

GX7404

Power Interface Prototyping

PXI Board

GXPXI Software

User's Guide

Last Updated: October 12, 2012

GEOTEST

MARVIN TEST SYSTEMS, INC.

Safety and Handling

Each product shipped by Geotest is carefully inspected and tested prior to shipping. The shipping box provides protection during shipment, and can be used for storage of both the hardware and the software when they are not in use.

The circuit boards are extremely delicate and require care in handling and installation. Do not remove the boards from their protective plastic coverings or from the shipping box until you are ready to install the boards into your computer.

If a board is removed from the computer for any reason, be sure to store it again in its original shipping box. Do not store boards on top of workbenches or other areas where they might be dropped or be exposed to strong electromagnetic or electrostatic fields. Boards should be stored in protective anti-electrostatic wrapping and away from electromagnetic fields.

Be sure to make a single copy of the software diskette and store the original in a safe place away from electromagnetic or electrostatic fields.

Warranty

Geotest products are warranted against defects in materials and workmanship for a period of 12 months (3 months for software products). Geotest shall repair or replace (at its discretion) any defective product during the stated warranty period. The software warranty includes any revisions or new versions released during the warranty period. If you need to return a board, please call 949-263-2222 for an RMA number.

If You Need Help

Visit our web site at <http://www.geotestinc.com> for more information about Geotest products, services and support options. Our web site contains sections describing: Support options, Application notes, Download area for downloading patches, example, patches and new or revised instrument drivers. To submit a support issue including suggestion, bug report or question please use the following link:

<http://www.geotestinc.com/magic>

You can also use Geotest technical support phone line (949) 263-2222. This service is available between 9:00 AM and 5:00 PM Pacific Standard Time.

Disclaimer

In no event shall Geotest or any of its representatives be liable for any consequential damages whatsoever (including unlimited damages for loss of business profits, business interruption, loss of business information, or any other losses) arising out of the use of or inability to use this product, even if Geotest has been advised of the possibility for such damages.

Copyright

Copyright © 2004-2012 by Geotest, Marvin Test Systems, Inc. All rights reserved. No part of this document can be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Geotest.

Trademarks

ATEasy®, CalEasy, DIOEasy®, DtifEasy, WaveEasy	Geotest – Marvin Test Systems Inc.
C++ Builder, Delphi	Embarcadero Technologies Inc.
LabView, LabWindowstm/CVI	National Instruments
Microsoft Developer Studio, Microsoft Visual C++, Microsoft Visual Basic,, .NET, Windows 95, 98, NT, ME, 2000, XP, VISTA and Windows 7	Microsoft Corporation

All other trademarks are the property of their respective owners.

Table of Contents

Safety and Handling.....	i
Warranty	i
If You Need Help.....	i
Disclaimer	i
Copyright	i
Trademarks	ii
Chapter 1 - Introduction	1
Manual Scope and Organization.....	1
Conventions Used in this Manual	1
Chapter 2 - Overview	3
Introduction.....	3
Features.....	3
Applications.....	3
Board Description	4
Output Voltages	4
Architecture	5
Block Diagram	5
Specifications.....	6
Virtual Panel Description.....	7
Virtual Panel Setup Page	8
Virtual Panel About Page.....	9
Chapter 3 - Installation and Connections	11
Getting Started	11
Installation of the GXPXI Software.....	12
Overview of the GXPXI Software	12
Configuring Your PXI System using the PXI/PCI Explorer.....	13
Board Installation.....	14
Before you Begin	14
Electric Static Discharge (ESD) Precautions	14
Installing a Board	14
Plug & Play Driver Installation.....	15
Removing a Board	15
Connectors	16
Output Channels Connector (J3).....	16
Connectors and Accessories	17

Installation Folders	17
Driver Files Description.....	18
Example Programs	18
Setup Maintenance Program	20
Chapter 4 - Programming the Board	21
The GXPXI Driver.....	21
Programming Using C/C++ Tools	21
Programming Using Visual Basic.....	21
Programming Using Pascal/Delphi	22
Programming Using ATEasy®	22
Programming Using LabView and LabView/Real Time	22
Using and Programming under Linux.....	22
Using the GXPXI driver functions.....	23
Initialization, HW Slot Numbers and VISA Resource	23
Board Handle	24
Reset.....	24
Error Handling	24
Driver Version.....	24
Panel.....	24
Distributing the Driver	25
Sample Programs	25
Sample Program Listing	26
Chapter 5 - Functions Reference	29
Introduction.....	29
GX7404 Functions	30
Gx7404ConnectChannel	31
Gx7404ConnectChannels	32
Gx7404GetBoardSummary.....	33
Gx7404GetChannel	34
Gx7404GetChannelCurrent	35
Gx7404GetChannelVoltage.....	36
Gx7404GetChannels	37
Gx7404GetExternalDisableState	38
Gx7404GetOnBoardChannelType.....	39
Gx7404Initialize	40
Gx7404InitializeVisa	41
Gx7404Panel.....	42

Gx7404Reset.....	43
GxPxiGetDriverSummary.....	44
GxPxiGetErrorString	45
Index	47

Chapter 1 - Introduction

Manual Scope and Organization

Manual Scope

The purpose of this manual is to provide all the necessary information to install, use, and maintain the GX7404 instrument. This manual assumes the reader has a general knowledge of PC based computers, Windows operating systems, and some understanding of digital I/O.





This manual also provides programming information using the GX7404 driver. Therefore good understanding of programming development tools and languages may be necessary.

Manual Organization

The GX7404 manual is organized in the following manner:

Chapter	Content
Chapter 1 – Introduction	Introduces the GX7404 manual. Lists all the supported board and shows warning conventions used in the manual.
Chapter 2 – Overview	Describes the GX7404 features, board description, its architecture, specifications and the panel description and operation.
Chapter 3 – Installation and Connections	Provides instructions on how to install the GX7404 Board and its accompanying GXPXI software.
Chapter 4 – Programming the Board	Provides a listing of GX7404 driver files, general purpose/generic driver functions, and programming methods. Discusses various supported operating systems and development tools.
Chapter 5 – Functions Reference	Contains a listing of the general GX7404 functions. Each function is described along with its syntax, parameters, and special programming comments. Samples are given for each function.

Conventions Used in this Manual

Symbol Convention	Meaning
	Static Sensitive Electronic Devices. Handle Carefully.
	Warnings that may pose a personal danger to your health. For example, shock hazard.
	Cautions where computer components may be damaged if not handled carefully.
	Tips that aid you in your work.

Formatting Convention	Meaning
Monospaced Text	Examples of field syntax and programming samples.
Bold type	Words or characters you type as the manual instructs. For example: function or panel names.
<i>Italic type</i>	Specialized terms. Titles of other reference books. Placeholders for items you must supply, such as function parameters

Chapter 2 - Overview

Introduction

The GX7404 is a 3U Power Interface and Prototype PXI card. The board provides direct controlled interface to the PXI +3.3VDC, +5VDC, +12VDC, and -12VDC backplane voltages and monitor each voltage level and current. The GX7404 has an additional high current voltage source that is installed as specified by the customer (default +5V).

The GX7404 Power Interface was designed for ATE, data acquisition, or process control systems.

Features

Principal capabilities of the GX7404 board are summarized in the following list of features.

- +3.3V, up to 2A.
- +5V, up to 2A.
- +12V, up to 0.5A.
- -12V, up to 0.5A.
- Customer defined high-current voltage source (default +5V).
- Each of the five voltage sources can be independently connected/disconnected to the output connector.
- Monitor each of the voltage sources voltage level and current.
- External Disable line enabling remote control of the four PXI voltages outputs.
- GXPXI software includes libraries to support controlling and viewing the board settings by 32/64 bit applications for various programming environments, or using the provided front panel application .

Applications

- Simulation of discrete levels, common in avionics testing.
- Limited use for UUT power.
- Requirements for multiple programmable, precision DC sources.

Architecture

Block Diagram

Figure 2-2:GX7404 Block Diagram illustrates the GX7404 architecture. The board is programmed by software using the PXI interface to connect the voltage sources. The board connector on the right (J3) has pins that output the channels output signals and input signal for the external input signals.

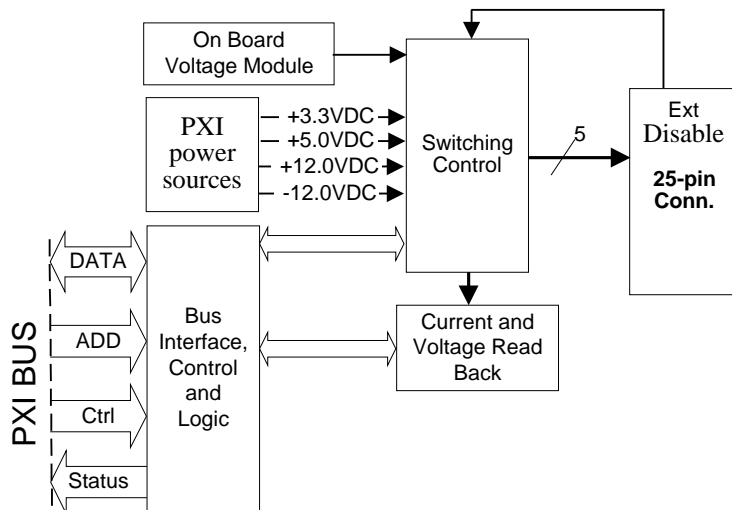


Figure 2-2:GX7404 Block Diagram

Specifications

Power Input:	
Per PXI System Alliance Power Supply Specs. <i>PXI Hardware Specifications Rev. 2.1 02/04/03, pp. 36-37.</i>	
*Power Output:	
+5V Channel	2A(max.)
+3.3V Channel	2A(max.)
+12V Channel	0.5A(max.)
-12V Channel	0.5A(max.)
**DCDC Channel (5V)	0.5A(max.)
Current Read back:	
Resolution	16 bits.
Accuracy	+/-1mA.
Voltage Read back:	
Resolution	12 bits.
Accuracy	+/-10mV.
Protection Circuitry:	
Isolation Relays	All output channels.
Remote Inhibit	Opens all output channel relays.
Fuse (+/- 12V)	0.5A, 2AG Fast Acting.
Fuse (3.3V, 5V)	2A, 2AG Fast Acting.
Fuse (DCDC)	1.5A, 2AG Slow Blow.
UUT Discharge	Discharge resistors footprint available on GX7404. Not install by default. User specified.
Operating Temperature	0 to+55°C
Storage Temperature	-20 to+70°C
Size	3U PXI
Weight	12 oz.

* No line sensing available.

** This channel output not referenced to ground. Factory default output voltage is 5V. User may specify other outputs i.e. 12V, 15.

