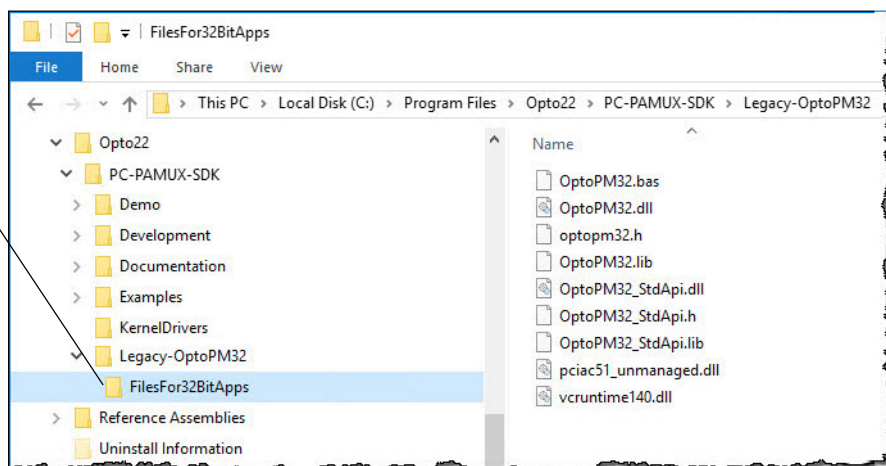
Programs developed with older versions of the Pamux Systems SDK will not work with Windows 10, Windows 8.1, or Windows 7 due to an incompatibility at the driver level of these operating systems.

To update existing applications to run on these newer operating systems, you must use the Legacy-OptoPM32 unmanaged software stack. When you install the SDK, these files are copied to the C:\Program Files\Opto22\PC-PAMUX-SDK\Legacy-OptoPM32\FilesFor32BitApps folder.

The files in this stack are 32-bit only and consist of these components:

- OptoPM32_StdApi.dll
NOTE: Include this file only for applications that require a STDAPI calling convention with OptoPM32.dll (for example, Delphi).
- OptoPM32.dll
- pciac51_unmanaged.dll
- vc140runtime.dll

Copy the files in this folder to the folder that contains the legacy app's .exe file



To upgrade a legacy program to work with Windows 10, Windows 8.1, and Windows 7:

1. Copy the legacy program files to the target computer.
2. Install the Pamux Systems SDK on the target computer. (For details, see [page 10](#).)
3. Copy the files from C:\Program Files\Opto22\PC-PAMUX-SDK\Legacy-OptoPM32\FilesFor32BitApps to the folder where the legacy application's executable (.exe) file resides.

To recompile the legacy program on the target operating system:

- Include this file with the legacy program files: OptoPM32.h
- Link this file to legacy program: OptoPM32.lib

To run the legacy program without recompiling:

- Include these files with the legacy program files: OptoPM32.dll
pci51_unmanaged.dll

Converting VB Legacy Programs

When converting Visual Basic (VB) legacy programs, be sure to register OCX files.

For example, to register COMCT232.OCX:

1. Open a command prompt with “run as administrator” permissions.
2. Navigate to the directory where COMCT232.OCX resides. Type `regsvr32 COMCT232.OCX` and then press Enter.

If the registration is successful, a dialog box will notify you.

Changing Hardware Access

Modern operating systems use interface drivers to prevent direct access to hardware.¹ Interface drivers access hardware via a handle. The handle also provides a simple reference method for the application. When using the Pamux Systems SDK to migrate a legacy application, the first operation your application must perform is to acquire a handle to the hardware from the operating system. At the end of the operation, the application must close the handle to release hardware access back to the operating system. In other words, when updating legacy applications, you must replace the direct hardware access logic, and instead, use a handle:

1. Open an adapter handle.
2. Configure the adapter.
3. Read and write to the adapter.
4. Close the adapter handle.

Updating Data Sizes

When updating a legacy application to run on a newer version of Windows, you must make sure that data sizes in the legacy code are compatible with the driver on the current version of the Pamux Systems SDK.

