

# ***GX1034***

## **Standards Module PXI Board**

### ***User's Guide***

Last updated May 12, 2008

***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 in its original shipping box. Do not store boards on top of workbenches or other areas where they might be susceptible to damage or exposure to strong electromagnetic or electrostatic fields. Store circuit boards in protective anti-electrostatic wrapping and away from electromagnetic fields.

Be sure to make a single copy of the software CD for installation. Store the original CD in a safe place away from electromagnetic or electrostatic fields. Return compact disks (CD) to their protective case or sleeve and store in the original shipping box or other suitable location.

## Warranty

---

Geotest products are warranted against defects in materials and workmanship for a period of 12 months. 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. Revisions and new versions may be covered by a software support agreement. If you need to return a board, please contact Geotest Customer Technical Services Department via <http://www.geotestinc.com/magic/> - the Geotest on-line support system.

## 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 and application notes, as well as a download area for downloading patches, example, patches and new or revised instrument drivers. To submit a support issue including suggestion, bug report or questions 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 8:30 AM and 5:30 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 © 2003-2008 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®	Geotest – MTS Inc.
C++ Builder, Borland C++, Pascal and Delphi	Borland Corporation
Microsoft Developer Studio, Microsoft Visual C++, Microsoft Visual Basic,, .NET, Windows 95, 98, NT, ME, 2000, XP and VISTA.	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
<b>Chapter 1 - Introduction .....</b>	<b>1</b>
Manual Scope and Organization.....	1
Manual Scope.....	1
Manual Organization.....	1
Conventions Used in this Manual .....	1
<b>Chapter 2 - Overview .....</b>	<b>3</b>
Introduction.....	3
Features.....	3
Applications.....	4
Board Description .....	5
Architecture .....	8
Block Diagram .....	8
Specifications.....	9
Virtual Panel Description.....	11
Virtual Panel Initialize Dialog .....	12
Virtual Panel Setup Page.....	13
Virtual Panel About Page.....	14
<b>Chapter 3 - Installation and Connections .....</b>	<b>15</b>
Getting Started .....	15
Packing List .....	15
Unpacking and Inspection.....	15
System Requirements.....	15
Installation of the GXCAL Driver .....	16
Overview of the GXCAL Software .....	17
Configuring Your PXI System using the PXI/PCI Explorer.....	18
Board Installation.....	19
Before you Begin .....	19
Electric Static Discharge (ESD) Precautions .....	19
Installing a Board.....	19

Plug & Play Driver Installation.....	20
Removing a Board .....	21
Connectors .....	22
Installation Directories.....	24
GXCAL Driver Files Description .....	24
Example Programs .....	25
Setup Maintenance Program.....	26
<b>Chapter 4 - Programming the Board.....</b>	<b>27</b>
The GX1034 Driver.....	27
Programming Using C/C++ Tools.....	27
Programming Using Visual Basic.....	27
Programming Using Pascal/Delphi.....	28
Programming GXCAL Boards Using ATEasy®.....	28
Using the GXCAL driver functions .....	29
Initialization, HW Slot Numbers and VISA Resource.....	29
Board Handle .....	30
Reset.....	30
Error Handling .....	30
Driver Version.....	30
Panel.....	31
Distributing the Driver.....	31
Sample Programs .....	32
Sample Program Listing .....	32
<b>Chapter 5 - Functions Reference.....</b>	<b>35</b>
Introduction.....	35
GxCAl Functions .....	36
GxCAlGetBoardSummary .....	37
GxCAlGetBoardTemperature .....	38
GxCAlGetRefClockToPxiRefClockState .....	39
GxCAlGetDriverSummary .....	40
GxCAlGetFunction .....	41
GxCAlGetFunctionStoredMeasuredValue .....	44
GxCAlInitialize.....	46
GxCAlInitializeVisa .....	47
GxCAlPanel .....	48
GxCAlMesure.....	49
GxCAlReset.....	50

GxCalSetFunction.....	51
GxCalSetRefClockToPxiRefClockState.....	54
GxCalGetErrorString.....	55
General Parameter Errors.....	55
Parameter Errors.....	55
In-System Calibration Errors.....	56
Built In Test error/warning.....	56
<b>Index.....</b>	<b>57</b>



# Chapter 1 - Introduction

## Manual Scope and Organization

---

### Manual Scope

This manual provides all the information necessary for installation, operation, and maintenance for the GX1034 Standards Module. This manual assumes the reader has a general knowledge of PC based computers, Windows operating systems, and a general knowledge of modular test equipment.

This manual also provides programming information for the GX1034 driver (referred to in this manual as **GXCAL**). Therefore, this manual assumes the user has a thorough understanding of Windows application development tools and languages.

### Manual Organization

The GX1034 manual is organized in the following manner:

Chapter	Content
Chapter 1 – Introduction	Introduces the GX1034 manual and shows warning conventions used in the manual.
Chapter 2 – Overview	Provides the GX1034 list of features, description of the board, architecture, specifications and the virtual panel description and operation.
Chapter 3 – Installation and Connections	Provides instructions on how to install a GX1034 board and the GXCAL software.
Chapter 4 – Programming the Board	Provides a list of the GXCAL software driver files, general purpose and generic driver functions, and programming methods. Discusses supported application development tools and programming examples.
Chapter 5 – In-System Calibration	Describes the procedure used to perform the in-system calibration.
Chapter 6 – Functions Reference	Provides a list of the GXCAL driver functions. Each function description provides syntax, parameters, and any special programming comments.

## 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.

<b>Formatting Convention</b>	<b>Meaning</b>
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.
Italic type	Specialized terms. Titles of other references and information sources. Placeholders for items you must supply, such as function parameters

## Chapter 2 - Overview

### Introduction

---

The GX1034 is a 3U PXI board that offers PXI system designers the capability to develop a system re-certification strategy, employing only internal system resources. By incorporating the GX1034 as part of a system configuration, it is possible to develop a system accuracy verification strategy that can recertify a system's source and measure base band instrumentation – resulting in simplified support and maintenance logistics and improved system availability. The GX1034's standards exhibit excellent long-term stability with absolute accuracy achieved by employing an on-board EEROM, which contains NIST traceable calibration values for source and resistor standards. The module also includes source and measure resources which can be used to support system self-test functions including continuity verification and verification of instrument functionality. The GX1034's standards accuracy is derived by measuring the module's standards values at the time of manufacture and subsequently saving those values into the module's EEROM, which can then be accessed by the user when re-certifying the test system's instrumentation.

The GX1034 provides a positive DC voltage source reference, a negative DC voltage source reference, an AC voltage source reference, eight 2-wire low drift resistor references, three 4-wire low drift resistor references and a precision 10 MHz frequency reference. The board also has an on-board programmable current source and a 16-bit ADC for voltage measurements.

An on-board temperature sensor is used to record the temperature of the module at the time of calibration time. This value is saved in the on-board non-volatile memory. The GXCAL driver uses the recorded temperature to correct for temperature variations that affect the module's on-board standards. The card also includes built-in-test functionality using its own internal source / measure functions. An external loop back connector available from Geotest is required to fully support the BIT functionality.

### Features

---

The GX1034 Standards Module offers the following features:

- 3U, single slot, PXI board
- On-board standards exhibit excellent long-term stability with traceable absolute accuracy
- On-board EEROM that contains NIST traceable calibration values for all source and resistor standards.
- On-board temperature sensor allows the GXCAL driver to correct the module's standards values for temperature variation and drift.
- The positive DC source supplies an exceptionally stable +9.0V and includes a programmable decade resistive divider network for attenuation of the output by factors of 10, 100 and 1000.
- The negative DC source supplies an exceptionally stable -9.0V and includes a programmable decade resistive divider network for attenuation of the output by factors of 10, 100 and 1000.
- The AC source supplies a nominal 3.5 VAC (RMS) and includes a programmable decade resistive divider network for attenuation of the output by factors 10, 100 and 1000. In addition, the user can program the AC frequency to any of the following values: 100Hz, 1KHz, 10KHz, 100KHz, 50Hz, 60Hz and 400Hz.
- Three on board, high stability, 4-wire 1 $\Omega$ , 10 $\Omega$  and 100 $\Omega$  resistors.
- Eight on board, high stability, 2-wire 1 $\Omega$ , 10 $\Omega$  and 100 $\Omega$ , 1K $\Omega$ , 10K $\Omega$ , 100K $\Omega$ , 1M $\Omega$  and 10M $\Omega$  resistors.
- The 10 MHz frequency output reference employs a calibrated high stability oven controlled crystal oscillator. In addition, a 24-bit divider can be used to divide the high precision 10 MHz frequency reference clock.

- Capability to measure differential voltages on the 9-pin connector using a 16-bit ADC with voltage range of  $-10\text{V}$  to  $+10\text{V}$ .
- On board programmable current source,  $0-19.9\text{mA}$  with 16-bits resolution.
- Capability to measure resistance using the combination of the on-board current source and the capability to measure differential voltages of  $-10\text{V}$  to  $+10\text{V}$  on the 9-pin connector.
- When plugged into one of the non-star controller slots (slots 3 and higher), the GX1034 can be source the PXI 10 MHz backplane clock.
- GXCAL software driver supports both the Virtual Instrument Software Architecture (VISA) standard and Geotest's HW Instrument driver.