Virtual Panel Description

The GX7404 includes a virtual panel program, which enables full utilization of the various configurations and controlling modes. To fully understand the front panel operation, it is best to become familiar with the functionality of the board.

To open the virtual panel application, select **GX7404 Panel** from the **Geotest**, **GXPXI** menu under the **Start** menu. The GX7404 virtual panel opens as shown here:

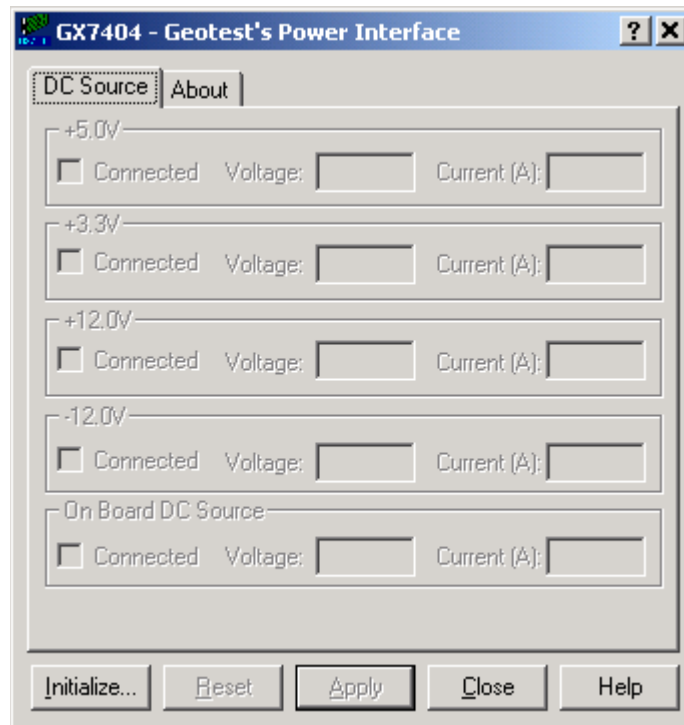


Figure 2-3: GX7404 Virtual Panel

The following controls are shown:

Initialize: The user is prompted to select the slot where the board is installed in the Initialize dialog box as shown in Figure 2-4 :



Figure 2-4: Initialize Dialog Box

The slot number in the Initialize dialog box refers to the PXI slot in which the GX7404 board is installed. Select the slot from the drop down list. The list displays the slots where GX7404 boards are installed. The specified slot number can also be reviewed or set by using the **PXI/PCI Explorer** applet that can be opened from the Windows Control Panel. Select the board slot number and click **OK** to initialize the driver for the specified board.

Reset: Resets all digital I/O ports to input direction and all counters to their default state.

Close: Closes (exits) the GX7404 panel.

Help: Opens the GX7404 on-line help window.

Virtual Panel Setup Page

After the board is initialized the panel is enabled and will display the current setting of the board. The following picture shows the **Setup page** settings:

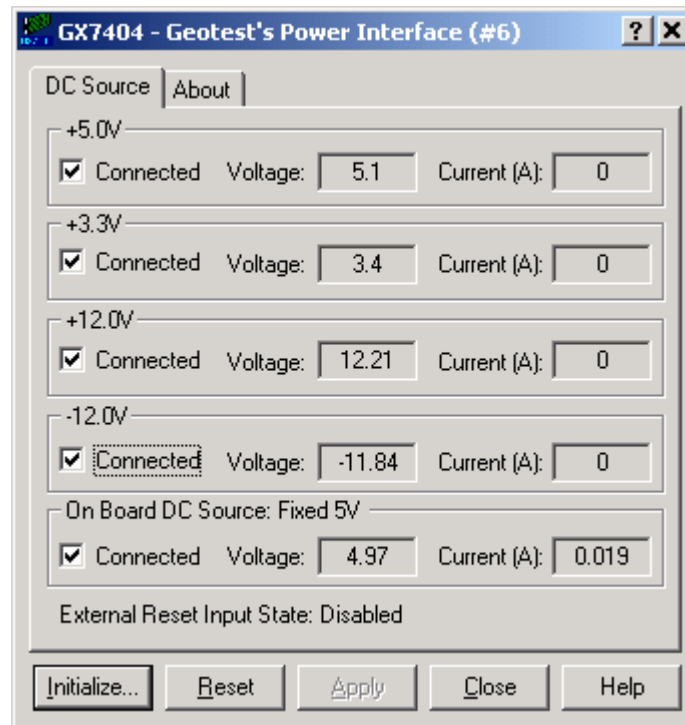


Figure 2-5: GX7404 Virtual Panel (Initialized)

The following controls are shown in the Setup page:

+5.0V, +3.3V, +12V, -12V and On Board DC Source Group Box:

Connected (Check Box): Sets/display the correspondent channel connection.

Voltage / Current (Text box): When channel is connected (checked) then the Voltage and Current of that channel is displayed.

External Reset Input state (Text): Display the External Reset Input state.

Virtual Panel About Page

Clicking on the **About** tab will show the **About page** as shown in Figure 2-6:

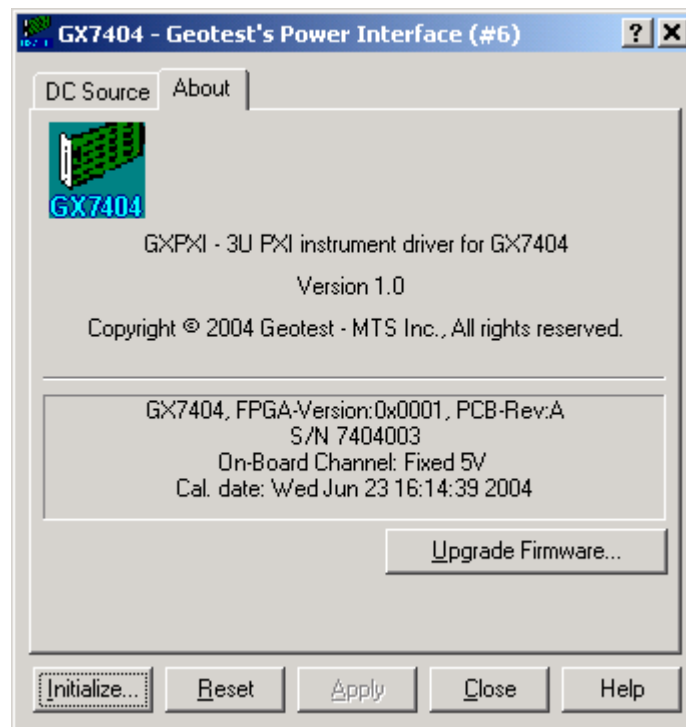


Figure 2-6:GX7404 Virtual Panel – About Page

The top part of the **About** page displays version and copyright of the GX7404 driver. The bottom part displays the board summary, including the EEPROM version, the board Revision, the FPGA version, the board serial number and the calibration time. The **About** page also contains a button **Upgrade Firmware...** used to upgrade the board FPGA. This button maybe used only when the board requires upgrade as directed by Geotest support. The upgrade requires a firmware file (.jam) that is written to the board FPGA. After the upgrade is complete you must shut down the computer to recycle power to the board.

Chapter 3 - Installation and Connections

Getting Started

This section includes general hardware installation procedures for the GX7404 board and installation instructions for the GX7404 (GXPXI) software. Before proceeding, please refer to the appropriate chapter to become familiar with the board being installed.

To Find Information on...	Refer to...
Hardware Installation	This Chapter
GX7404 Driver Installation	This Chapter
Programming	Chapter 4
GX7404 Function Reference	Chapter 5

Packing List

All GX7404 boards have the same basic packing list, which includes:

1. GX7404 board
2. Disk with GXPXI software

Unpacking and Inspection

After removing the board from the shipping carton:



Caution - Static sensitive devices are present. Ground yourself to discharge static.

1. Remove the board from the static bag by handling only the metal portions.
2. Be sure to check the contents of the shipping carton to verify that all of the items found in it match the packing list.
3. Inspect the board for possible damage. If there is any sign of damage, return the board immediately. Please refer to the warranty information at the beginning of the manual.

System Requirements

All GX7404 instrument boards are designed for use with a 3U or 6U cPCI or PXI compatible chassis and the software is compatible with any computer system running Windows 9x, Windows Me, Windows NT or Windows 2000 and above operating systems. In addition, Microsoft Windows Explorer version 4.0 or above is required to view the online help.

Each board requires one unoccupied 3U PXI bus slot.

Installation of the GXPXI Software

Before installing the board, it is recommended to install the software as described in this section:

1. Insert the Geotest CD-ROM and locate the **GXPXI.EXE** setup program. If your computer's Auto Run is configured, when inserting the CD a browser will show several options, select the Geotest Files option, then locate the setup file. If Auto Run is not configured you can open the Windows explorer and locate the setup files (usually located under \Files\Setup folder). You can also download the file from Geotest web site (www.geotestinc.com).
2. Run the setup and follow the instruction on the Setup screen to install the software.

Note: You may be required to restart the setup after logging-in as a user with Administrator privileges. This is required in-order to upgrade your system with newer Windows components and to install the HW kernel-mode device drivers that are required by the GXPXI driver to access resources on your board.

3. The first setup screen to appear is the Welcome screen. Click **Next** to continue.
4. Enter the folder where the software is to be installed. Either click **Browse** to set up a new folder, or click **Next** to accept the default entry of C:\Program Files\Geotest\GXPXI.
5. Select the type of Setup you wish and click **Next**. You can choose between **Typical**, **Run-Time** and **Custom** setups. **Typical** setup type installs all files. **Run-Time** setup type will install only the files required for controlling the board either from its driver or from its virtual panel. **Custom** setup type lets you select from the available components.

The program will now start its installation. During the installation, Setup may upgrade some of the Windows shared components and files. The Setup may ask you to reboot after it complete if some of the components it replaced were used by another application during the installation – do so before attempting to use the software.

You can now continue with the installation to install the board. After the board installation is complete you can test your installation by starting a panel program that let you control the board interactively. The panel program can be started by selecting it from the Start, Programs, GXPXI menu located in the Windows Taskbar.

Overview of the GXPXI Software

Once the software installed, the following tools and software components are available:

- **PXI/PCI Explorer applet** – use to configure the PXI chassis, controllers and devices. This is required for accurate identification of your PXI instruments later on when installed in your system. The applet configuration is saved to PXISYS.ini and PXIeSYS.ini that are used by Geotest instruments, the VISA provider and VISA based instruments drivers. In addition, the applet can be used to assign chassis numbers, Legacy Slot numbers and instruments alias names.