The Pamux Systems SDK supports both 64-bit and 32-bit versions of:

- Windows 10 Professional
- Windows 8.1 Professional
- Windows 7 Professional

Function Reference Help File

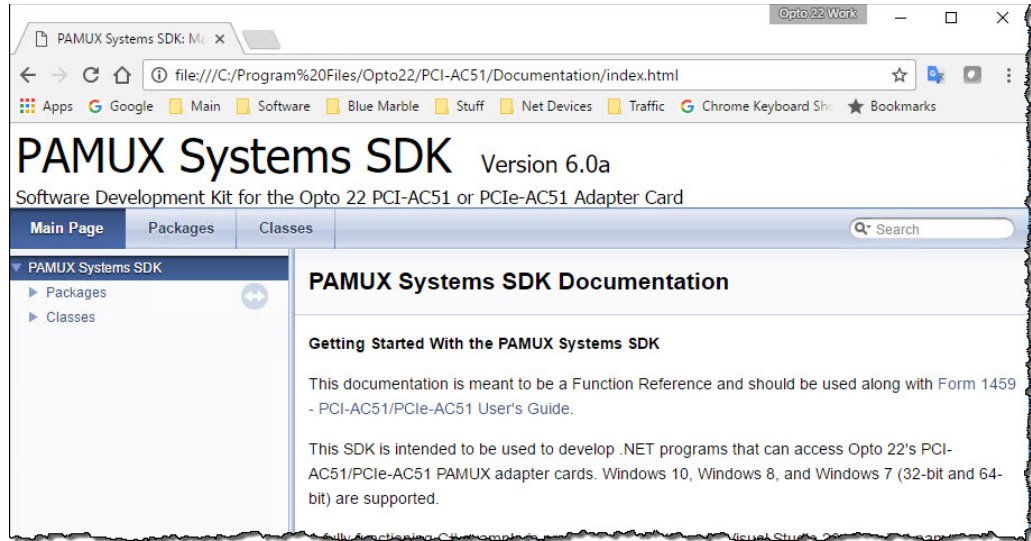
The Function Reference help file is installed on your PC when you install the SDK.

To start the Function Reference, open a browser and type or paste this URL:

`file:///C:/Program%20Files/Opto22/PC-PAMUX-SDK/Documentation/html/index.html`

¹ In legacy applications that use the `inp`, `outp`, or `PortIO` functions, hardware is accessed directly and in two steps: first, you configure the adapter, then you read and write to the adapter. The direct access method relies on the assumption that no other process will attempt to access the hardware. This can cause resource conflicts, which is part of the reason Microsoft changed the driver architecture in Windows 7 (and higher).

If your browser blocks the ActiveX controls, make sure to unblock them so that you can see the navigation pane.



Understanding the Pamux Bus Reset Line

This section explains the terminology and functioning of the Pamux bus reset line and its interaction between the PCI-AC51 or PCIe-AC51 card and the Pamux brains.

The Reset Line

The purpose of the reset line on the Pamux bus is to provide a way to reset all the brains in the system to their default powerup state. Digital brains will turn OFF the digital output modules and configure them as inputs. Analog brains will set all analog outputs to zero scale and configure them as inputs.

The reset line is pin 49 of the Pamux ribbon cable. The state of the reset line can be verified by measuring the voltage between pin 49 and any even-numbered pin. (Even-numbered pins are connected to logic ground.) For more information about the reset line and other pins on the Pamux bus, see Opto 22's form 0726, the [Pamux User's Guide](#).

Terminology

“LOW” and “HIGH”

The reset line is considered to be “LOW” when the voltage on the reset line is LOW (~ 0 VDC).

The reset line is considered to be “HIGH” when the voltage on the reset line is HIGH (~ 3.1 to 5.0 VDC).

Brain Reset Configuration: “Active HIGH” and “Active LOW”

Each Pamux brain has a jumper to configure whether the brain will reset when the reset line is HIGH or LOW. This is referred to as reset “Active HIGH” or reset “Active LOW”.