## **Applications**

---

- Standards resource for cPCI or PXI Automatic Test Equipment Systems
- Re-certification of PXI baseband instrumentation
- System self-test resource
- 10 MHz frequency standard for PXI applications

## Board Description

Figure 2-1 shows the GX1034 with its front panel connectors. P2 a 25 pin, D type and P1 a BNC 10MHz output TTL clock. Two additional cPCI connectors (J1 and J2) are used to connect to the back plane and to pass in the PCI and PXI signals to the board.

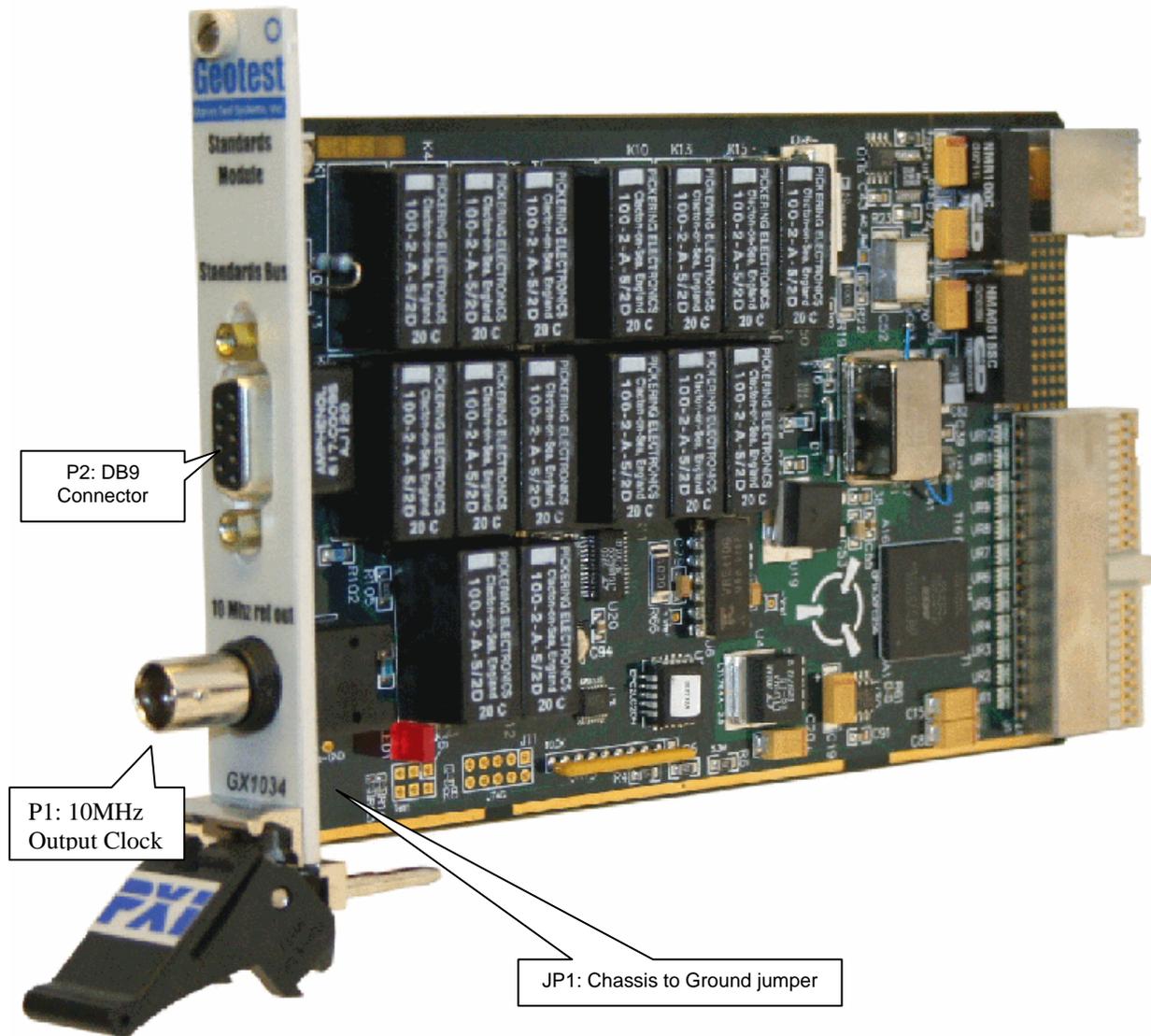


Figure 2-1: GX1034 Board Side View

The GX1034 has an internal 4-wire bus that interfaces to the front through a DB9 connector, providing Hi Sense, Lo Sense, Hi Measure and Lo Measure connections for all on-board standards. In addition, the board has a BNC connector for the 10MHz output clock. The GX1034's standards exhibit excellent long-term stability with absolute accuracy achieved by employing an on-board EEROM, which contains NIST traceable calibration values for source and resistor standards. The module also includes source and measure resources which can be used to support system self-test functions including continuity verification and verification of instrument functionality. Absolute accuracy of the standards is achieved by measuring all standard's values at the time of manufacture and subsequently saving these values into the module's EEROM, which can then be accessed by the user when certifying system instrumentation, achieving traceability of the measurement. Features and capabilities of the board are described below.

#### 4-wire Resistors

The 4-wire resistor references include 1 $\Omega$ , 10 $\Omega$  and 100 $\Omega$  resistors values. The user can select under program control which of those resistors values are connected to the 4-wire output bus.

#### 2-wire Resistors

The 2-wire resistor references include 1 $\Omega$ , 10 $\Omega$ , 100 $\Omega$ , 1 $\Omega$ K, 10 $\Omega$ K, 100 $\Omega$ K, 1 $\Omega$ M and 10 $\Omega$ M resistor values. The user can program which of those resistors values are connected at any given time. Connections will be made to the Hi Measure and Lo Measure busses.

#### Positive and Negative DC Voltage Sources

The DC source supplies an exceptionally stable +9.0V and -9.0V source and includes a programmable decade resistive divider network for attenuation of the output by factors of 10, 100 and 1000. The DC source output impedance will vary depending on the selected voltage value. The DC source values stored in the on-board EEROM reflect output voltages measured into a 10 GOhm load with an 8.5 digit DMM. The output voltage for each value is available on the Hi Measure and Lo Measure output busses. Approximate output impedance for each attenuator range is presented in the following table. Depending on the application, it may be necessary to compensate for the effects of the voltage source's output impedance.

Voltage Range	Approximate Output Impedance
x 1	990 ohms
x .1	900 ohms
x .01	100 ohms
x .001	10 ohms

AC and DC Source Output Impedance

#### AC Voltage Source

The AC source supplies a nominal 3.5 VAC (RMS) and employs the decade resistive divider network for attenuation of the output by factors of 10, 100 and 1000. In addition, for each programmable attenuated AC voltage range the user can program the AC frequency to any of the following values: 100Hz, 1KHz, 10KHz, 100KHz, 50Hz, 60Hz and 400Hz. AC source output impedances are similar to those shown in the above table.

#### 10 MHz Frequency Output Reference

The 10 MHz frequency output reference employs a calibrated, high stability oven controlled crystal oscillator. In addition, a 24-bit divider can be programmed to divide this high precision 10 MHz frequency reference clock. The 10 MHz high stability oven controlled crystal oscillator when plugged into the non-star controller slot, can be programmed to source the standard 10MHz PXI backplane clock.

#### Output Current Source

The card has an on board programmable current source. Output current can be programmed to any value between 0-19.9mA with 16-bit resolution. The current source's voltage compliance range is 0 to +12.5 volts.

#### Voltage measurement (ADC)

The card also has the capability to measure differential voltages on the 9-pin connector using a 16-bit ADC with voltage range of -10V to +10V.

### **Resistance measurement**

The card also has the capability to measure resistance using the combination of the on-board current source and the capability to measure differential voltages of  $-10\text{V}$  to  $+10\text{V}$  on the 9-pin connector.

### **Temperature sensor**

The board has an on-board temperature sensor that is been used to record the temperature at calibration time. This value is then saved on the on-board non-volatile memory. The driver then uses the recorded temperature to correct for temperature variations.

### **Built-in-test (BIT)**

Using its internal components, the card can execute a built-in-test (BIT) by connecting a special connector available from by Geotest.

## Architecture

### Block Diagram

The block diagram below in Figure 2-2: GX1034 Block Diagram illustrates the GX1034 architecture. The board communicates with the host computer using the PCI interface. The software driver and application software configures board resources and makes connections to the 4-wire bus. The board has a 9 pin sub-D connector and a BNC connector on the front panel.