VISA is a standard maintained by the VXI Plug & Play System Alliance and the PXI Systems Alliance organizations (<http://www.vxipnp.org/>, <http://www.pxisa.org/>). VISA provides a standard way for instrument manufacturers and users to write and use instruments drivers. The VISA resource managers such as National Instruments **Measurement & Automation** (NI-MAX) can display and configure instruments and their address (similar to Geotest's PXI/PCI Explorer).

- **GXPXI Panel** – use to configure, control and display the board settings and counter reading.
- **GXPXI driver** - A DLL based function library (GXPXI.DLL or GXPXI64.DLL, located in the Windows System folder) used to program and control the board. The driver uses Geotest's HW driver or VISA supplied by third party vendor to access and control the GXPXI board.
- **Programming files and examples** – interface files and libraries for various programming tools, see later in this chapter for a complete list of files, programming languages and development tools supported by the driver.
- **Documentation** – On-Line help and User's Guide.

Configuring Your PXI System using the PXI/PCI Explorer

To configure your PXI/PCI system using the **PXI/PCI Explorer** applet follow these steps:

1. **Start the PXI/PCI Explorer applet.** The applet can be start from the Windows Control Panel or from the Windows Start Menu, **Geotest, HW, PXI/PCI Explorer**.
2. **Identify Chassis and Controllers.** After the PXI/PCI Explorer started it will scan your system for changes and will display the current configuration. The PXI/PCI Explorer automatically detects systems that have Geotest controllers and chassis. In addition, the applet detects PXI-MXI-3/4 extenders in your system (manufactured by National Instruments). If your chassis is not shown in the explorer main window, use the Identify Chassis/Controller commands to identify your system. Chassis and Controller manufacturers should provide INI and driver files for their chassis and controllers to be used by these commands.
3. **Change chassis numbers, PXI devices Legacy Slot numbering and PXI devices Alias names.** These are optional steps to be performed if you would like your chassis to have different numbers. Legacy slots numbers are used by older Geotest or VISA drivers. Alias names can provide a way to address a PXI device using your logical name (e.g. "DMM1"). For more information regarding these numbers see the **GXPXIInitialize** and **GXPXIInitializeVisa** functions.
4. **Save you work.** PXI Explorer saves the configuration to the following files located in the Windows folder: PXISYS.ini, PXIeSYS.ini and GxPxiSys.ini. Click on the **Save** button to save you changes. The PXI/Explorer prompt you to save the changes if changes were made or detected (an asterisk sign ' * ' in the caption indicated changes).

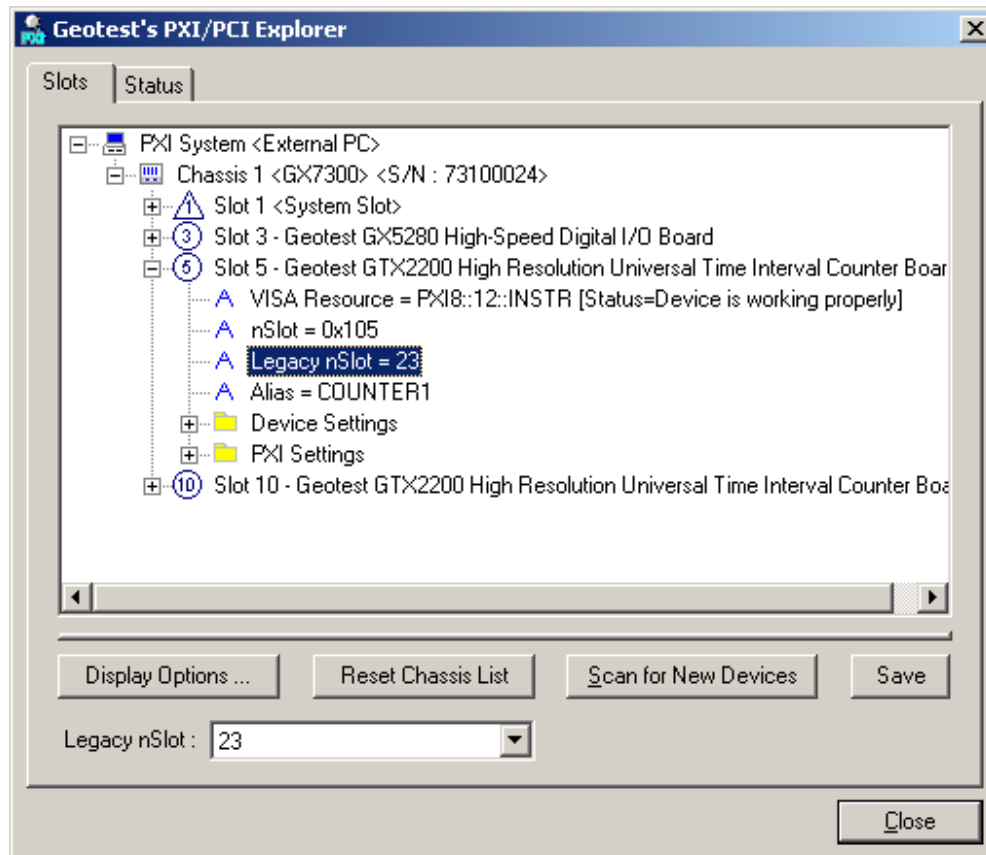


Figure 3-1: PXI/PCI Explorer

Board Installation

Before you Begin

- Install the software driver as described in the prior section.
- Configure your PXI/PC system using **PXI/PCI Explorer** as described in the prior section.
- Verify that all the components listed in the packing list (see previous paragraph) are present.

Electric Static Discharge (ESD) Precautions

To reduce the risk of damage to the board, the following precautions should be observed:

- Leave the board in the anti-static bags until installation requires removal. The anti-static bag protects the board from harmful static electricity.
- Save the anti-static bag in case the board is removed from the computer in the future.
- Carefully unpack and install the board. Do not drop or handle the board roughly.
- Handle the board by the edges. Avoid contact with any components on the circuit board.



Caution - Do not insert or remove any board while the computer is on. Turn off the power from the PXI chassis before installation.

Installing a Board

Install the board as follows:

1. Turn off the PXI chassis and unplug the power cord.
2. Locate a PXI empty slot on the PXI chassis.
3. Place the module edges into the PXI chassis rails (top and bottom).
4. Carefully slide the PXI board to the rear of the chassis, make sure that the ejector handles are pushed **out** (as shown in Figure 3-2).

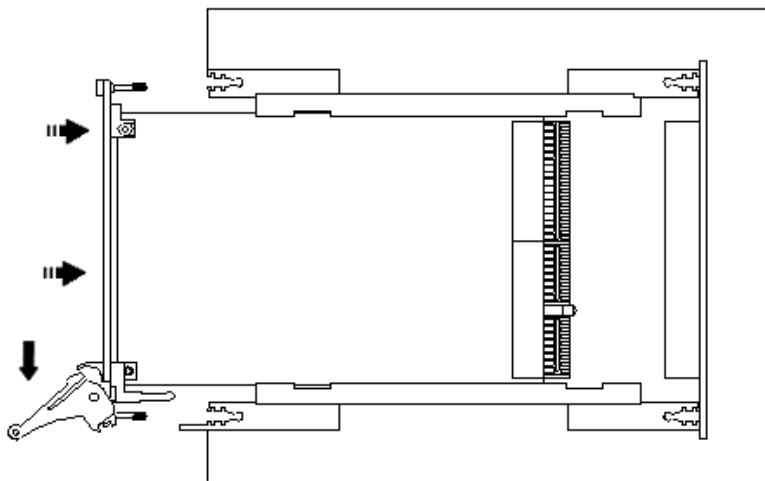


Figure 3-2: Ejector handles position during module insertion

5. After you feel resistance, push in the ejector handles as shown in Figure 3-3 to secure the module into the frame.

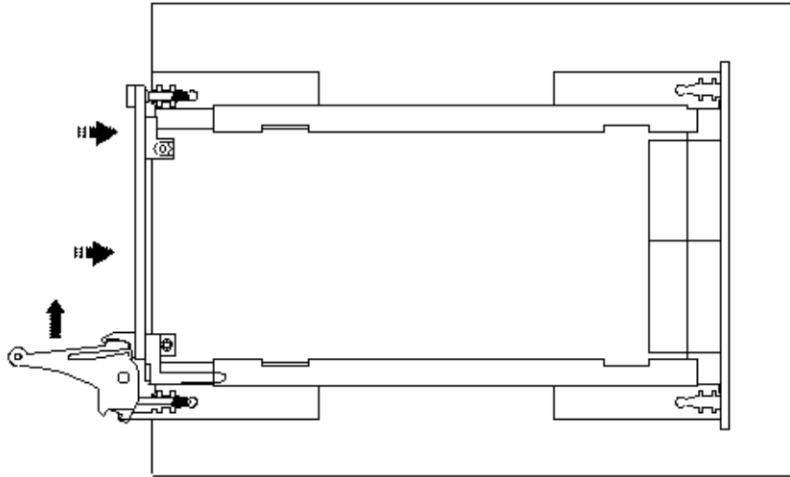


Figure 3-3: Ejector handles position after module insertion

6. Tighten the module's front panel to the chassis to secure the module in.
7. Connect any necessary cables to the board.
8. Plug the power cord in and turn on the PXI chassis.

Plug & Play Driver Installation

Plug & Play operating systems such as Windows notifies the user that a new board was found using the **New Hardware Found** wizard after restarting the system with the new board.

If another Geotest board software package was already installed, Windows will suggest using the driver information file: HW.INF. The file is located in your Program Files\Geotest\HW folder. Click **Next** to confirm and follow the instructions on the screen to complete the driver installation.

If the operating system was unable to find the driver (since the driver was not installed prior to the board installation), you may install the software as described in the prior section, then click on the **Have Disk** button and browse to select the HW.INF file located in C:\Program File\Geotest\HW.

If you are unable to locate the driver click **Cancel** to the found New Hardware wizard and exit the New Hardware Found Wizard, install the GXPXI driver, reboot your computer and repeat this procedure.

The Windows Device Manager (open from the System applet from the Windows Control Panel) must display the proper board name before continuing to use the board software (no Yellow warning icon shown next to device). If the device is displayed with an error you can select it and press delete and then press F5 to rescan the system again and to start the New Hardware Found wizard.

Removing a Board

Remove the board as follows:

1. Turn off the PXI chassis and unplug the power cord.
2. Locate a PXI slot on the PXI chassis.
3. Disconnect and remove any cables/connectors connected to the board.
4. Un-tighten the module's front panel screws to the chassis.

5. Push out the ejector handles and slide the PXI board away from the chassis.
6. Optionally - uninstall the software by running the setup again (or from the Windows Control Pane, Programs and Features or Add Remove Programs applet) and selecting Remove/Uninstall.
7. Push out the ejector handles and slide the PXI board away from the chassis.