If the brains are jumpered for reset “Active HIGH”, then the system will be “in reset” (the brains will be reset) when the reset line is HIGH. The brains will function normally (not be reset) when the reset line is LOW.

Similarly, if the brains are jumpered for reset “Active LOW”, then the system will be “in reset” (the brains will be reset) when the reset line is LOW. The brains will function normally (not be reset) when the reset line is HIGH.

You must determine whether you want your Pamux system to operate with reset Active HIGH or reset Active LOW. Then consistently set the reset jumper on each brain in the system to match your choice; all brains must use the same reset configuration. One consideration is how you want your system to function if the computer loses power. Without power to the computer, the PCI-AC51 or PCIe-AC51 cannot drive the reset line high, so by default it will be low. You would want to analyze your system needs to determine what is necessary for your system to automatically go into failsafe mode.

A: Requirements, Specifications, & LEDs

This appendix provides system requirements, card specifications, and information on card LEDs.

System Requirements

Here's what you need to install and run the software provided for the PCI-AC51 and PCIe-AC51:

- A computer with a 64- or 32-bit processor, 1 GHz or higher. The most recent version of BIOS must be installed.
- One of the following operating systems:
 - Microsoft Windows 10 Professional (64-bit or 32-bit)
 - Microsoft Windows 8.1 Professional (64-bit or 32-bit)
 - Microsoft Windows 7 Professional (64-bit or 32-bit)

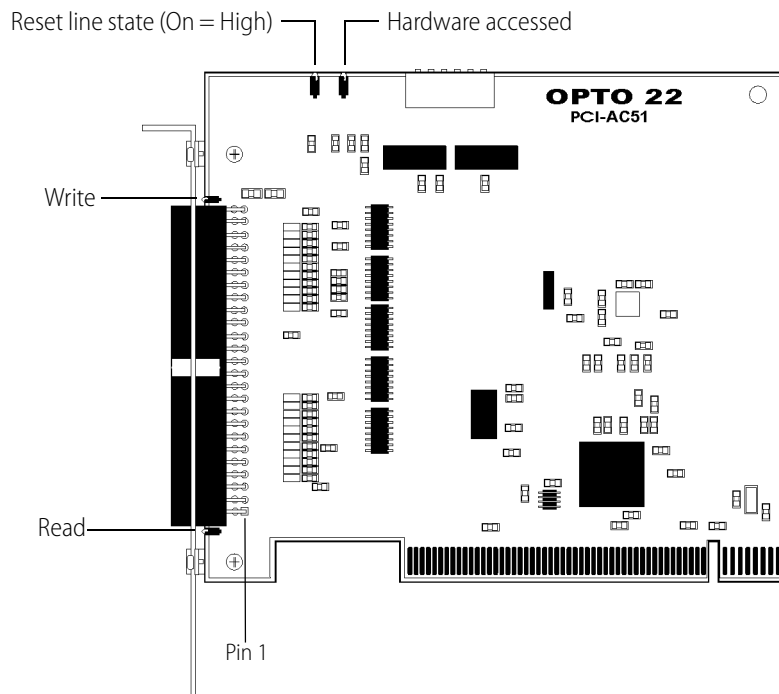
Specifications

	PCI-AC51	PCIe-AC51
Power requirements (provided by PCI/PCIe bus)	5 VDC @ 500 mA 3.3 VDC @ 300 mA	12 VDC @ 50 mA 3.3 VDC @ 500 mA
Computer compatibility	32- or 64-bit PCI bus	PCI Express x1 or higher
SDK compatibility	64- and 32-bit versions of Windows 10 Professional, 8.1 Professional, and 7 Professional. Supports C# and VB.NET	
PCI compliance	PCI Specification Revision 2.2	PCI Express Specification Revision 1.1*
Opto 22 brain compatibility	B4, B5, B6, SNAP-B4, SNAP-B6	B4, B5, B6, SNAP-B4, SNAP-B6
LEDs	Four, indicating board access, reset level, read, and write	Four, indicating board access, reset level, read, and write
Operating temperature	0 to 70 °C	0 to 60 °C
Storage temperature	-30 to 85 °C	-30 to 85 °C

* Because PCI Express is backward compatible, the card is also compatible with version 2 and 3 PCIe slots.