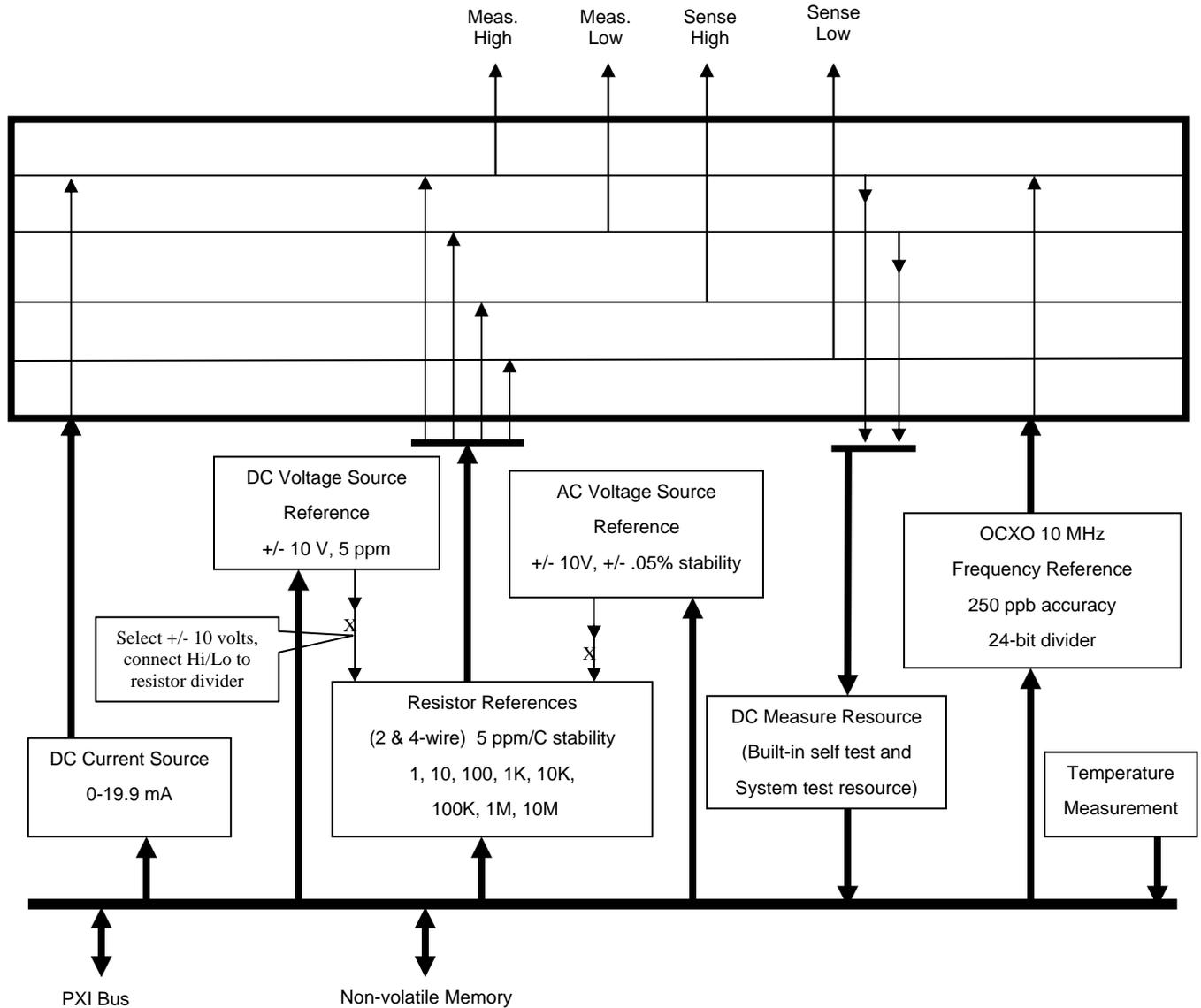


Figure 2-2: GX1034 Block Diagram

## Specifications

The following table outlines the specifications of the GX1034. All accuracies are based on values measured with a NIST traceable 8.5 digit DMM and stored in the GX1034 card's EEPROM.

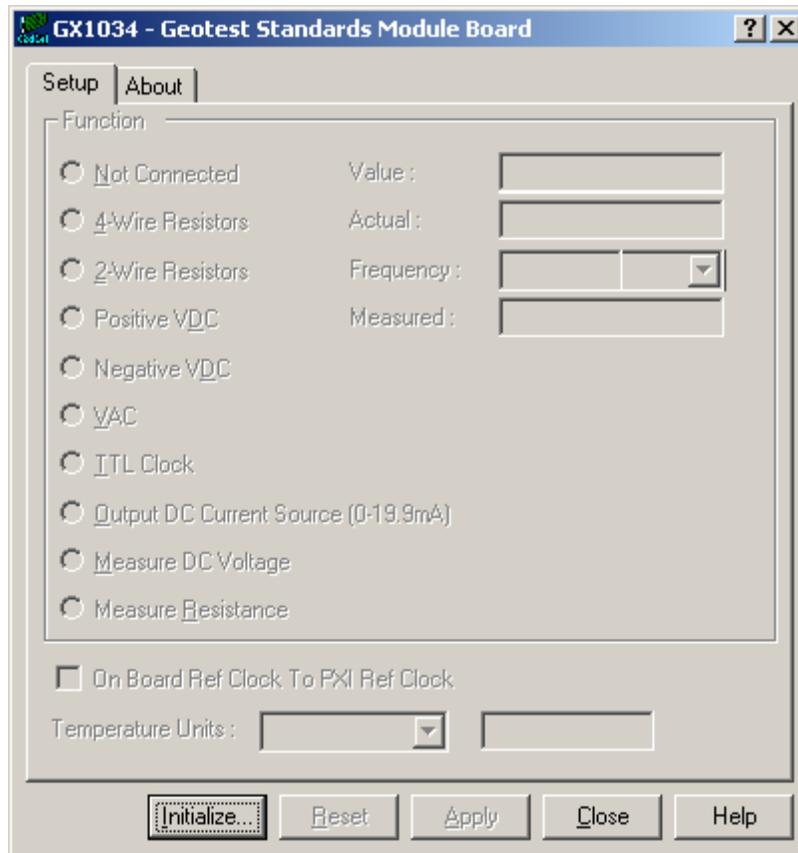
<b>4-wire Resistance</b>	
Value	1 $\Omega$ , nominal
Accuracy	+/- 0.005%
Stability	3 ppm / $^{\circ}\text{C}$
Value	10 $\Omega$ , nominal
Accuracy	+/- 0.005%,
Stability	0.2 ppm / $^{\circ}\text{C}$
Values	100 $\Omega$ , nominal
Accuracy	+/- 0.005%
Stability	0.2 ppm / $^{\circ}\text{C}$
<b>2-wire Resistance</b>	
Value:	1K $\Omega$ , nominal
Accuracy	+/- 0.002%
Stability	0.2 ppm / $^{\circ}\text{C}$
Value:	10K $\Omega$ , nominal
Accuracy	+/- 0.002%
Stability	0.2 ppm / $^{\circ}\text{C}$
Value:	100K $\Omega$ , nominal
Accuracy	+/- 0.004%
Stability	5 ppm / $^{\circ}\text{C}$
Value	1M $\Omega$ , nominal
Accuracy	+/- 0.008%
Stability	3 ppm / $^{\circ}\text{C}$
Value	10M $\Omega$ , nominal
Accuracy	+/- 0.03%
Stability	2 ppm / $^{\circ}\text{C}$
<b>DC Voltage Source</b>	
Values:	+/- 9.0V, +0.9V, +0.09V, +0.009V, nominal
Accuracy	+/- .002%, Using EEPROM values
Temperature Coefficient	0.6 ppm / $^{\circ}\text{C}$
Long Term Drift	6 ppm for 1000 operating hours

<b>AC Voltages (RMS)</b>	
Values:	3.5VAC, 0.35VAC, 0.035VAC, 0.0035VAC, nominal 50 Hz, 60 Hz, 100 Hz frequencies
Accuracy	+/- .08%
Values:	3.5VAC, 0.35VAC, 0.035VAC, 0.0035VAC, nominal 1 KHz, 10 KHz, frequencies
Accuracy	+/- .05%
Values:	3.5VAC, 0.35VAC, 0.035VAC, 0.0035VAC, nominal 100 KHz frequency
Accuracy	+/- 0.4%
Temperature Coefficient	3 ppm / °C, (voltage reference for D/A)
Long Term Drift	6 ppm for 1000 operating hours
Frequency Accuracy	+/- 0.1 ppm
<b>10 MHz Frequency Output Reference</b>	
Value	10 MHz crystal oscillator
Initial accuracy	± 0.1 ppm
Drift	100 ppb / °C , max
<b>Output Current Source</b>	
Range	0-19.9mA, 16-bit resolution
Accuracy	+/- .05% of programmed value
Compliance	0 – 12.5 volts
<b>Voltage Measurement (ADC)</b>	
Resolution	16-bit
Mode	Differential
Input impedance	20 KΩ, nominal
Range	-10V to +10V
Accuracy	+/- 1% of reading
<b>Power Requirements</b>	
3.3V	500 mA
5V	1 A
12 V	500 mA
<b>Physical</b>	
Size	3U, single slot
App. Weight:	210 grams
Operating Temp	0 to 50°C
Humidity	10% - 80% non-condensing

## Virtual Panel Description

The **GXCAL** includes a virtual panel program, which provides full access to the various configuration settings and operating 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 **GX1034 Panel** from the **Geotest, GXCAL** menu under the **Start** menu. The GX1034 virtual panel opens as shown here:



**Figure 2-3: GX1034 Virtual Panel (not Initialized)**

The function of the panel button controls are shown below:

**Initialize** Opens the Initialize Dialog (see Initialize Dialog paragraph) in order to initialize the board driver. The current settings of the selected counter **will not change after calling initialize**. The panel will reflect the current settings of the counter after the Initialize dialog closes.