Connectors

Figure 3-4: GX7404 J3 Output Channels Connector shows the available GX7404 board connector with description:

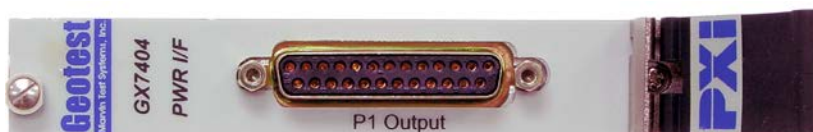


Figure 3-4: GX7404 J3 Output Channels Connector

Output Channels Connector (J3)

The following are the connector signal pin assignments for GX7404 module:

Pin#	Signal	Pin#	Signal	Pin#	Signal
1	+5V	10	GND	19	N/C
2	+5V	11	+3.3V	20	GND
3	N/C	12	+DC/DC	21	N/C
4	GND	13	Disable	22	GND
5	+12V	14	+5V	23	GND
6	N/C	15	+5V	24	-DC/DC
7	GND	16	N/C	25	GND
8	-12V	17	GND		
9	N/C	18	+12V		

Table 3-1: J3 – Output Channels Connector

Legend

+DC/DC - -DC/DC: On-board installed Channel.

Disable: External Disable input.

GND: Ground.

N/C: Not connected

Connectors and Accessories

The following accessories are available from Geotest for your switching board.

Part / Model Number	Description
GT25-CON	Connector, D-type 25 Male with solder pins.
GT25-MCON	Military Grade Connector, D-type 25 Male with solder pins.

Table 3-2: Spare Connectors and Accessories

Installation Folders

The GXPXI driver files are installed in the default directory C:\Program Files\Geotest\GXPXI. You can change the default GXPXI directory to one of your choosing at the time of installation.

During the installation, GXPXI Setup creates and copies files to the following directories:

Name	Purpose / Contents
...\Geotest\GXPXI	The GXPXI directory. Contains panel programs, programming libraries, interface files and examples, on-line help files and other documentation.
...\Geotest\HW	HW device driver. Provide access to your board hardware resources such as memory, IO ports and PCI board configuration. See the README.TXT located in this directory for more information.
...\ATEasy\Drivers	ATEasy drivers directory. GXPXI Driver and example are copied to this directory only if <i>ATEasy</i> is installed to your machine.
Windows System Folders	Windows System directory. Contains the GXPXI.DLL or GXPXI64.DLL driver HW driver shared files and some upgraded system components, such as the HTML help viewer, etc.

Driver Files Description

The Setup program copies the GXPXI driver, a panel executable, the GXPXI help file, the README.TXT file, and driver samples. The following is a brief description of each installation file:

Driver File and Virtual Panel

- GXPXI.DLL and GXPXI64.DLL - 32/64-Bit MS-Windows DLL for applications running under Windows
- GXPXIPANEL.EXE and GXPXIPANEL64.EXE– An instrument front panel program for all GXPXI supported boards.

Interface Files

The following GXPXI interface files are used to support the various development tools:

- GXPXI.H - header file for accessing the DLL functions using the C/C++ programming language. The header file compatible with the following 32 bit development tools:
 - Microsoft Visual C++, Microsoft Visual C++ .NET
 - Borland C++
- GXPXI.LIB/GXPXI64.LIB - Import library for GXPXI.DLL/GXPXI64.DLL (used when linking C/C++ application).
- GXPXIBC.LIB - Import library for GXPXI.DLL (used when linking Borland C/C++ application that uses GXPXI.DLL).
- GXPXI.PAS - interface file to support Borland Pascal Borland Delphi.
- GXPXI.BAS - Supports Microsoft Visual Basic 4.0, 5.0 and 6.0.
- GXPXI.VB - Supports Microsoft Visual Basic .NET.
- GXPXI.CS – Supports Microsoft Visual C#
- GX7404.DRV - ATEasy driver File for Gx7404 and GXPXI Virtual Panel Program
- GXPXI.LLB – LabView library

On-line Help and Manual

GXPXI.CHM – On-line version of the GX7404 User's Guide. The help file is provided in a Windows Compiled HTML help file (.CHM). The file contains information about the GX7404 board, programming reference and panel operation.

GX7404.PDF – On line, printable version of the GX7404 User's Guide in Adobe Acrobat format. To view or print the file you must have the reader installed. If not, you can download the Adobe Acrobat reader (free) from <http://www.adobe.com>.

ReadMe File

README.TXT – Contains important last minute information not available when the manual was printed. This text file covers topics such as a list of files required for installation, additional technical notes, and corrections to the GXPXI manuals. You can view and/or print this file using the Windows NOTEPAD.EXE or any other text file editors.

Example Programs

The sample program includes a C/C++ sample compiled with various development tools, Visual Basic example and an ATEasy sample. Other examples may be available for other programming tools.

Microsoft Visual C++ .NET example files:

- GxPxiExampleC.cpp - Source file
- GxPxiExampleC.ico - Icon file
- GxPxiExampleC.rc - Resource file
- GxPxiExampleC.vcproj - VC++ .NET project file
- GxPxiExampleC.exe - Example executable

Microsoft Visual C++ 6.0 example files:

- GxPxiExampleC.cpp - Source file
- GxPxiExampleC.ico - Icon file
- GxPxiExampleC.rc - Resource file
- GxPxiExampleC.dsp - VC++ project file
- GxPxiExampleC.exe - Example executable

Borland C++ example files:

- GxPxiExampleC.cpp - Source file
- GxPxiExampleC.ico - Icon file
- GxPxiExampleC.rc - Resource file
- GxPxiExampleC.bpr - Borland project file
- GxPxiExampleC.exe - Example executable

Microsoft Visual Basic .NET example files:

- GxPxiExampleVB.vb - Example form.
- GxPxiExampleVB.resx - Example form resource.
- GxPxiExampleVBapp.config - Example application configuration file.
- GxPxiExampleVBAssemblyInfo.vb - Example application assembly file
- GxPxiExampleVB.vbproj - Project file
- GxPxiExampleVB.exe - Example executable

Microsoft Visual C# example files:

- GXPxiExampleCS.cs - Example source
- GXPxiExampleCS.csproj - Project file
- GXPxiExampleCS.exe - Example executable

Microsoft Visual Basic 6.0 example files:

- GxPxiExampleVB6.frm - Example form
- GxPxiExampleVB6.frx - Example form binary file
- GxPxiExampleVB6.vbp - Project file
- GxPxiExampleVB6.exe - Example executable.

ATEasy driver and examples files (ATEasy Drivers directory):

- Gx7404.prj - example project

- Gx7404.sys - example system
- G x7404.prg - example program

Setup Maintenance Program

You can run Setup again after GX7404 has been installed from the original disk or from the Windows Control Panel – Add Remove Programs applet. Setup will be in the Maintenance mode when running for the second time. The Maintenance window shown below allows you to modify the current GX7404 installation. The following options are available in Maintenance mode:

- **Modify.** When you want to add or remove GXPXI components.
- **Repair.** When you have corrupted files and need to reinstall.
- **Remove.** When you want to completely remove GXPXI.

The Maintenance mode screen is shown below:

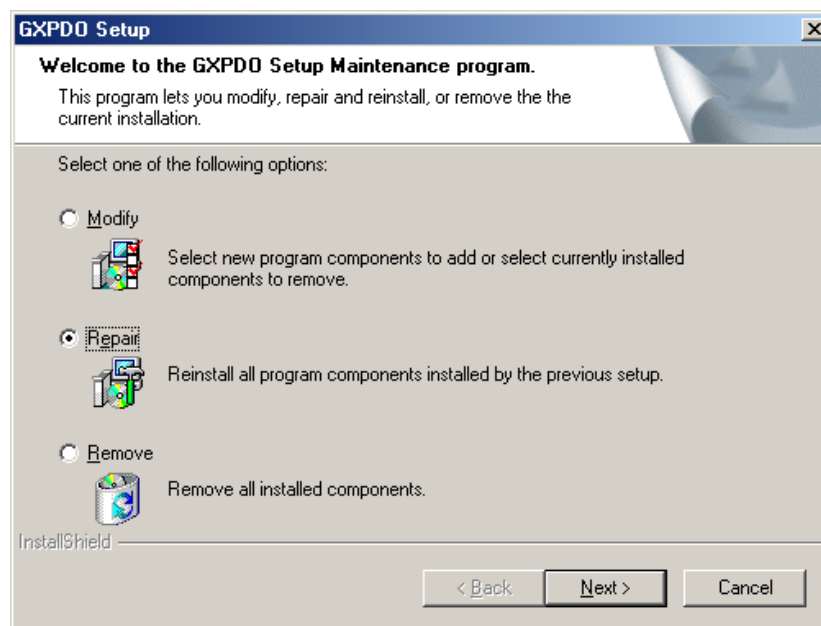


Figure 3-5: Setup Maintenance Window

Select one of the options and click **Next**.

Follow the instruction on the screen until Setup is complete.

Chapter 4 - Programming the Board

This chapter contains information about how to program the switching instruments using the GXPXI driver. The GXPXI driver contains functions to initialize, reset, and control the switching instruments. A brief description of the functions, as well as how and when to use them, is included in this chapter. Chapter 5 and the specific instrument User's Guide contain a complete and detailed description of the available programming functions.

The GXPXI driver supports many development tools. Using these tools with the driver is described in this chapter. In addition, the GXPXI directory contains examples written for these development tools. Refer to Chapter 3 for a list of the available examples.

An example using the DLL driver with Microsoft Visual C++ is at the end of this chapter. The driver functions and parameters are identical for all operating systems and development tools, and this example serves as a guide for use other programming languages.

The GXPXI Driver

The driver is a 32-bit Windows DLL file: GXPXI.DLL and 64-bit DLL: GXPXI64.DLL. The DLL is used with 32-bit or a 64-bit applications running under Windows. The HW device driver is installed by the setup program and is shared by other Geotest products (ATEasy, GXPIO, GXSW, etc). The DLLs can also use VISA (provided by a third party) to access the board hardware instead of the provided HW driver.

The DLL can be used with various development tools such as Microsoft Visual C++, Borland C++ Builder, Microsoft Visual Basic, Borland Pascal or Delphi, ATEasy and more. The following paragraphs describe how to create an application that uses the driver with various development tools. Refer to the paragraph describing the specific development tool for more information.