LEDs

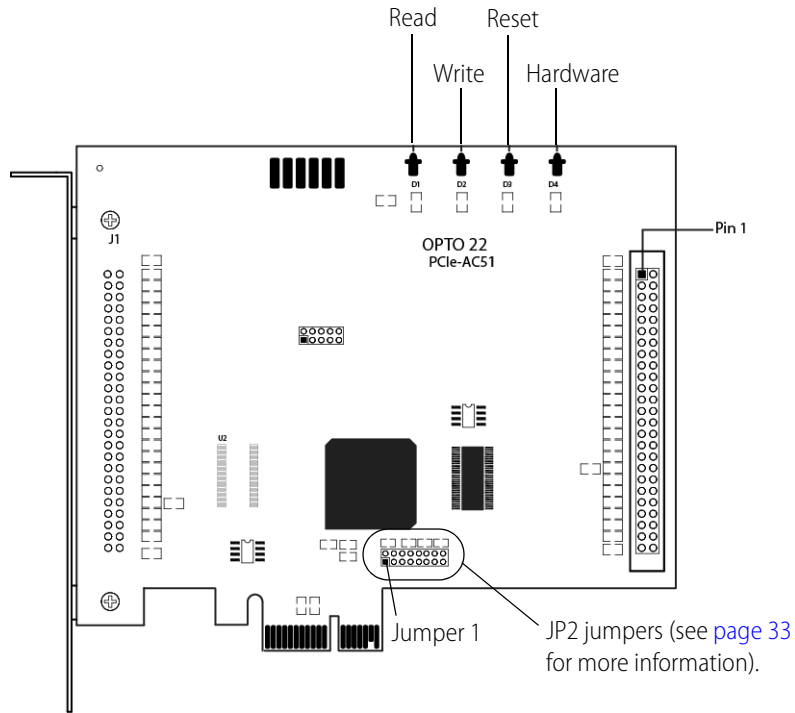
PCI-AC51



- **Reset**—Indicates the state of the reset line, which can be either High or Low (On = High).
- **Hardware**—Blinks when the PCI-AC51 is accessed.
- **Write**—Blinks when a Pamux write is executed.
- **Read**—Blinks when a Pamux read is executed.

NOTE: If the PCI-AC51 is rapidly or continually accessed, the blinking indicators may appear to stick on (will not blink). This is normal.

PCIe-AC51



- **Read**—Blinks when a Pamux read is executed.
- **Write**—Blinks when a Pamux write is executed.
- **Reset**—Indicates the state of the reset line, which can be either High or Low (On = High).
- **Hardware**—Blinks when the PCIe-AC51 is accessed.

NOTE: If the PCIe-AC51 is rapidly or continually accessed, the blinking indicators may appear to stick on (will not blink). This is normal.

B: Converting Applications to a Newer Card

This section describes how to convert legacy applications that used the AC28 and PCI-AC28 adapter cards.

Applications that Used the OptoPMUX.dll for the AC5

This version of the library does not support PCI-AC5 or PCIe-AC5 adapter cards. Special historic versions of the OptoPMux.dll supported the AC5. This support is not possible due to independent hardware identification issues and the PCI bus. For details, see form 1211, the [PCI-AC5, PCIeAC5, AC5, and G4AC5 User's Guide](#).

Converting Applications that Use Inp() and Outp()

This PCI SDK library provides a consistent application code model for Windows 10 Professional, Windows 8.1 Professional, and Windows 7 Professional operating systems.

The primary advantage of converting your legacy application to this library is the encapsulation of the Pamux functions. This library provides high-level functionality, as opposed to setting and clearing bits.

Inp() and Outp() function calls at the user level are unsupported because of the Windows hardware abstraction layer.