**Reset** - resets the PXI board settings to their default state and clears the reading.

**Apply** – applies changed settings to the board

**Close** - closes the panel. Closing the panel **does not affect** the board settings.

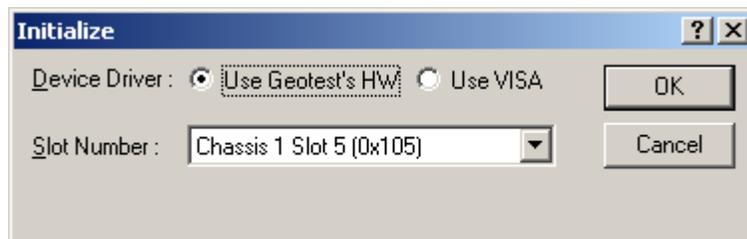
**Help** - opens the on-line help window. In addition to the help menu, the caption shows a **What's This Help** button (?) button. This button can be used to obtain help on any control that is displayed in the panel window. To display the What's This Help information click on the (?) button and then click on the control – a small window will display the information regarding this control.

## Virtual Panel Initialize Dialog

The Initialize dialog initializes the driver for the selected counter board. The counter settings **will not change** after initialize is called. Once initialize, the panel will reflect the current settings of the counter.

The Initialize dialog supports two different device drivers that can be used to access and control the board:

1. **Use Geotest's HW** – this is the device driver installed by the setup program and is the default driver. When selected, the **Slot Number** list displays the available counter boards installed in the system and their slots. The chassis, slots, devices and their resources are also displayed by the HW resource manager, **PXI/PCI Explorer** applet that can be opened from the Windows Control Panel. The **PXI/PCI Explorer** can be used to configure the system chassis, controllers, slots and devices. The configuration is saved to PXISYS.INI and PXIeSYS.INI located in the Windows folder. These configuration files are also used by VISA. The following figure shows the slot number 0x105 (chassis 1 Slot 5). This is the slot number argument (*nSlot*) passed by the panel when calling the driver **GxCalInitialize** function used to initialize driver with the specified board.  
**Use VISA** – this is a third party device driver usually provided by National Instrument (NI-VISA). When



**Figure 2-4: Initialize Dialog Box using Geotest's HW driver**

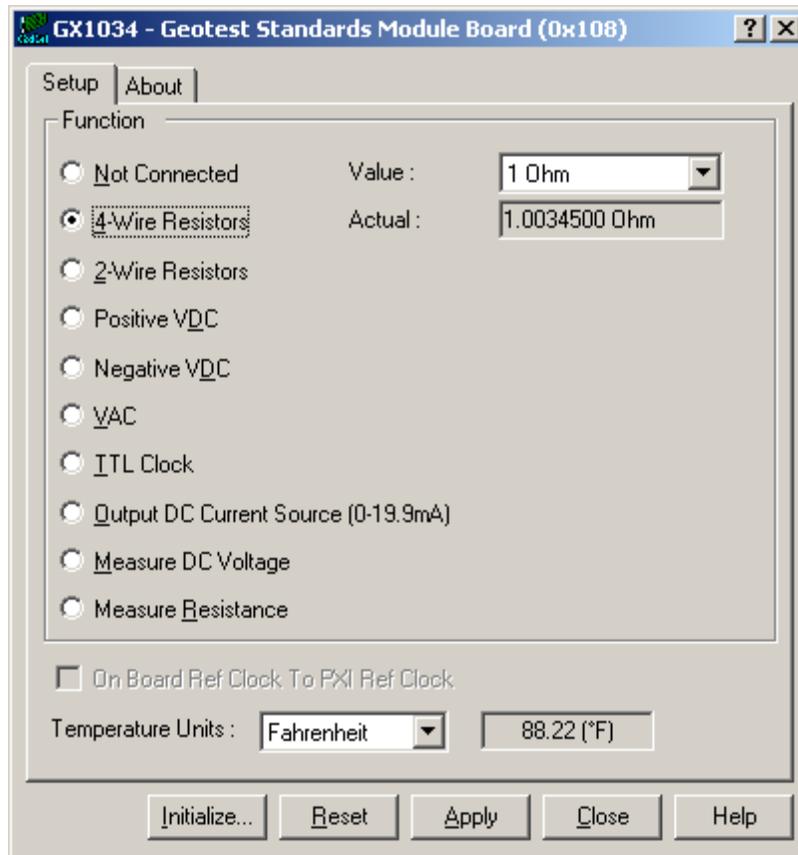
selected, the **Resource** list displays the available boards installed in the system and their VISA resource address. The chassis, slots, devices and their resources are also displayed by the VISA resource manager, **Measurement & Automation (NI-MAX)** and in Geotest **PXI/PCI Explorer**. The following figure shows PXI9::13::INSTR as the VISA resource (PCI bus 9 and Device 13). This is VISA resource string argument (*szVisaResource*) passed by the panel when calling the driver **GxCalInitializeVisa** function to initialize the driver with the specified board.



**Figure 2-5: Initialize Dialog Box using VISA resources**

## 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 2-6: GX1034 Virtual Panel (Initialized)**

The following controls are shown in the Setup page:

**Function Settings:** Sets/displays the connected standards (connections and settings will only take place after pressing the Apply button).

**Not Connected:** All standards disconnected from the output.

**4-Wire Resistors:** when selected, allows connection to one of the 4-wire 1 $\Omega$ , 10 $\Omega$  and 100 $\Omega$  resistors values. Actual value displayed is the measured resistor value, as was recorded at calibration time.

**2-Wire Resistors:** when selected, allows connection to one of the 2-wire 1 $\Omega$ , 10 $\Omega$  and 100 $\Omega$ , 1 $\Omega$ K, 10 $\Omega$ K, 100 $\Omega$ K, 1 $\Omega$ M and 10 $\Omega$ M resistor values. Actual value displayed is the measured resistor value, as was recorded at calibration time.

**Positive VDC:** when selected allows connection to one of the DC voltage +9.0V, +0.9V, +0.09V and +0.009V values. Actual value displayed is the value that was recorded and stored in the EEROM at calibration time.

**Negative VDC:** when selected allows connection to one of the DC voltage -9.0V, -0.9V, -0.09V and -0.009V values. Actual value displayed is the value that was recorded and stored in the EEROM at calibration time..

**AC:** when selected allows connection to one of the AC voltage +3.5VAC, 0.35VAC, 0.035VAC and 0.0035VAC values. Actual value displayed is the value recorded and stored in the EEROM at calibration time. It also enables selection of the 100Hz, 1KHz, 10KHz, 100KHz, 50Hz, 60Hz or 400Hz frequency.

**TTL Clock:** when selected displays/sets the Clock Divider value and the actual divided output frequency (the function will not be connected until pressing the Apply button).

**Measure Voltage:** when checked disconnects any currently connected standard and the **Output Current Source** if currently checked. Allows voltage measurements of on the output connector.

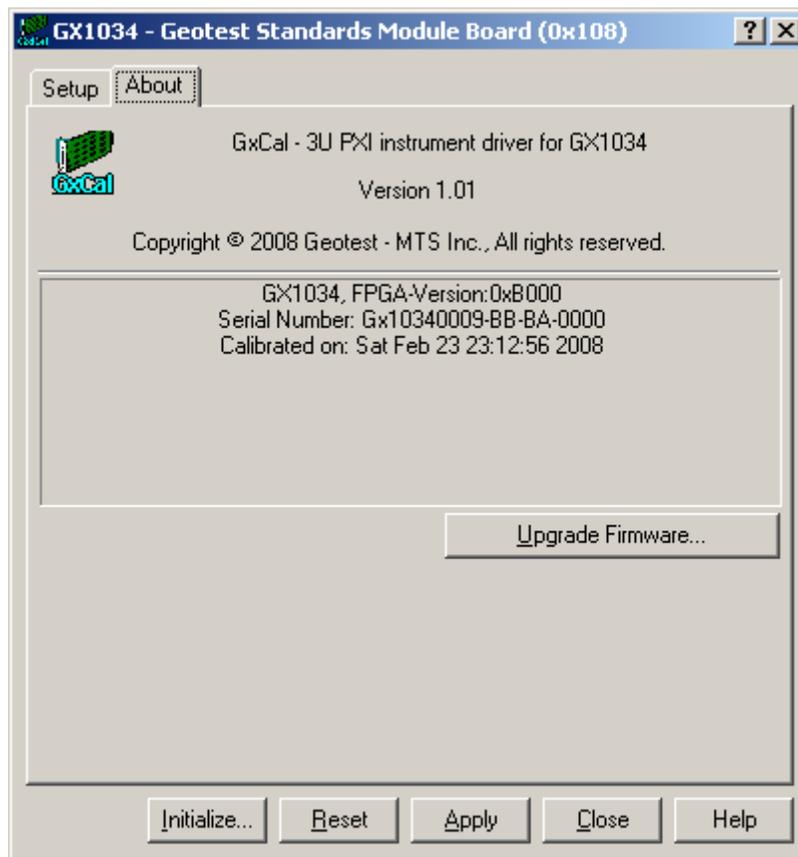
**Output Current Source (0-19.9mA):** when checked disconnects any currently connected standards and the **Measure Voltage** if currently checked. Allows user to set the Output Current Source form 0 to 19.9mA

**Temperature Units:** Sets/displays temperature units for Celsius or Fahrenheit.

**On Board Ref Clock To PXI Ref Clock:** When connected the internal 10MHz clock will drive the PXI backplane clock.

### Virtual Panel About Page

Clicking on the **About** tab will show the **About** page as shown in Figure 2-7:



**Figure 2-7: GX1034 Virtual Panel – About Page**

The following controls are shown in the About page:

The top part of the **About** page displays version and copyright of the GXCAL driver. The bottom part displays the board summary.