Programming Using C/C++ Tools

The following steps are required to use the GXPXI driver with C/C++ development tools:

- Include the GXPXI.H header file in the C/C++ source file that uses the GC2200/GTX2200 function. This header file is used for all driver functions. The file contains function prototypes and constant declarations to be used by the compiler for the application.
- Add the required .LIB file to the projects. This can be the import library GXPXI.LIB for Microsoft Visual C++ for 32-bit applications, GXPXI64.LIB for 64-bit applications and GXPXIBC.LIB for Borland C++. Windows- based applications that explicitly load the DLL by calling the Windows **LoadLibrary** API should not include the .LIB file in the project.
- Add code to call the GXPXI library functions as required by the application.
- Build the project.
- Run, test, and debug the application.

Programming Using Visual Basic

To use the driver with Visual Basic 4.0, 5.0 or 6.0 (for 32-bit applications), the user must include the GXPXI.BAS to the project. Visual Basic .NET developers must use the GXPXI.VB.

The file can be loaded within the integrated development environment (IDE) using *Add File* from the Visual Basic IDE *File menu*. The GXPXI.BAS/.VB contains function declarations for the DLL driver.

Programming Using Pascal/Delphi

To use the driver with Borland Pascal or Delphi, the user must include the GXPXI.PAS to the project. The GXPXI.PAS file contains a **unit** with function prototypes for the DLL functions. Include the GXPXI unit in the **uses** statement before making calls to the GXPXI functions.

Programming Using ATEasy®

The GXPXI package supplies a separate ATEasy driver for the PXI counters. The ATEasy driver uses the same GXPXI.DLL to program the board for all development environments. In addition, there is an example project that contains a program and a system file pre-configured with the ATEasy driver. Use the property dialog to change the driver shortcut property in the System Drivers sub-module and change the PCI slot number to reflect your current installation before running the example.

The ATEasy driver plain-languages commands are easier to use than using the DLL functions directly. The driver commands will generate an exception if the function call fails. That allows the ATEasy application to trap errors without directly checking the status code returned by the DLL function. The driver commands check the status parameter after each function call.

The ATEasy driver contains commands that are similar to the DLL functions in name and parameters, with the following exceptions:

- The driver handles the *nHandle* parameter automatically. ATEasy handles board instances as driver logical names i.e. PS1, PS2 for GX7404. Direct access to the parameter is possible using the property operator (the “dot” operator) once the parameter is set public in the ATEasy driver.
- The driver handles the *nStatus* parameter automatically. The Get Status command in the ATEasy Driver provides access to the board’s status. After calling a DLL function, the ATEasy driver will check the returned status and will call the CheckError procedure. An error status generates an exception that can be easily trapped by the application using the **OnError** module event or using the **try-catch** statement. ATEasy will notify the user or developer with an error dialog if the condition is not handled by other ATEasy code. Direct access to the parameter is possible using the property operator (the “dot” operator) once the parameter is set public in the ATEasy driver.

Some ATEasy drivers contain additional commands to permit easier access to the board features. For example, parameters for a function may be omitted by using a command item instead of typing the parameter value. The plain-language commands are self-documenting. Their syntax is similar to an English language statement. In addition, you can use the commands from the code editor context menu or by using the ATEasy’s code completion feature instead of typing them directly.

Programming Using LabView and LabView/Real Time

To use the driver with LabView use the provided lab view library GXPXI.llb. The library is located in the GXPXI folder. An example for LabView is also provided in the Examples folder. A DLL located in the LabViewRT folder can be used for deployment with LabView/Real-Time.

Using and Programming under Linux

Geotest provides a separate software package with a Linux driver (Geotest Drivers Pack for Linux). The software package can be download from the Geotest website. See the ReadMe.txt in that package for more information regarding using and programming the driver under Linux.

Using the GXPXI driver functions

The GXPXI driver contains a set of functions for the GX7404. Functions names that starts with the **GxPXi** prefix applies to all GXPXI boards (i.e. **GxPxiGetDriverSummary**). The GXPXI functions are designed with consistent set of arguments and functionality. All boards have a function that initializes the GXPXI driver for a specific board, reset the board, and display the virtual panel. All the functions use handles to identify and reference a specific board and all functions return status and share the same functions to handle error codes.

Initialization, HW Slot Numbers and VISA Resource

The GXPXI driver supports two device drivers HW and VISA which are used to initialize, identify and control the board. The user can use the **Gx7404Initialize** to initialize the board 's driver using HW and **Gx7404InitializeVisa** to initialize using VISA. The following describes the two different methods used:

1. **Geotest's HW** - the default device driver that is installed by the GXPXI driver. To initialize and control the board using the HW use the **Gx7404Initialize**(*nSlot*, *pnHandle*, *pnStatus*) function. The function initializes the driver for the board at the specified PXI slot number (*nSlot*) and returns a board handle. The **PXI/PCI Explorer** applet in the Windows Control Panel displays the PXI slot assignments. You can specify the *nSlot* parameter in the following way:
 - A combination of chassis number (chassis # x 256) with the chassis slot number, e.g. 0x105 for chassis 1 and slot 5. Chassis number can be set by the **PXI/PCI Explorer** applet.
 - Legacy nSlot as used by earlier versions of HW/VISA. The slot number contains no chassis number and can be changed using the **PXI/PCI Explorer** applet: 23 in this example.

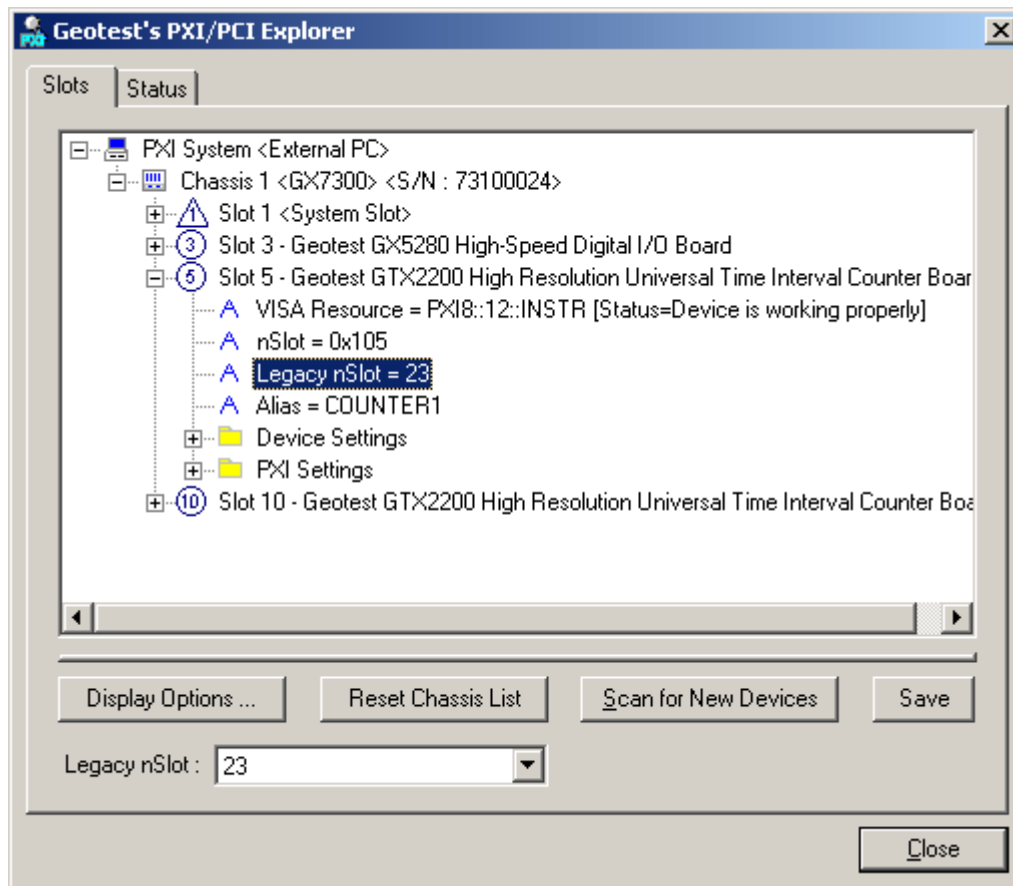


Figure 4-1: PXI/PCI Explorer

2. **VISA** – this is a third party library usually by National Instruments (NI-VISA). You must ensure that the VISA installed supports PXI and PCI devices (not all VISA providers supports PXI/PCI). GXPXI setup installs a VISA compatible driver for the GXPXI board in-order to be recognized by the VISA provider. Use the GXPXI function **Gx7404InitializeVisa** (*szVisaResource*, *pnHandle*, *pnStatus*) to initialize the driver board using VISA. The first argument *szVisaResource* is a string that is displayed by the VISA resource manager such as NI **Measurement and Automation** (NI_MAX). It is also displayed by Geotest **PXI/PCI Explorer** as shown in the prior figure. The VISA resource string can be specified in several ways as the following examples:

- Using chassis, slot: “PXI0::CHASSIS1::SLOT5”
- Using the PCI Bus/Device combination: “PXI9::13::INSTR” (bus 9, device 9).
- Using alias: “COUNTER1”. Use the PXI/PCI Explorer to set the device alias.

Information about VISA is available at <http://www.pxisa.org>.

The **Gx7404Initialize** function returns a handle that is required with other driver functions to program the board. This handle is usually saved in the program in a global variable for later use when calling other functions. The initialize function does not change the state of the board or its settings.

Board Handle

The board handle argument, *nHandle*, passed (by reference) to the parameter *pnHandle* of the **Gx7404Initialize** or the **Gx7404InitializeVisa** functions is a short integer (16 bits) number. It is used by the GXPXI driver functions to identify the board being accessed by the application. Since the driver supports many boards at the same time, the *nHandle* argument is required to uniquely identify which board is being programmed.

The *nHandle* is created when the application calls the **Gx7404Initialize** function. But there is no need to destroy the handle. Calling **Gx7404Initialize** with the same slot number will return the same handle.

Once the board is initialized the handle can be used with other functions to program the board.

Reset

The Reset function causes the driver to change all settings to their default state. The application software issue a Reset after the initializing the Counter, but a Reset can be issued any time. All counter boards have the **Gx7404Reset**(*nHandle*, *nStatus*) function. See the Function Reference for more information regarding the specific board.

Error Handling

All GXPXI functions pass a fail or success status - *pnStatus* - in the last parameter. A successful function call passes zero in the status parameter upon return. If the status is non-zero, then the function call fails. This parameter can be later used for error handling. When the status is error, the program can call the **GxPxiGetErrorString** function to return a string representing the error. The **GxPxiGetErrorString** reference contains possible error numbers and their associated error strings.

Driver Version

The **GxPxiGetDriverSummary** function can be used to return the current GXPXI driver version. It can be used to differentiate between the driver versions. See the Function Reference for more information.

Panel

Calling the **Gx7404Panel** will display the instrument's front panel dialog window. The panel can be used to initialize and control the board interactively. The panel function may be used by the application to allow the user to directly interact with the board.