To integrate with the existing Pamux interface, the PCI-AC51 and PCIe-AC51 mimic the 50-wire IDC connector and the Pamux timing interface. The hardware model to the PC is radically different compared to the AC28. The AC28 relied on jumper settings for configuration; these jumpers are not included on the PCI-AC51 or the PCIe-AC51.

C: PCI-AC51 and PCIe-AC51 Technical Reference

This technical reference is provided for those who are either technically curious or are interested in authoring a device driver for an unsupported operating system or a Windows kernel-mode driver.

The information provided is *as is*; you use this information at your own risk. Opto 22 does not support any kernel layer or drivers other than those we provide.

In this appendix:

Architecture.....	page 30
Adapter Card Identification	page 30
Adapter Card Base Address Configuration	page 30
Register Description	page 31
Powerup Conditions.....	page 32
Controlling the Reset Line	page 32
DIP Switches (PCI-AC51) and Jumpers (PCIe-AC51).....	page 33
High and Low Latency Modes	page 34
Performing Pamux Operations.....	page 34
Notes for the SNAP-B6 (Digital Aspect)	page 36

Architecture

The PCI-AC51 and PCIe-AC51 cards contain an Altera FPGA (U8 on the PCI-AC51 and U1 on the PCIe-AC51). The Altera FPGA implements the PCI or PCI Express interface to the host computer, regulating data transactions between the host computer and the adapter card while also providing adapter card configuration information to the host computer's BIOS. The FPGA also implements Opto 22's Pamux state machine, which drives the Pamux bus cycle and provides the register architecture visible to the software application.

Adapter Card Identification

PCI Vendor ID, 0x148a

PCI Device ID, 0xac51

Type "PCI Simple Communications Controller"

Adapter Card Base Address Configuration

Base Address Register 1 (BAR1) is memory mapped. It is configured as a 16 consecutive byte-only array. Only the first 7 bytes are used. Accessing unmapped registers may cause the computer to bus lock.

BAR1 Register Table

An X indicates that data is ignored.

Offset	Direction	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	R	Reset Status Register	Reset Busy	X	X	X	X	X	X	Reset Level
0x00	W	Reset Control Register	X	X	X	X	X	X	X	Reset Level
0x01	R/W	Bank Address Register	X	X	Pamux A5	Pamux A4	Pamux A3	Pamux A2	Pamux A1	Pamux A0
0x02	R/W	Data Register	Pamux D7	Pamux D6	Pamux D5	Pamux D4	Pamux D3	Pamux D2	Pamux D1	Pamux D0
0x03	R	Start Read Register	1	0	1	0	1	0	1	1
0x03	W	Start Write Register	X	X	X	X	X	X	X	X
0x04	R	Status Register	X	Read Status	Write Status	0	0	Pamux ID1	Pamux ID0	Pamux Busy
0x05	R	Switch/Jumper Register*	0	0	S6/J6	S5/J5	S4/J4	S3/J3	S2/J2	S1/J1
0x06	R	Version Register	D7	D6	D5	D4	D3	D2	D1	D0

* PCI-AC51 has switches. PCIe-AC51 has jumpers. See [page 25](#) for jumper location.

Register Description

Reset Status Register (0x00) The Reset Status Register reports the reset state (Reset Level, bit 0) and the status of a 150 ms timer (Reset Busy, bit 7). The timer provides the application a way to ensure a minimum reset duration for all Pamux brains. The timer is restarted (and the Reset Busy Bit set to one) when register 0x00 (Reset Control Register) is written. When the duration expires, the Reset Busy Bit is set to zero.

Reset Control Register (0x00) Bit 0 drives the reset signal on the Pamux bus. Writing a one asserts the reset signal to the “low” level. Writing a zero asserts the reset signal to the “high” level.

Bank Address Register (0x01) The Pamux address for the next Pamux read/write cycle is stored in this register. This register must not be modified while the Busy Bit of the Status Register is set to one.

Data Register (0x02) Do not access this register while a Pamux operation is in progress.

Writing: The data to be written in the next Pamux write cycle should be stored in this register before the Pamux write cycle is started.

Reading: After the Pamux read cycle is complete, this register holds the data that was read from the Pamux brain.