The **About** page also contains a button **Upgrade Firmware...** used to upgrade the board FPGA. This button may be 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 GX1034 board and installation instructions for the GX1034 (GXCAL) 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
GX1034 Driver Installation	This Chapter
Programming	Chapter 4
GX1034 Function Reference	Chapter 5

### Packing List

All GX1034 boards have the same basic packing list, which includes:

1. GX1034 Board
2. GX1034 Driver Disk

### 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 GX1034 instrument boards are designed for use with a 3U or 6U cPCI or PXI compatible chassis. The software is compatible with any computer system running Windows 98, Windows Me, Windows 2000, Windows XP, Windows VISTA (32 bit) 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 GXCAL Driver

---

Before installing the board it is recommended to install the GXCAL driver as described in this section. To install the GXCAL driver follow the instruction described here:

1. Insert the Geotest CD-ROM and locate the **GXCAL.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](http://www.geotestinc.com)).
2. Run the GXCAL setup and follow the instruction on the Setup screen to install the GXCAL driver.

---

**Note:** When installing under Windows NT/2000/XP/VISTA, you may be required to restart the setup after logging-in as a user with an Administrator privileges. This is required in-order to upgrade your system with newer Windows components and to install kernel-mode device drivers (HW.SYS and HWDEVICE.SYS) required by the GXCAL 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 GXCAL 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\GXCAL.
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, GXCAL menu located in the Windows Taskbar.

## Overview of the GXCAL Software

---

Once the GxChassis 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).

- **GXCAL Panel** – use to configure the smart chassis features includes over-temperature behavior, control the system fans, measure slot temperature and system power supply usage and program trigger lines direction and connection between PXI bus segments.
- **GXCAL driver** - a DLL (GXCAL.DLL located in the Windows System folder) used to program and control the board.
- **Programming files and examples** – interface files and libraries for various programming tools, see later in this chapter for a complete list of files 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 **GxCAlInitialize** and **GxCAlInitializeVisa** 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).

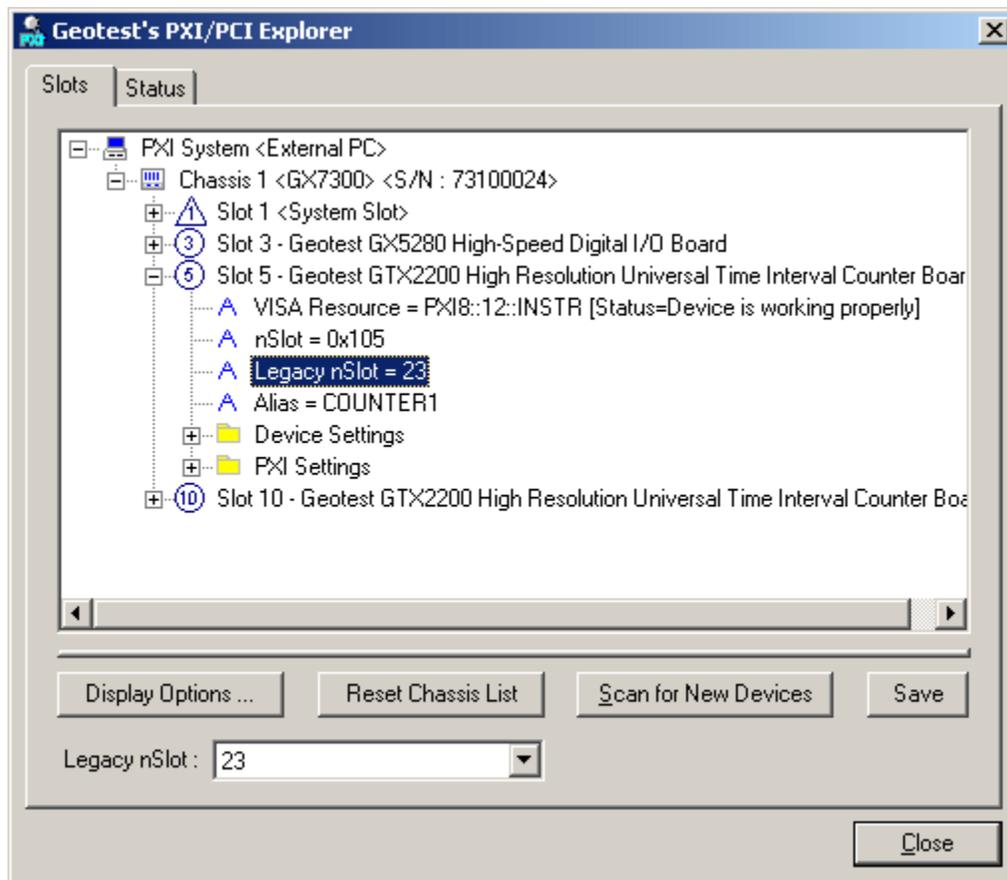


Figure 3-1: PXI/PCI Explorer

## Board Installation

---

### Before you Begin

- Install the GXCAL 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 GX1034 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.



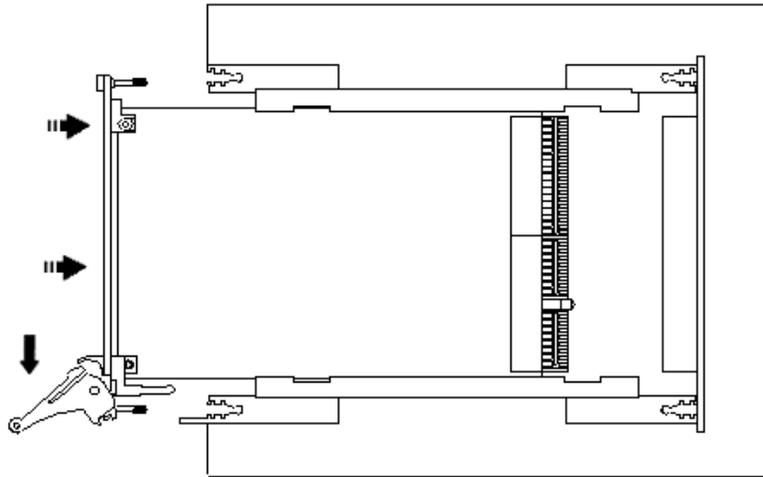
**Caution** - Do not insert or remove any board while the computer is on. Turn off the power from the PXI chassis before installation.

---

### Installing a Board

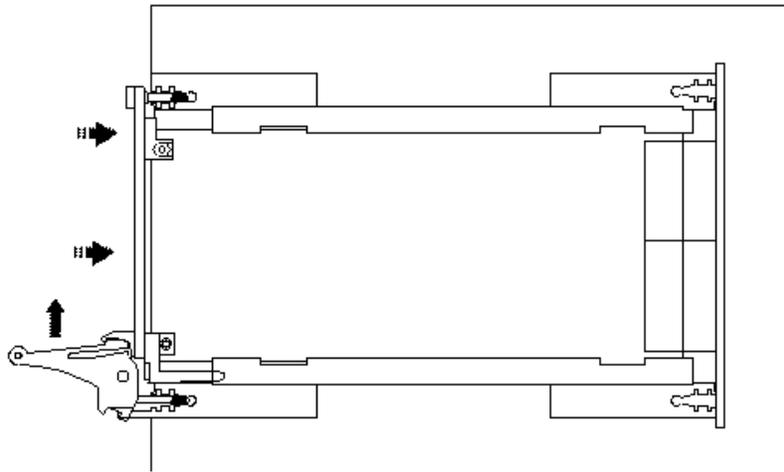
Install the board as follows:

1. Install first the GXCAL Driver as explain in the next section.
2. Turn off the PXI chassis and unplug the power cord.
3. Locate a PXI empty slot on the PXI chassis.
4. Place the module edges into the PXI chassis rails (top and bottom).
5. 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**

6. 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**

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

### Plug & Play Driver Installation

Plug & Play operating systems such as Windows 9x, Me, Windows 2000 or XP (Not Windows NT) notify 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 GXCAL driver was not installed prior to the board installation), you may install the GXCAL driver 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 GXCAL 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 GXCAL driver.