The **Gx7404Panel** function is also used by the GXPXIPANEL.EXE panel program that is supplied with this package and provides a stand-alone Windows application that displays the instrument panel.

Distributing the Driver

Once the application is developed, the driver files (GXPXI.DLL or GXPXI64.DLL) and the HW device driver files located in the HW folder) can be shipped with the application. Typically, the DLLs should be copied to the Windows System directory. The HW device driver files should be installed using a special setup program HWSETUP.EXE that is provided with GXPXI driver files. Alternatively, you can provide the GXPXI disk to be installed along with the board.

Sample Programs

The following example demonstrates how to program the board using the C programming language under Windows. The example shows how to close or open a relay.

To run, Enter the following command line:

GxPxiExample <PciSlot> <operation> <channel | dac | rail> <rail_number | voltage | rail_source>

Where:

<PciSlot>	PCI/PXI Explorer slot number where the board reside.
<operation>	Operation code: SCR=Set Channel Rail GCR=Get Channel Rail SDV=Set Dac Voltage GDV=Get Dac Voltage SRS=Set Rail Source
< channel dac rail >	Depends on the operation: Channel number: 0-7 for SCR/GCR operations DAC number: 0-2 for SDV/GDV operations Rail number: 0-2 for SRS/GRS operations
<rail_numbe voltage rail_source>	Depends on operation: Rail_number: 0 to 2 Voltage: -10V to 32V Rail_source: 0 to 4

Sample Program Listing

```

/*****

FILE      : GxPxiExampleC.cpp

PURPOSE : WIN32/LINUX example program for GX7404 boards
          using the GXPXI driver.

CREATED  : June 2004

COPYRIGHT: Copyright 2004-2012 GEOTEST - MTS Inc.

COMMENTS:

To compile the example:

1. Microsoft VC++
Load GxPxiExampleC.dsp, .vcproj or .mak, depends on
the VC++ version from the Project/File/Open... menu
Select Project/Rebuild all from the menu

2. Borland C++ Builder
Load GxPxiExampleC.bpr from the Project/Open
Project... menu
Select Project/Build all from the menu

3. Linux (GCC for CPP and Make must be available)
make -fGxPxiExampleC.mk [CFG=Release[64] | Debug[64]]
[rebuild | clean]

*****/
#ifdef __GNUC__
#include "windows.h"
#endif
#include "GxPxi.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#ifdef __BORLANDC__
#pragma hdrstop
#include <condefs.h>
USELIB("GxPxiBC.lib");
USERC("GxPxiExampleC.rc");
#endif

//*****
//      DisplayMsg
//*****
void DisplayMsg(PCSTR lpszMsg)
{
#ifdef __GNUC__
    MessageBeep(0);
    MessageBox(0, lpszMsg, "GxPxi example program", MB_OK);
#else
    printf("\r\nGxPxi example program: %s\r\n", lpszMsg);
#endif
    return;
}

//*****
//      __strupr
//*****
char * __strupr(char * sz)
{
    int i;

```

```

    for (i=0; sz[i]; i++)
        sz[i]=toupper(sz[i]);
    return sz;
}

//*****
//          DispUsage
//*****
void DispUsage(void)
{
    DisplayMsg(
        "\r\nThis example shows how to "
        "Close or Open GXPXI board relays \r\n"

        "Usage:\r\n"
        "GxPxiExample <model> <slot_number> <specif_command>"
        "\r\n\r\nWhere:\r\n"
        "<model> can be"
        "\t7404\r\n"
        "<slot_number>"
        "\t(example : 1)\r\n"
        "<specif_command> depends on the model:\r\n"

        "\t7404 - C | O\r\n"
        "\t\tWhere: C=Close, O=Open\r\n"
        "\t\tChannel#=0-4 relay number\r\n\r\n"

        "Example:\r\n"
        "GxPxiExample 7404 3 C 1\r\n"
        "Closes relay number 1 on the GX7404 reside at slot 3\r\n"

        "\r\nTo change command line under Windows \r\n"
        "Right click on the example shortcut from the start
        menu and type the new command line\r\n"
    );
    exit(1);
}

//*****
//          CheckStatus
//*****
void CheckStatus(SHORT nStatus)
{
    CHAR    sz[256];

    if (!nStatus) return;
    GxPxiGetErrorString(nStatus, sz, sizeof sz, &nStatus);
    DisplayMsg(sz);
    DisplayMsg("Aborting the program...");
    exit(nStatus);
}

//*****
//  MAIN
//
//  This main function receives four to five parameters:
//
//  GXPXI board type (e.g. 7404=GX7404)
//  GXPXI board PXI slot number(e.g. 1)
//  GXPXI switching operation(e.g. C=Close relay, O=Open relay)
//  for GXPXI GX7404 channel #(0-4)
//
//*****
int main(int argc, char **argv)
{
    short  nBoardType;    // Board Type
    short  nSlotNum;      // Board PXI slot number
    short  nChannel;      // Channel number
    char*  cOperation;    // Board Operation
    short  nHandle;       // Board handle
    short  nStatus;       // Returned status

```

```

// ** Check number of arguments rcvd
if (argc<4) DispUsage();

// ** Parse command line parameters
nBoardType=(SHORT)strtol(++argv, NULL, 0);
nSlotNum=(SHORT)strtol(++argv, NULL, 0);
cOperation=__strpr(++argv);

if (cOperation[0]!='C' && cOperation[0]!='O')
    DispUsage();

// ** Get channel, initialize, and perform operation on //GXPXI card
switch (nBoardType)
{
    case 7404:
        nChannel=(SHORT)strtol(++argv, NULL, 0);
        Gx7404Initialize(nSlotNum, &nHandle, &nStatus);
        CheckStatus(nStatus);

        switch (cOperation[0])
        {
            case 'C':
                Gx7404ConnectChannel(nHandle, nChannel,\ true, &nStatus);
                CheckStatus(nStatus);
                DisplayMsg("You have closed a relay");
                break;
            case 'O':
                Gx7404ConnectChannel(nHandle, nChannel,\
                false, &nStatus);
                CheckStatus(nStatus);
                DisplayMsg("You have opened a relay");
                break;
            default:
                break;
        }
        break;
    default:
        DispUsage();
}

return 0;
}

//*****
//          End Of File
//*****

```

Chapter 5 - Functions Reference

Introduction

The GX7404 driver functions reference chapter is organized in alphabetical order. Each function description contains the function name, purpose, syntax, parameters description and type followed by a Comments, an Example (written in C), and a See Also sections.

All function parameters syntax follows the same rules:

- Strings are ASCIIZ (null or zero character terminated).
- The first parameter of most functions is *nHandle* (16-bit integer). This parameter is required for operating the board and is returned by the **Gx7404Initialize** function. The *nHandle* is used to identify the board when calling a function for programming and controlling the operation of that board.
- All functions return a status with the last parameter named *pnStatus*. The *pnStatus* is zero if the function was successful, or less than a zero on error. The description of the error is available using the **GxPxiGetErrorString** function or by using a predefined constant, defined in the driver interface files: GXPXI.H, GXPXI.BAS, GXPXI.PAS or GX7404.DRV.
- Parameter name are prefixed as follows:

Prefix	Type	Example
a	Array - prefix this before the simple type.	anArray (Array of Short)
b	BOOL – Boolean, 0 for FALSE; <>0 for TRUE	bUpdate
d	DOUBLE - 8 bytes floating point	dReading
dw	DWORD - double word (unsigned 32-bit)	dwTimeout
hwnd	Window handle (32-bit integer).	hwndPanel
l	LONG - (signed 32-bit)	lBits
n	SHORT - (signed 16-bit)	nMode
p	Pointer - Usually used to return a value. Prefix this before the simple type.	pnStatus
sz	Null - (zero value character) terminated string	szMsg
uc	BYTE - (8 bits) unsigned.	ucValue
w	WORD - Unsigned short (unsigned 16-bit)	wParam

Table 5-1: Parameter Name Prefixes

GX7404 Functions

The following list is a summary of functions available for the GX7404:

Driver Functions	Description
General Driver Functions	
Gx7404GetBoardSummary	Returns the board summary.
Gx7404Initialize	Initializes the GX7404 driver for the specified PXI slot.
Gx7404InitializeVisa	Initializes the GX7404 driver using VISA
Gx7404Panel	Opens a virtual panel used to interactively control the GX7404 board.
Gx7404Reset	Opens all boards' channels and all connections between groups and to External Coax.
Voltage Channel interface Functions:	
Gx7404ConnectChannel	Connects or disconnect the specified channel to the output connector.
Gx7404ConnectChannels	Connects or disconnect all channels to or from the output connector.
Gx7404GetChannel	Returns the status of the specified voltage channel connection.
Gx7404GetChannelCurrent	Returns the specified channel current reading in Amperes
Gx7404GetChannelVoltage	Returns the specified channel voltage reading.
Gx7404GetChannels	Returns the connection state of all channels.
Gx7404GetExternalDisableState	Returns the External disable input line state.
Gx7404GetOnBoardChannelType	Returns the On-Board installed channel type.
General Functions	
GxPxiGetDriverSummary	Returns the driver name and version.
GxPxiGetErrorString	Returns the error string associated with the specified error number.

Gx7404ConnectChannel

Purpose

Connects or disconnect the specified channel to the output connector.

Syntax

Gx7404ConnectChannel (*nHandle*, *nChannel*, *bOn*, *pnStatus*)

Parameters

Name	Type	Comments
<i>nHandle</i>	SHORT	Handle to a GX7404 board.
<i>nChannel</i>	SHORT	Channel number: <ul style="list-style-type: none"> • 0 = GX7404_CHANNEL_PLUS5 • 1 = GX7404_CHANNEL_PLUS3_3 • 2 = GX7404_CHANNEL_PLUS12 • 3 = GX7404_CHANNEL_NEGATIVE12 • 4 = GX7404_CHANNEL_INTERNAL
<i>bOn</i>	BOOL	Connecting modes are as follow: <ul style="list-style-type: none"> • 0 = FALSE, disconnect the specified channel. • 1 = TRUE, connect the specified channel.
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Channel numbers 0-3 are the four voltages sources from the PXI back plan and channel 4 is the onboard internal voltage source.

Example

The following example connects the +5 volts.

```
SHORT nStatus;
Gx7404ConnectChannel (nHandle, GX7404_CHANNEL_PLUS5, TRUE, &nStatus);
```

See Also

Gx7404ConnectChannels, **Gx7404GetChannel**, **Gx7404GetChannels**, **Gx7404GetChannelVoltage**, **Gx7404GetChannelCurrent**, **GxPxiGetErrorString**

Gx7404ConnectChannels

Purpose

Connects or disconnect all channels to or from the output connector.