Start Read Register (0x03) A read initiates a Pamux read bus cycle. Reading from this register initiates a Pamux read. The bank Address Register must be configured before reading from this register. A value of 0xAB is always returned in this register and can be ignored. The data read from the Pamux brain is returned in the Data Register. Do not access this register while a Pamux operation is in progress.

Start Write Register (0x03) Writing to this register initiates a Pamux write cycle. The Bank Address Register and the Data Register must be configured before writing to this register. Do not access this register while a Pamux operation is in progress.

Status Register (0x04) This register contains the Busy Bit (busy = a value of one), to indicate if a Pamux bus cycle is in progress. The ID bits hold the value of the last brain communicated with. These bits are valid only after the Busy Bit has returned to a value of zero. The Write Status bit is one if the current or last Pamux bus cycle was a Pamux write cycle. The Read Status bit is one if the current or last Pamux bus cycle was a Pamux read.

Switch/Jumper Register (0x05) Provides state of the switches on the PCI-AC51 or state of the jumpers on the PCIe-AC51. Custom driver applications may use S1 through S4 (or J1 through J4) in any desired fashion. S5/J5 and S6/J6 may be verified for proper hardware settings.

Version Register (0x06) Returns the version of the state machine architecture.

Powerup Conditions

An X indicates that data is ignored.

Offset	Direction	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	R/W	Reset Control Register	X	X	X	X	X	X	X	0
0x01	R/W	Bank Address Register	X	X	X	X	X	X	X	X
0x02	R/W	Data Register	X	X	X	X	X	X	X	X
0x04	R	Status Register	X	0	0	0	0	X	X	0
0x05	R	Switch/Jumper Register*	X	X	S6/J6	S5/J5	S4/J4	S3/J3	S2/J2	S1/J1

* PCI-AC51 has switches. PCIe-AC51 has jumpers. See [page 25](#) for jumper location.

Controlling the Reset Line

Controlling the Reset Line Using the Pamux Driver

When using the Pamux driver, it is necessary to tell the driver how the reset level is configured on the brains. This is done as part of the GetCard() function. Use one of the following values:

- eResetLevel.BrainsSetForResetActiveLow
- eResetLevel.BrainsSetForResetActiveHigh

You will then be able to reset the brains in the system or take the brains out of reset by using the following functions:

- ResetBrains()
Be sure to keep the brains in reset for at least 150 milliseconds to ensure all brains properly reset.
- ReturnToNormalAfterReset()

You can also reset the brains in the system and take the brains back out of reset using a single function, which does a 150 milliseconds rest:

- ResetBrainsAndReturnToNormal()

Controlling the Reset Line When Not Using the Pamux Driver

Bit 0 of the Reset Control Register on the PCI-AC51 and the PCIe-AC51 is used to control the reset line.

- When bit 0 is set to zero, the reset line will be HIGH and LED 1 will be ON.
- When bit 0 is set to one, the reset line will be LOW and LED 1 will be OFF.

The reset line must be in the appropriate state (depending on the reset configuration on the brains) for a minimum of 150 milliseconds to ensure all brains reset.

Default Power-Up State of Reset Line

When a computer with a PCI-AC51 or PCIe-AC51 card installed has just been powered up, bit 0 of the reset control register will be in its default power-up condition, which is a bit value of zero. This drives the reset line HIGH and turns LED 1 (reset line) ON.

Reset Configuration of Brains When Using PamScan

One of the parameters you must choose when using the PamScan utility is the reset level of the brains in your system. Set the option in PamScan to match how your Pamux brains are configured. PamScan needs to know the brain configuration so that it can determine whether to set the reset line HIGH or LOW in order to take the system out of reset. This allows PamScan to configure and control the brains and output modules.

Testing

The following table shows the results of using each of the possible reset settings on the brains and in the PamScan utility. After choosing the reset setting in PamScan, click Connect to establish communication with the PCI-AC51 or PCIe-AC51 card and brains.