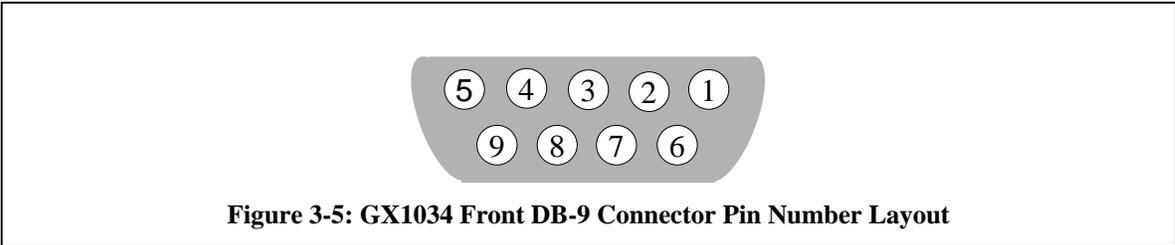
## Connectors

---

Figure 3-4: GX1034 Front Connector shows the available GX1034 board connector with description:



**Figure 3-4: GX1034 Front Connectors**



The following are the connector signals for the GX1034 board:

Pin#	Signal	Description
1	Isolated GND (AGND)	Isolated GND (AGND)
2	Measure Low	AC and DC voltages and resistance reference
3	Sense Low	4-wire connections
4	Not Connected	Not Connected
5	PXI Ground	PXI Ground
6	Measure High	AC and DC voltages and resistance reference
7	Sense High	4-wire connections
8	Not Connected	Not Connected
9	Chk BIT adaptor is Connected	Check BIT adaptor is Connected

**Table 3-1: Output Connectors**

## Installation Directories

---

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

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

Name	Purpose / Contents
...\Geotest\GXCAL	The GX1034 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. GXCAL Driver and example are copied to this directory only if <i>ATEasy</i> is installed to your machine.
...\Windows\System (Windows 9x/Me), or ...\Windows\System32 when running Windows NT/2000/XP	Windows System directory. Contains the GXCAL DLL driver and some upgraded system components, such as the HTML help viewer, etc.

## GXCAL Driver Files Description

---

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

### Driver File and Virtual Panel

- GXCAL.DLL - 32-Bit MS-Windows DLL for applications running under Windows, 95, 98, Me, NT, 2000, XP or above.
- GXCALPANEL.EXE – An instrument front panel program for all GXCAL supported boards.

### Interface Files

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

- GXCAL.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++
- GXCAL.LIB - Import library for GXCAL.DLL (used when linking C/C++ application that uses GXCAL.DLL).
- GXCALBC.LIB - Import library for GXCAL.DLL (used when linking Borland C/C++ application that uses GXCAL.DLL).
- GXCAL.PAS - interface file to support Borland Pascal Borland Delphi.
- GXCAL.BAS - Supports Microsoft Visual Basic 4.0, 5.0 and 6.0.
- GXCAL.VB - Supports Microsoft Visual Basic .NET.
- GX1034.DRV - ATEasy driver File for GX1034.GXCAL Virtual Panel Program.
- GxCAl.llb – LabView library.

### GXCAL On-line Help and Manual

GXCAL.CHM – On-line version of the GX1034 User’s Guide. The help file is provided in a Windows Compiled HTML help file (.CHM). The file contains information about the GX1034 board, programming reference and panel operation.

GX1034.PDF – On line, printable version of the GX1034 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 GXCAL manuals. You can view and/or print this file using the Windows NOTEPAD.EXE or 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:

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

#### Microsoft Visual C++ 6.0 example files:

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

#### Borland C++ example files:

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

**Microsoft Visual Basic .NET example files:**

- GxCalExampleVB.vb - Example form.
- GxCalExampleVB.resx - Example form resource.
- GxCalExampleVBapp.config - Example application configuration file.
- GxCalExampleVBAssebleyInfo.vb - Example application assembly file
- GxCalExampleVB.vbproj - Project file
- GxCalExampleVB.exe - Example executable

**Microsoft Visual Basic 6.0 example files:**

- GxCalExampleVB6.frm - Example form
- GxCalExampleVB6.frx - Example form binary file
- GxCalExampleVB6.vbp - Project file
- GxCalExampleVB6.exe - Example executable.

**ATEasy driver and examples files (ATEasy Drivers directory):**

- Gx1034.prj - example project
- Gx1034.sys - example system
- Gx1034.prg - example program

**LabView example:**

- GxCalExample.vi

## **Setup Maintenance Program**

---

You can run the Setup again after GXCAL 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 show below allows you to modify the current GXCAL installation. The following options are available in Maintenance mode:

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

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 GXCAL driver. The GXCAL 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 driver supports many development tools. Using these tools with the driver is described in this chapter. In addition, the GX1034 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++ 6.0 is included at the end of this chapter. Since the driver functions and parameters are identical for all operating systems and development tools, the example can serve as an outline for other programming languages, programming tools, and other GX1034 driver types.

### The GX1034 Driver

---

The GX1034 driver is a 32 bit Windows DLL file: GXCAL.DLL. The DLL is used with 32 bit applications running under Windows 95/98/ME and Windows NT/2000/XP. The DLL uses a device driver to access the board resources. The device driver HW.SYS (on Windows NT/2000/XP) or HW.VXD (on Windows 9x/Me) is installed by the setup program and is shared by other Geotest products (ATEasy, GXDIO, etc).

The 32-bit 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 GX1034 driver with C/C++ development tools:

- Include the GXCAL.H header file in the C/C++ source file that uses the GX1034 function. This header file is used for all driver types. 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 import library GXCAL.LIB for Microsoft Visual C++ and GXCALBC.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 GX1034 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 GXCAL.BAS to the project. For Visual Basic .NET use the GXCAL.VB.

The file can be loaded using *Add File* from the Visual Basic *File menu*. The GXCAL.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 GXCAL.PAS to the project. The GXCAL.PAS file contains a **unit** with function prototypes for the DLL functions. Include the GX1034 unit in the **uses** statement before making calls to the GX1034 functions.

## Programming GXCAL Boards Using ATEasy®

---

The GX1034 package is supplied with a separate ATEasy driver for each of board types. The ATEasy driver uses the GXCAL.DLL to program the board. In addition, each driver is supplied with an example that contains a program and a system file pre-configured with the ATEasy driver. Use the driver shortcut property page from the System Drivers sub-module to change the PCI slot number before attempting to run the example.

Using commands declared in the ATEasy driver are easier to use than using the DLL functions directly. The driver commands will also generate exception that allows the ATEasy application to trap errors without checking the status code returned by the DLL function 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 *nHandle* parameter is omitted. The driver handles this parameter automatically. ATEasy uses driver logical names instead i.e. PRES1, PRES2 for GX1034.
- The *nStatus* parameter was omitted. Use the Get Status commands instead of checking the status. After calling a DLL function the ATEasy driver will check the returned status and will call the error statement (in case of an error status) to generate exception that can be easily trapped by the application using the **OnError** module event or using the **try-catch** statement.

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 commands are self-documented. Their syntax is similar to English. In addition, you may generate the commands from the code editor context menu or by using the ATEasy's code completion feature instead of typing them directly.

## Using the GXCAL driver functions

The GXCAL driver contains a set of functions for the GX1034. Functions names that starts with the **GxCal** prefix applies to all GXCAL boards (i.e. **GxCalGetDriverSummary**). The GXCAL functions are designed with consistent set of arguments and functionality. All boards have a function that initializes the GXCAL 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 GXCAL driver supports two device drivers HW and VISA which are used to initialize, identify and control the board. The user can use the **GxCalInitialize** to initialize the board 's driver using HW and **GxCalInitializeVisa** 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 GXCAL driver. To initialize and control the board using the HW use the **GxCalInitialize**(*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.

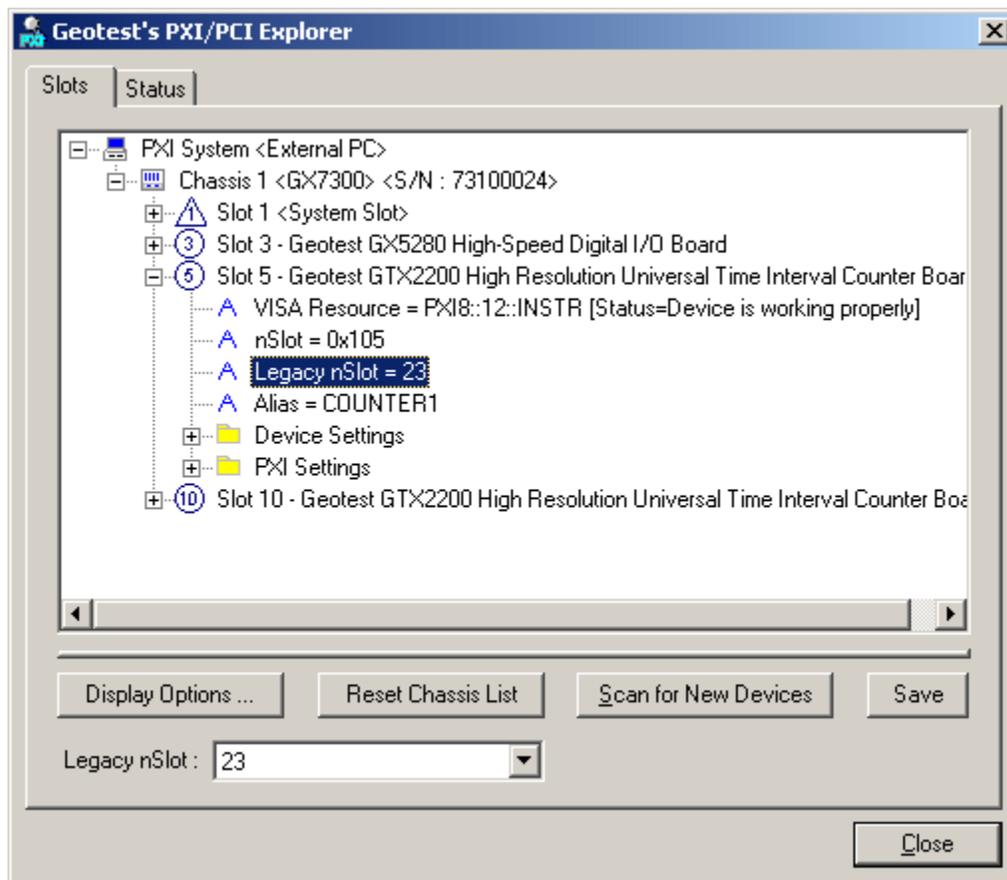


Figure 4-1: PXI/PCI Explorer

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). GXCAL setup installs a VISA compatible driver for the GXCAL board in-order to be recognized by the VISA provider. Use the GXCAL function **GxCalInitializeVisa** (*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 **GxCalInitialize** 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 **GxCalInitialize** or the **GxCalInitializeVisa** functions is a short integer (16 bits) number. It is used by the GXCAL 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 **GxCalInitialize** function. But there is no need to destroy the handle. Calling **GxCalInitialize** 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 **GxCalReset**(*nHandle, nStatus*) function. See the Function Reference for more information regarding the specific board.