Syntax

Gx7404ConnectChannels (*nHandle*, *ucState*, *pnStatus*)

Parameters

Name	Type	Comments
<i>NHandle</i>	SHORT	Handle to a GX7404 board.
<i>UcState</i>	BYTE	The state of all 5 channels. Each bit corresponds to a channel, see comments for details.
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The *ucState* parameter sets the state of all channels, the following table describes the *ucState* bit description:

Bit #	Description
0	Plus 5.0V.
1	Plus 3.3V.
2	Plus 12.0V.
3	Negative 12.0V.
4	Onboard Voltage source.

Example

The following example connects the +5V and +12V.

```
SHORT nStatus;
Gx7404ConnectChannels (nHandle, 0x5, &nStatus);
```

See Also

Gx7404GetChannels, **Gx7404GetChannel**, **Gx7404GetChannelVoltage**, **Gx7404GetChannelCurrent**, **GxPxiGetErrorString**

Gx7404GetBoardSummary

Purpose

Returns the board summary.

Syntax

Gx7404GetBoardSummary (*nHandle*, *szSummary*, *nSumMaxLen*, *pnStatus*)

Parameters

Name	Type	Comments
<i>NHandle</i>	SHORT	Handle to a GX7404 board.
<i>SzSummary</i>	PSTR	Buffer to contain the returned board info (null terminated) string.
<i>NSumMaxLen</i>	SHORT	Size of the buffer to contain the board info string.
<i>PnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The GX7404 summary string provides the following data from the on-board EEPROM in the order shown:

GX7404, FPGA-Version:0x0001, PCB-Rev:A, S/N 7404003, On-Board Channel: Fixed 5V , Cal.
date: Tue Jun 22 19:01:27 2004

See Also

Gx7404GetOnBoardChannelType , **GxPxiGetDriverSummary**, **GxPxiGetErrorString**

Gx7404GetChannel

Purpose

Returns the status of the specified voltage channel connection.

Syntax

Gx7404GetChannel (*nHandle*, *nChannel*, *pbOn*, *pnStatus*)

Parameters

Name	Type	Comments
<i>nHandle</i>	SHORT	Handle to a GX7404 board.
<i>nChannel</i>	SHORT	Channel number: <ul style="list-style-type: none"> • 0 = GX7404_CHANNEL_PLUS5 • 1 = GX7404_CHANNEL_PLUS3_3 • 2 = GX7404_CHANNEL_PLUS12 • 3 = GX7404_CHANNEL_NEGATIVE12 • 4 = GX7404_CHANNEL_INTERNAL
<i>pbOn</i>	PBOOL	Returned connecting modes are as follow: <ul style="list-style-type: none"> • 0 = FALSE, disconnect the specified channel. • 1 = TRUE, connect the specified channel.
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Channel numbers 0-3 are the four voltages sources from the PXI back plan and channel 4 is the onboard internal voltage source.

Example

The following example returns the connection of the +5 volts channel.

```
SHORT nStatus;
BOOL bOn;
Gx7404GetChannel (nHandle, GX7404_CHANNEL_PLUS5, TRUE, &bOn &nStatus);
```

See Also

Gx7404ConnectChannel, **Gx7404ConnectChannels**, **Gx7404GetChannels**, **Gx7404GetChannelVoltage**, **Gx7404GetChannelCurrent**, **GxPxiGetErrorString**

Gx7404GetChannelCurrent

Purpose

Returns the specified channel current reading in Amperes.

Syntax

Gx7404GetChannelCurrent (*nHandle*, *nChannel*, *pdCurrent*, *pnStatus*)

Parameters

Name	Type	Comments
<i>nHandle</i>	SHORT	Handle to a GX7404 board.
<i>nChannel</i>	SHORT	Channel number: <ul style="list-style-type: none"> • 0 = GX7404_CHANNEL_PLUS5 • 1 = GX7404_CHANNEL_PLUS3_3 • 2 = GX7404_CHANNEL_PLUS12 • 3 = GX7404_CHANNEL_NEGATIVE12 • 4 = GX7404_CHANNEL_INTERNAL
<i>pdCurrent</i>	PDOUBLE	Returned channel's current in Amperes
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Channel numbers 0-3 are the four voltages sources from the PXI back plan and channel 4 is the onboard internal voltage source.

Example

The following example returns the +5V channel current.

```
SHORT nStatus;
DOUBLE dCurrent;
Gx7404GetChannelCurrent (nHandle, GX7404_CHANNEL_PLUS5, &dCurrent, &nStatus);
```

See Also

Gx7404ConnectChannel, **Gx7404ConnectChannels**, **Gx7404GetChannels**, **Gx7404GetChannelVoltage**, **GxPxiGetErrorString**

Gx7404GetChannelVoltage

Purpose

Returns the specified channel voltage reading.

Syntax

Gx7404GetChannelVoltage (*nHandle*, *nChannel*, *pdVolts*, *pnStatus*)

Parameters

Name	Type	Comments
<i>nHandle</i>	SHORT	Handle to a GX7404 board.
<i>nChannel</i>	SHORT	Channel number: <ul style="list-style-type: none"> • 0 = GX7404_CHANNEL_PLUS5 • 1 = GX7404_CHANNEL_PLUS3_3 • 2 = GX7404_CHANNEL_PLUS12 • 3 = GX7404_CHANNEL_NEGATIVE12 • 4 = GX7404_CHANNEL_INTERNAL
<i>pdVolts</i>	PDOUBLE	Returned channel's voltage.
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Channel numbers 0-3 are the four voltages sources from the PXI back plan and channel 4 is the onboard internal voltage source.

Example

The following example returns the +5V channel voltage.

```
SHORT nStatus;
DOUBLE dVolts;
Gx7404GetChannelVoltage (nHandle, GX7404_CHANNEL_PLUS5, &dVolts, &nStatus);
```

See Also

Gx7404ConnectChannel, **Gx7404ConnectChannels**, **Gx7404GetChannels**, **Gx7404GetChannelCurrent**, **GxPxiGetErrorString**

Gx7404GetChannels

Purpose

Returns the connection state of all channels.

Syntax

Gx7404GetChannels (*nHandle*, *pucState*, *pnStatus*)

Parameters

Name	Type	Comments
<i>nHandle</i>	SHORT	Handle to a GX7404 board.
<i>pucState</i>	PBYTE	Returned state of all 5 channels. Each bit corresponds to a channel, see comments for details.
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The *ucState* parameter represents the state of all channels, the following table describes the *ucState* bit description:

Bit #	Description
0	Plus 5..0V.
1	Plus 3.3V.
2	Plus 12.0V.
3	Negative 12.0V.
4	Onboard Voltage source.

Example

The following example returns the connection state of all channels.

```
SHORT nStatus;
BYTE ucState;
Gx7404GetChannels (nHandle, &ucState, &nStatus);
```

See Also

Gx7404ConnectChannel, **Gx7404ConnectChannels**, **Gx7404GetChannel**, **Gx7404GetChannelVoltage**, **Gx7404GetChannelCurrent**, **GxPxiGetErrorString**

Gx7404GetExternalDisableState

Purpose

Returns the External disable input line state.

Syntax

Gx7404GetExternalDisableState (*nHandle*, *pbHigh*, *pnStatus*)

Parameters

Name	Type	Comments
<i>nHandle</i>	SHORT	Handle to a GX7404 board.
<i>pbHigh</i>	PBOOL	External disable input line state can be as follow: 0. FALSE. 1. TRUE
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

When the External disable input line state is LOW, i.e. pbHigh=FALSE, the GX7404 will be reset. As long as this line is low non of the channels can be turned on.

Example

The following example returns the External disable input line state.

```
SHORT nStatus;
BOOL bHigh;
Gx7404GetExternalDisableState (nHandle, &bHigh, &nStatus);
```

See Also

Gx7404ConnectChannel, **Gx7404ConnectChannels**, **Gx7404GetChannel**, **Gx7404GetChannelVoltage**, **Gx7404GetChannelCurrent**, **GxPxiGetErrorString**, **Gx7404GetBoardSummary**, **GxPxiGetErrorString**

Gx7404GetOnBoardChannelType

Purpose

Returns the On-Board installed channel type.

Syntax

Gx7404GetOnBoardChannelType (*nHandle*, *pnType*, *pnStatus*)

Parameters

Name	Type	Comments
<i>nHandle</i>	SHORT	Handle to a GX7404 board.
<i>pnType</i>	PSHORT	Returned On-Board installed channel type. Channel type can be as follow: <ol style="list-style-type: none"> 0. GX7404_ONBOARD_CHANNEL_NOT_INSTALLED 1. GX7404_ONBOARD_CHANNEL_PLUS5 2. GX7404_ONBOARD_CHANNEL_PLUS3_3 3. GX7404_ONBOARD_CHANNEL_PLUS9 4. GX7404_ONBOARD_CHANNEL_PLUS12 5. GX7404_ONBOARD_CHANNEL_PLUS15
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Example

The following example returns the On-Board channel type.

```
SHORT nType, nStatus;
Gx7404GetOnBoardChannelType (nHandle, &nType, &nStatus);
```

See Also

Gx7404GetBoardSummary, **GxPxiGetErrorString**

Gx7404Initialize

Purpose

Initializes the GX7404 driver for the specified PXI slot.

Syntax

Gx7404Initialize (*nSlot*, *pnHandle*, *pnStatus*)

Parameters

Name	Type	Comments
<i>nSlot</i>	SHORT	GX7404 board slot number on the PCI bus.
<i>pnHandle</i>	PSHORT	Returned handle for the board. The handle is set to zero on error and $\neq 0$ on success.
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function returns a handle that can be used with other GX7404 functions to program the board. The function does not change any of the board settings.

The specified PCI slot number can be retrieved by the **PCI Explorer** applet that can be opened from the Windows Control Panel. You may also use the label on the chassis below the PCI slot where the board is installed.

Example

The following example initializes two GX7404 boards at slot 1 and 2.

```
SHORT  nHandle1, nHandle2, nStatus;

Gx7404Initialize(1, &nHandle1, &nStatus);
Gx7404Initialize(2, &nHandle2, &nStatus);
if (nHandle1==0 || nHandle2==0)
    printf("Unable to Initialize the board");
```

See Also

Gx7404Reset, **GxPxiGetErrorString**

Gx7404InitializeVisa

Purpose

Initializes the GX7404 driver for the specified PXI slot.