Jumper 7 Setting on Brains	Pam Scan Configuration	Reset Line Voltage	State of Reset Line	Status
Set "Active Low"	Choose "Active Low"	3.48 VDC	HIGH	Brains allow control of outputs
Set "Active Low"	Choose "Active High"	0 VDC	LOW	Brains are reset
Set "Active High"	Choose "Active Low"	3.48 VDC	HIGH	Brains are reset
Set "Active High"	Choose "Active High"	0 VDC	LOW	Brains allow control of outputs

DIP Switches (PCI-AC51) and Jumpers (PCIe-AC51)

DIP switches or jumpers are provided on the adapter cards for custom implementation and some Pamux settings.

- The **PCI-AC51** has six DIP switches (the DS1 bank). A switch in the ON position reads as a zero and OFF reads as a one. The card is shipped with all switches in the ON position.
- The **PCIe-AC51** has eight jumpers (labeled JP2), of which only the first six are used. A jumper removed reads as a zero, and a jumper in place reads as a one. The card is shipped with jumpers removed. (Do not use the jumper group labeled J3. See the diagram on [page 25](#) for jumper group location.)

DIP switches and jumpers may be modified on a system that is powered; however, the applications that use the PCI-AC51 or PCIe-AC51 must be stopped.

Switches or jumpers 1 through 4 are provided for custom implementation. One common use will be for identifying certain cards in a multiple card system. The settings of these switches or jumpers do not affect the Pamux operation.

Switches or jumpers 5 and 6 are used to set the duration of Pamux read and write strobes.¹ Under most conditions, these do not have to be changed.

S5 (Switch)	Jumper 5	Strobe Duration
On	Out	2.7 μ S*
Off	In	3.3 μ S

*Factory default. Also the legacy AC28 (K Revision) Pamux bus timing.

S6 (Switch)	Jumper 6	Strobe Duration
On	Out	High Latency Mode**
Off	In	Low Latency Mode

** Factory Default

High and Low Latency Modes

The amount of idle time between Pamux bus cycles has never been specified. Historic Pamux hardware assumed that there was adequate software delay time between each bank access. Recent PCs are capable of performing back-to-back bus cycles with practically no delay—as little as 60 nanoseconds (ns).

The PCI-AC51 (state machine version 2) and PCIe-AC51 enforce idle time between bus cycles. In high latency mode (S6 on, or jumper 6 not installed), 3,600 ns of idle time is added between each bus cycle. This means the Busy Bit may be busy for as much as 7,700 ns. The additional delay time is required for many B6 analog brains. These brains are software driven and need the additional time to properly operate.

In some cases of digital only systems (systems that contain only B4, SNAP-B4, or B5 brains), S6 may be turned off (low latency mode) if the additional latency is undesirable.

Performing Pamux Operations

NOTE: See “BAR1 Register Table” on page 30 for the register names, bit names, and bit positions referred to in this section.

Issuing Pamux Reset

The PCI-AC51 and PCIe-AC51 provide a level-oriented reset control feature.

1. To assert the Pamux Reset, write a value (either zero or one) to the Reset Control Register (BAR1, offset 0x00), bit 0. A zero sets the Pamux reset level to high. A one sets the Pamux reset level to low.

¹ Legacy Pamux documentation indicates that the on-time specification of the read and write strobes were 2,000 nanoseconds (ns). Through the lifecycle of Pamux hardware, this time was increased to 2,700 ns (brain and AC28s with a Lattice chip). When S5 is on (PCI-AC51) or jumper 5 is not installed (PCIe-AC51), the card maintains the last AC28 (Rev K) timing specification, 2,700 ns. Switching to the 3,300 ns setting may improve long cable issues.

2. Poll the status register and inspect the state of Reset Busy (bit 7). The Reset Busy enforces a 150 ms delay interval, the recommended reset duration for all Pamux hardware. Digital-only brains may reset with a much shorter interval.
3. Wait for the Busy Bit to return to zero. Then, write a complementary value to bit 0 of the Reset Control Register. This action will negate the Pamux Reset signal.