### Error Handling

All GXCAL 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 **GxCalGetErrorString** function to return a string representing the error. The **GxCalGetErrorString** reference contains possible error numbers and their associated error strings.

### Driver Version

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

## Panel

Calling the **GxCAlPanel** 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 **GxCAlPanel** function is also used by the GXCALPANEL.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 (GXCAL.DLL and the HW device driver files located in the HW folder) can be shipped with the application. Typically, the GXCAL.DLL 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 GXCAL driver files. Alternatively, you can provide the GXCAL.EXE setup program to be installed along with the board. The setup program can be invoked in silent mode (with /s) to install the driver using the command line specified settings with no user interface. Geotest provides permission to re-distribute the driver software providing it is in support of an application using the target hardware.

## Sample Programs

---

The following example demonstrates how to program the board using the C programming language under Windows. The example shows how to initialize the counter, set it up for measurement or trigger settings and get the reading.

To run, enter the following command line:

```
GxCALExample <Slot> <function> <value >
```

Where:

<Slot>	PXI Explorer slot number where the board reside.
< function >	Set Function: <ul style="list-style-type: none"> <li>• GXCAL_FUNCTION_NOT_CONNECTED=Disconnect all Disconnect all functions</li> <li>• GXCAL_FUNCTION_4WIRE=Connect 4-wire</li> <li>• GXCAL_FUNCTION_2WIRE=Connect 2-wire</li> <li>• GXCAL_FUNCTION_POSITIVE_VDC=Connect positive voltage</li> <li>• GXCAL_FUNCTION_NEGATIVE_VDC=Connect negative voltage</li> </ul>
<value >	Set Function value

## Sample Program Listing

---

```

/*****
FILE      : GxCALExampleC.cpp

PURPOSE   : WIN32 sample program for GX2470 board using the GxCAL driver.

CREATED   : Feb. 2008
COPYRIGHT : Copyright 2008 GEOTEST - MTS Inc.
COMMENTS  :

```

To compile the WIN32 example:

1. Microsoft VC++
  - Load GxCALExampleC.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 GxCALExampleC.bpr from the Project/Open Project... menu
  - Select Project/Build all from the menu

```

/*****/
#include "windows.h"
#include "GxCAL.h"
#include "stdio.h"

// Borland C++ Builder compat. block
#ifdef __BORLANDC__
#pragma hdrstop
#include <condefs.h>
USELIB("GxCALBc.lib");
USERC("GxCALExampleC.rc");
//-----
#endif // defined(__BORLANDC__)

```

```

//*****
//          DisplayMsg
//*****
void DisplayMsg(PSTR lpszMsg)
{
    MessageBeep(0);
    MessageBox(0, lpszMsg, "GxCAL", MB_OK);
    return;
}

//*****
//          DisplayUsage
//*****
void DisplayUsage(void)
{
    DisplayMsg(
        "\r\nThis example shows how to use the GX1034:\r\n\r\n"
        "Usage:\r\n"
        "GxCALExampleC <slot_number> <function> <value> \r\n"
        "Where : \r\n"
        "<slot_number> PXI slot number as shown by the PXI explorer\r\n"
        "<function> Set Function :\r\n"
        "\tGXCAL_FUNCTION_NOT_CONNECTED=Disconnect all Disconnect all
functions\r\n"
        "\tGXCAL_FUNCTION_4WIRE=Connect 4-wire\r\n"
        "\tGXCAL_FUNCTION_2WIRE=Connect 2-wire\r\n"
        "\tGXCAL_FUNCTION_POSITIVE_VDC=Connect positive voltage\r\n"
        "\tGXCAL_FUNCTION_NEGATIVE_VDC=Connect negative voltage\r\n"
        "<value> Set Function value\r\n\r\n"

        "NOTE: To change command line under Windows right click on the example
shortcut\r\n"
        "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;
    GxCALGetErrorString(nStatus, sz, sizeof sz, &nStatus);
    DisplayMsg(sz);
    DisplayMsg("Aborting the program...");
    exit(nStatus);
}

//*****
//          MAIN
//
// This main function receives five parameters
//

```

```

// Assigned board slot number (e.g. 1) always need to be passed.
// <function> Set Function :
// GXCAL_FUNCTION_NOT_CONNECTED=Disconnect all Disconnect all functions
// GXCAL_FUNCTION_4WIRE=Connect 4-wire
// GXCAL_FUNCTION_2WIRE=Connect 2-wire
// GXCAL_FUNCTION_POSITIVE_VDC=Connect positive voltage
// GXCAL_FUNCTION_NEGATIVE_VDC=Connect negative voltage
// <value> Set Function value
//
//*****
int main(int argc, char **argv)
{
    short nSlotNum;          // Board slot number
    short nFunction;        // Board function
    short nHandle;          // Board handle
    short nStatus;          // Returned status
    double    dValue;

    // Check number of arguments received
    if (argc<3) DisplayUsage();

    // Parse command line parameters
    nSlotNum=(SHORT)strtol(++argv, NULL, 0);
    nFunction =(SHORT)strtol(++argv, NULL, 0);
    dValue =strtod(++argv, NULL);

    if (nSlotNum<0)
        DisplayUsage();

    GxCALInitialize(nSlotNum, &nHandle, &nStatus);
    CheckStatus(nStatus);

    switch(nFunction)
    {
        case GXCAL_FUNCTION_NOT_CONNECTED:
        case GXCAL_FUNCTION_4WIRE:
        case GXCAL_FUNCTION_2WIRE:
        case GXCAL_FUNCTION_POSITIVE_VDC:
        case GXCAL_FUNCTION_NEGATIVE_VDC:
            GxCALSetFunction(nHandle, nFunction, dValue, 0, &nStatus);
            CheckStatus(nStatus);
            break;
        default:
            DisplayUsage();
    }
    return 0;
}

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

```

## Chapter 5 - Functions Reference

### Introduction

---

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

All function and parameter syntax follow 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 **GxCalInitialize** 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 non-zero on error. The description of the error is available using the **GxCalGetErrorString** function or by using a predefined constant, defined in the driver interface files: GXCAL.H, GXCAL.BAS, GXCAL.VB, GXCAL.PAS or GX1034.DRV.
- Parameter name are prefixed as follows:

Prefix	Type	Example
a	Array - prefix this before the simple type.	<i>anArray</i> (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**

## GxCal Functions

---

The following list is a summary of functions available for the GX1034:

<b>Driver Functions</b>	<b>Description</b>
<b>General</b>	
GxCalInitialize	Initializes the driver for the specified slot using the HW device driver.
GxCalInitializeVisa	Initializes the driver for the specified slot using VISA.
GxCalPanel	Opens a virtual panel used to interactively control the GX1034.
GxCalReset	Resets the GX1034 board to its default settings.
GxCalGetBoardSummary	Returns the board information.
GxCalGetDriverSummary	Returns the driver name and version.
GxCalGetErrorString	Returns the error string associated with the specified error number.
<b>Functions</b>	
GxCalGetBoardTemperature	Returns the board temperature.
GxCalGetCounterRefClockToPxiRefClockState	Returns the connection state of the Counter Reference Clock to the PXI Reference Clock.
GxCalGetFunction	Returns the connected function and the function settings.
GxCalGetFunctionStoredMeasuredValue	Returns the specified function stored measured values.
GxCalMeasure	Measure the output voltage or resistance.
GxCalSetFunction	Sets the function and the function settings.
GxCalSetCounterRefClockToPxiRefClockState	Sets the connection state of the Counter Reference Clock to the PXI Reference Clock.

## GxCALGetBoardSummary

---

### Purpose

Returns the board information.

### Syntax

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

### Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1034 board.
szSummary	PSTR	Buffer to contain the returned board info (null terminated) string.
nSumMaxLen	SHORT	Size of the buffer to contain the error string.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

### Comments

The counter summary string provides the following data from the board in the order shown:

- Instrument Name (e.g., GX1034)
- Firmware (FPGA) Version (e.g., 0xB000)
- Serial Number (e.g. GX10340005-BB-BA-0000)
- Calibration date and time (e.g. Calibrated on: Sat Feb 23 23:36:14 2008)

For example, the returned string could look like the following:

“GxCAL, FPGA-Version:0xB000, Serial Number: GxCAL0005-BB-BA-0000, Calibrated on: Sat Feb 23 23:36:14 2008”

### Example

The following example returns the board summary:

```
SHORT nHandle, nStatus;
CHAR szSummary [256];

GxCALGetBoardSummary(nHandle, szSummary, sizeof(szSummary), &nStatus);
```

### See Also

**GxCALGetDriverSummary**, **GxCALInitialize**, **GxCALGetErrorString**

## GxCalGetBoardTemperature

---

### Purpose

Returns the board temperature.