Syntax

Gx7404Initialize (*szVisaResource*, *pnHandle*, *pnStatus*)

Parameters

Name	Type	Comments
<i>szVisaResource</i>	SHORT	String identifying the location of the specified board in order to establish a session.
<i>pnHandle</i>	PSHORT	Returned handle for the board. The handle is set to zero on error and <> 0 on success.
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The **Gx7404InitializeVisa** opens a VISA session to the specified resource. The function uses the default VISA provider configured in your system to access the board. You must ensure that the default VISA provider support PXI/PCI devices and that the board is visible in the VISA resource manager before calling this function.

The first argument *szVisaResource* is a string that is displayed by the VISA resource manager such as NI Measurement and Automation (NI_MAX). It is also displayed by Geotest PXI/PCI Explorer as shown in the prior figure. The VISA resource string can be specified in several ways as follows:

- Using chassis, slot: "PXI0::CHASSIS1::SLOT5"
- Using the PCI Bus/Device combination: "PXI9::13::INSTR" (bus 9, device 9).
- Using alias: "COUNTER1". Use the PXI/PCI Explorer to set the device alias.

The function returns a board handle (session identifier) that can be used to call any other operations of that resource. The session is opened with VI_TMO_IMMEDIATE and VI_NO_LOCK VISA attributes. On terminating the application the driver automatically invokes **viClose()** terminating the session.

Example

The following example initializes a Gx7404 board at PXI bus 5 and device 11.

```
SHORT nHandle, nStatus;
Gx7404InitializeVisa("PXI5::11::INSTR", &nHandle, &nStatus);
if (nHandle==0)
{
    printf("Unable to Initialize the board")
    return;
}
```

See Also

Gx7404Initiazlize, **Gx7404Reset**, **GxPxiGetErrorString**

Gx7404Panel

Purpose

Opens a virtual panel used to interactively control the GX7404 board.

Syntax

Gx7404Panel (*pnHandle*, *hwndParent*, *nMode*, *phwndPanel*, *pnStatus*)

Parameters

Name	Type	Comments
<i>pnHandle</i>	PSHORT	Handle to a GX7404 board. This number may be zero if the board is to be initialized by the panel window.
<i>hwndParent</i>	HWND	Sets the panel parent window handle. A value of 0 sets the desktop as the parent window.
<i>nMode</i>	SHORT	The mode in which the panel main window is created. 0 for modeless and 1 for modal window.
<i>phwndPanel</i>	LPHWND	Returned window handle for the panel (for modeless panel only).
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function is used to create the virtual panel. The panel window may be opened as a modal or a modeless window, depending on the *nMode* parameters.

If the mode is set to modal dialog (*nMode*=1), the panel will disable the parent window (*hwndParent*) and the function will return only after the window is closed by the user. In that case the *pnHandle* returns the handle created by the user using the panel Initialize dialog. This handle then may be used when calling other GX7404 functions.

If a modeless dialog was created (*nMode*=0), the function returns immediately after creating the panel window, returning the window handle to the panel - *phwndPanel*. It is the responsibility of the calling program to dispatch window messages to this window, so that the window can respond to messages.

Example

The following example opens the panel in modal mode:

```
HWND    hwndPanel;
SHORT   nHandle=0, nStatus;
...
Gx7404Panel (&nHandle, 0, 1, &hwndPanel, &nStatus);
```

See Also

Gx7404Initialize, **GxPxiGetErrorString**

Gx7404Reset

Purpose

Open all boards' channels and all connections between groups and to External Coax.

Syntax

Gx7404Reset (*nHandle*, *pnStatus*)

Parameters

Name	Type	Comments
<i>nHandle</i>	SHORT	Handle to a GX7404 board.
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function is normally called after calling the **Gx7404Initialize** function, since the initialize function will not reset the board.

Example

The following example initializes the driver and then resets the board.

```
Gx7404Initialize(1, &nHandle, &nStatus);  
if (nStatus<0)  
    return nStatus;    // return error  
Gx7404Reset (nHandle, &nStatus);
```

See Also

Gx7404Initialize, **Gx7404SetGroupRelays**, **GxPxiGetErrorString**,

GxPxiGetDriverSummary

Purpose

Returns the driver name and version.

Syntax

GxPxiGetDriverSummary (*pszSummary*, *nSummaryMaxLen*, *pdwVersion*, *pnStatus*)

Parameters

Name	Type	Comments
<i>pszSummary</i>	PSTR	Buffer to the returned driver summary string.
<i>nSummaryMaxLen</i>	SHORT	The size of the summary string buffer.
<i>pdwVersion</i>	PDWORD	Returned version number. The high order word specifies the major version number where the low order word specifies the minor version number.
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The returned string is:

```
"GXPXI Driver for GX7404. Version 1.0, Copyright © Geotest 2004"
```

Example

The following example prints the driver version:

```
CHAR sz[128];
DWORD dwVersion;
SHORT nStatus;
GxPxiGetDriverSummary (sz, sizeof sz, &dwVersion, &nStatus);
printf("Driver Version %d.%d", (INT)(dwVersion>>16),
(INT)dwVersion &0xFFFF);
```

See Also

GxPxiGetErrorString

GxPxiGetErrorString

Purpose

Returns the error string associated with the specified error number.

Syntax

GxPxiGetErrorString (*nError*, *pszMsg*, *nErrorMaxLen*, *pnStatus*)

Parameters

Name	Type	Comments
<i>nError</i>	SHORT	Error number as returned by the <i>pnStatus</i> of any of the driver functions. See table below for possible values. The number should be a negative number, otherwise the function returns the “No error has occurred” string.
<i>pszMsg</i>	LPSTR	Buffer containing the returned error string (null terminated string).
<i>nErrorMaxLen</i>	SHORT	Size of the buffer <i>pszMsg</i> .
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function returns the error string associated with the *nError* as returned from other driver functions.

The following table displays the possible error values; not all errors apply to this board type:

Resource Errors

- 1 Board does not exist in this slot
- 2 Unable to open the HW device/Service
- 3 Different board exist in the specified PCI slot
- 4 PCI slot not configured properly. You may configure it by using the **PCIExplorer** from the Control Panel
- 5 Unable to register the PCI device
- 6 Unable to allocate system resource or memory for the PCI device
- 7 Too many boards
- 8 Unable to create panel
- 9 Unable to create a Windows timer

Parameter Errors

- 20 Invalid parameter
- 21 Invalid PCI slot number
- 22 Invalid board handle
- 23 Invalid channel number
- 24 Invalid bus number
- 25 Invalid mode

- 26 Invalid group number
- 27 Invalid string length
- 28 Invalid row number
- 29 Invalid column number
- 30 Invalid Relay number.
- 31 Invalid Daisy Chain Mode

Board specific parameter error

- 40 Invalid channel number
- 41 Invalid voltage number
- 42 Error: Voltage out of range
- 43 Channel busy timeout
- 44 Invalid channel number

Execution error

- 70 Voltage measurement timeout, check the specified channel's fuse
- 71 Current measurement timeout, check the specified channel's fuse

Miscellaneous Errors

- 99 Invalid or unknown error number

Example

The following example initializes the board. If the initialization failed, the following error string is printed:

```
CHAR    sz[256];
SHORT   nStatus, nHandle;
..
Gx7404Initialize(3, &Handle, &Status);
if (nStatus<0)
{
    GxPxiGetErrorString(nStatus, sz, sizeof sz, &nStatus);
    printf(sz);          // prints the error string returns
}
```

Index

.	ii
.NET	ii
3	
3U PXI.....	6
A	
Accessories	17
Architecture	1, 5
ATEasy	ii, 17, 18, 21, 22
B	
Block Diagram.....	5
Board Description.....	1, 4
Board Handle.....	24
Borland	18, 21, 22
Borland-Delphi	22
C	
C/C++	18, 21
C++	21
Connector	16
Connectors	15, 16, 17
Connectors and Accessories	17
Corrupt files.....	20
D	
Delphi	ii, 18, 21, 22
Distributing.....	25
Driver	
Directory.....	17
Files	17
Driver Version	24
E	
Error Handling.....	24
Error-Handling	24
Example.....	18
F	
Functions Reference	29

G

GX7404	1, 4
GX7404 Functions.....	30
Gx7404ConnectChannel.....	31
Gx7404ConnectChannels	32
Gx7404GetBoardSummary	33
Gx7404GetChannel	34
Gx7404GetChannelCurrent	35
Gx7404GetChannels.....	37
Gx7404GetChannelVoltage.....	36
Gx7404GetExternalDisableState	38
Gx7404GetOnBoardChannelType.....	39
Gx7404Initialize	23, 24, 40
Gx7404InitializeVisa	23, 24, 41
Gx7404Panel	24, 25, 42
Gx7404Reset	24, 43, 45
GXPXI.....	12
Driver-Description.....	21
Header-file	21
Help-File-Description	18
Panel-File-Description	18
GXPXI.BAS	18, 21
GXPXI.DLL	18
GXPXI.DLL	12, 18
GXPXI.DLL	18
GXPXI.DLL	21
GXPXI.DLL	22
GXPXI.EXE	12
GXPXI.H.....	18, 21
GXPXI.LIB	18
GXPXI.LIB	21
GXPXI.llb.....	22
GXPXI.PAS.....	18, 22
GXPXI.VB	18

GXPXI64.DLL	12, 21
GXPXI64.LIB	21
GXPXIBC.LIB	18
GxPxiGetDriverSummary	24, 44
GxPXIGetDriverSummary	23
GxPxiGetErrorString	24
GXPXIPANEL.EXE	25
GXPXIPANEL.EXE	18
H	
Handle	14, 15, 23, 24
HW	15, 17, 21, 25
I	
Installation:	14, 15
J	
J3	16
L	
LabView	22
LabView/Real Time	22
Linux	22
Listing	26
N	
<i>nHandle</i>	22, 24
O	
OnError	22
Output Channels Connector	16
Output Voltages	4
P	
Panel	7, 9, 18, 20, 23, 24
Pascal	18, 21, 22
PCI	17
Plug & Play	15
<i>pnStatus</i>	24

Program-File-Descriptions	18
Programming	
Borland-Delphi	22
Error-Handling	24
Panel-Program	25
Programming Using Visual Basic	21
PXI	3, 4, 7, 11, 13, 14, 15, 16, 23
PXI/PCI Explorer	7, 13, 23, 24, 41
R	
README.TXT	17, 18
Readme-File	18
Reset	24
S	
Sample	25, 26
Setup	12, 17, 18, 20
Setup Maintenance	20
Setup-and-Installation	11
Slot	11, 14, 15, 23
Specifications	1
Supported-Development-Tools	21
System	
Directory	17
System-Requirements	11
U	
Using the GXPXI driver functions	23
V	
Virtual Panel	7, 8, 9, 12, 18, 23
About Page	9
Setup Page	8
VISA	23, 24, 41
Visual Basic	ii, 21
Visual C++	ii, 18, 21