NOTE: Reset is asynchronous to all Pamux operations. Reset may be asserted while the Pamux bus cycle is in operation (The Busy Bit in the Status Register is asserted). If a Pamux read was in progress, the result is uncertain. If a write is in progress, the reset must be asserted well after the Pamux bus cycle is completed. If the reset is negated before the end of the current Pamux write, the written result is uncertain.

NOTE: A substantial reset period is required for analog or mixed brains. Typically, a Pamux analog brain will reset if the Reset is asserted for 150 ms. The Busy bit in the Reset Status Register provides a 150 ms busy indication. Note this is simply an indicator. Additional changes to the reset state will immediately appear on the Pamux Reset signal and the 150 ms timer is reset upon each write to the Reset Control Register.

Reading the State Machine Version

To get the state machine version, read the Version Register. All 8 bits of the register are significant to the state machine's implementation revision.

This register is independent of all Pamux operations. It may be read without impacting any current operations the state machine is performing.

Reading the Switch/Jumper Values

The bit positions in the Switch/Jumper Register² correspond to the switches or jumpers on the board:

- PCI-AC51—A switch in the on position reads as a zero.
- PCIe-AC51—An uninstalled jumper reads as a zero.

Polled PCI-AC51 and PCIe-AC51 Operation

NOTE: All of the directions assume a byte-wide access to the card.

Performing a Pamux Bank Write

1. Wait until the Busy Bit (bit 0) in the Status Register is zero.
2. Write the Pamux bank address to BAR1 offset 1.
3. Write the Pamux data to BAR1 offset 2.
4. A write to the Start Cycle Register (BAR1, offset 3) begins a Pamux bus cycle. The data written to the register is ignored.
5. Read the Status Register (BAR1, offset 4) and inspect the Busy Bit. If the bit has a value of one (asserted), a Pamux cycle is in progress. Do not modify the Bank Address Register, Data Register, or the Start Cycle Register while the busy status is asserted.

²PCI-AC51 has switches. PCIe-AC51 has jumpers.

6. (Optional—Not all Pamux brain versions support the brain identification function.) To get the Pamux brain identification (ID), wait until the Busy Bit has a value of zero (negated). Then, read bits 1 and 2 of the Status Register (BAR1 offset 4).

Performing a Pamux Bank Read

1. Wait until the Busy Bit (bit 0) in the Status Register is zero.
2. Write the Pamux bank address to the Bank Address Register (BAR1, offset 1).
3. Read the Start Read Register (BAR1, offset 3) to start a Pamux read bus cycle. Ignore the data read from this register. (The returned data is hard coded as 0xab.)
4. Read the Status Register (BAR1, offset 4) and inspect the Busy Bit. If the bit is one (asserted), a Pamux cycle is in progress.
5. Wait until the Busy Bit is zero (negated). Then, read the Pamux data from the Data Register (BAR1, offset 2).

Notes for the SNAP-B6 (Digital Aspect)

The SNAP-B6's digital I/O is software driven. This is currently the only brain that uses software to emulate a hardware-only digital brain.

If operations between digital accesses are too fast, intermittent and unpredictable behavior may occur. If you should see this, here are workarounds and recommendations:

1. Read and write banks to the SNAP-B6, only one bank at a time.
2. Allow a 2 millisecond delay before the next register operation occurs.
3. For PCIAC51.dll, use Set Slow Mode Delay() and Enable Slow Mode() to control the timing of operations. If you are using the OptoPM32.dll for the PCI-AC51 or PCIe-AC51, use the Pamux_Set_Slow_Bank_Range function.
4. If high-speed digital is required, separate the digital and analog I/O by using a SNAP-B4 and for the digital I/O.

Alternate Method

(OptoPM32.dll only) Create a text file named "o22_pamux_slow_bank.txt" in the folder where the application starts. There are two numbers in this file, as follows:

(starting bank number) (ending bank number)

So, if you have SNAP-B6s located, for example, at address 32 and ending at bank 40 (decimal addresses), you would want to set the numbers to be:

32 40

If you have multiple adapters, this delay applies to all PCI-AC51 or PCIe-AC51 adapters.