### Syntax

**GxCalGetBoardTemperature** (*nHandle, nUnits, pdTemp, pnStatus*)

### Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1034 board.
nUnits	SHORT	Returned temperature unite are: GXCAL_TEMPERATURE_IN_CELSIUS: temperatures are in Celsius. GXCAL_TEMPERATURE_IN_FAHRENHEIT: temperatures are in Fahrenheit.
pdTemp	PDOUBLE	Returned measured board temperature.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

### Comments

Calling this function initiates a new temperature measurement.

### Example

The following example returns the board temperature in Celsius:

```
SHORT nHandle, nStatus;
PDOUBLE dTemp;
```

```
GxCalGetBoardTemperature (nHandle, GXCAL_TEMPERATURE_IN_CELSIUS, &dTemp, &nStatus);
```

### See Also

**GxCalGetDriverSummary, GxCalInitialize, GxCalGetErrorString**

## GxCALGetRefClockToPxiRefClockState

---

### Purpose

Returns the connection state of the on board reference Clock to the PXI reference Clock.

### Syntax

**GxCALGetRefClockToPxiRefClockState** (*nHandle, pnState, pnStatus*)

### Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1034 board.
pnState	PSHORT	Reference Clock to PXI Reference Clock state: 0 GXCAL_REF_CLOCK_TO_PXI_REF_CLOCK_OFF: Disconnect the on board reference Clock from the PXI Reference Clock. 1 GXCAL_REF_CLOCK_TO_PXI_REF_CLOCK_ON: Connect the on board reference Clock to the PXI Reference Clock.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

### Comments

The PXI chassis has a 10 MHz clock line (PXI\_CLK10) in its backplane that allows devices plugged into the non-star controller slots (slots 3 and higher) to phase-lock their oscillators to the backplane clock. A single backplane clock can control this way the drift in the oscillators on all the devices in the non-star controller slots. The GX1034 board, when plugged into the star-controller slot (slot 2) can replace the PXI backplane clock with its much more stable oscillator. Alternatively, when plugged into one of the non-star controller slots (slots 3 and higher), the GX1034 board can be programmed that its clock source will use the PXI backplane clock see the GxCALSetClockSource function for details.

This call is valid only if the GX1034 board is plugged into the star-controller slot (slot 2) of the PXI chassis.

### Example

The following example returns the connection state of the on board reference Clock to the PXI reference Clock:

```
SHORT nState;
GxCALGetRefClockToPxiRefClockState (nHandle, &nState, &nStatus);
```

### See Also

**GxCALGetRefClockToPxiRefClockState, GxCALGetErrorString**

## GxCalGetDriverSummary

---

### Purpose

Returns the driver description string and version number.

### Syntax

**GxCalGetDriverSummary** (*szSummary*, *nSummaryMaxLen*, *pdwVersion*, *pnStatus*)

### Parameters

Name	Type	Comments
pszSummary	LPSTR	Buffer to receive the summary string.
nSummaryMaxLen	SHORT	Buffer size passed by pszSummary.
pdwVersion	LPDWORD	Driver version
pnStatus	LPSHORT	Returned status: 0 on success, negative number on failure.

### Example

The following example returns the driver summary.

```
SHORT nHandle, nStatus;  
DWORD dwVersion;  
CHAR szSummary[128];  
GxCalGetDriverSummary(szSummary, 128, &dwVersion, &nStatus);  
After the function call the parameter SzSummary will be set to:  
"GXCAL Driver for GX1034, Version 1.00, Copyright(c) 2008 Geotest - MTS inc."
```

### See Also

**GxCalGetErrorString**

## GxCalGetFunction

---

### Purpose

Returns the connected function and the function settings.

### Syntax

**GxCalGetFunction** (*nHandle*, *pnFunction*, *pdValue1*, *pdValue2*, *pnStatus*)

### Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1034 board.
pnFunction	PSHORT	Returned connected function number (see comments for details): GXCAL_FUNCTION_NOT_CONNECTED: No function connected. GXCAL_FUNCTION_4WIRE: 4-wires resistors is connected. GXCAL_FUNCTION_2WIRE: 2-wires resistors is connected. GXCAL_FUNCTION_POSITIVE_VDC: Positive DC voltage is connected. GXCAL_FUNCTION_NEGATIVE_VDC: Negative DC voltage is connected. GXCAL_FUNCTION_AC: AC voltage is connected. GXCAL_FUNCTION_TTL_CLOCK: TTL clock is connected. GXCAL_FUNCTION_CURRENT_SOURCE: Current source is connected GXCAL_FUNCTION_MEASURE_VDC: Output voltage measuremnt is connected. GXCAL_FUNCTION_MEASURE_RESISTANCE: Cutput voltage measuremnt and Current source re connected.
pdValue1	PDOUBLE	See comments.
pdValue2	PDOUBLE	See comments.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure. See comments for details.

### Comments

Returned function is as follow:

#### Connected function: 4-wires resistors (GXCAL\_FUNCTION\_4WIRE):

- *pdValue1*: returns the actual value of the connected 4-wires ressitore, supported values are: 1, 10, 100  
 There are three 4-wire resistors on-board of 1Ω, 10Ω, 100Ω. Actual measured values are stored in the on-board EEPROM and can be retrieve by calling the **GxCalGetFunctionMeasuredValue** function.
- *pdValue2*: Not Used.

#### Connected function: 2-wires resistors (GXCAL\_FUNCTION\_2WIRE):

- *pdValue1*: returns the actual value of the connected 2-wires ressitore, supported values are: 1, 10, 100, 1000, 10000, 100000, 1000000, 10000000.
- There are eight 2-wire resistors on-board of 1Ω, 10Ω, 100Ω, 1KΩ, 10KΩ, 100KΩ, 1MΩ, 10MΩ. Actual measured values are stored in the on-board EEPROM and can be retrieved by calling the **GxCalGetFunctionMeasuredValue** function.
- *pdValue2*: Not Used.

#### Connected function: Positive DC voltage (GXCAL\_FUNCTION\_POSITIVE\_VDC):

- *pdValue1*: returns the actual value of the connected poitive VDC, supported values are: 9.0V, 0.9V, 0.09V, 0.009V.

There are four positive DC voltages on-board of 9.0V, 0.9V, 0.09V, 0.009V, actual measured values are stored in the on-board EEPROM and can be retrieved by calling the **GxCalGetFunctionMeasuredValue** function.

- *pdValue2*: Not Used.

**Connected function: Negative DC voltage (GXCAL\_FUNCTION\_NEGATIVE\_VDC):**

- *pdValue1* returns the actual value of the connected negative VDC, supported values are: -9.0V, -0.9V, -0.09V, -0.009V.

There are four negative DC voltages on-board of -9.0V, -0.9V, -0.09V, -0.009V, actual measured values are stored in the on-board EEPROM and can be retrieved by calling the **GxCalGetFunctionMeasuredValue** function.

- *pdValue2*: Not Used.

**Connected function: AC voltage (GXCAL\_FUNCTION\_AC):**

- *pdValue1* returns the actual value of the connected VAC, VAC (nominal) voltage supported values are: 3..5VAC, 0.35VAC, 0.035VAC, 0.0035VAC.

There are four AC voltages on-board of 3..5VAC, 0.35VAC, 0.035VAC and 0.0035VAC, actual measured values are stored in the on-board EEPROM and can be retrieved by calling the **GxCalGetFunctionMeasuredValue** function.

- *pdValue2* returns the actual value of the connected VAC frequency, supported values are: 100, 1000, 10000, 100000, 50, 60, 400.

The supported AC frequencies are 100Hz, 1KHz, 10KHz, 100KHz, 50Hz, 60Hz and 400Hz, actual measured values are stored in the on-board EEPROM and can be retrieved by calling the **GxCalGetFunctionMeasuredValue** function.

**Connected function: TTL clock (GXCAL\_FUNCTION\_TTL\_CLOCK):**

- *pdValue1*: returns the TTL clock divider value, supported values are integers 1 through 10000000.

The TTL clock frequency is set by deviding an on-board 10MHz time base with integer values between 1 and 10e6 with a Duty Cycle of 50%.

- *pdValue2*: Not Used.

**Connected function: Current Source (GXCAL\_FUNCTION\_CURRENT\_SOURCE):**

- *pdValue1*: returns the the ouput current settings, range is 0.0A through 0.0199A.
- *pdValue2*: Not Used.

Connect to the ouptut the bord's current source.

**Note:** Make sure that there ar noe components in the connected cuircuit that can be damaged by the 20mA (Max) current settings.

**Connected function: Output voltage measurement (GXCAL\_FUNCTION\_MEASURE\_VDC):**

Connect the board's internal ADC to the output connector. Voltage measurements can then be done by calling **GxCalMeasure**.

- *pdValue1*: Not Used.
- *pdValue2*: Not Used.

**Connected function: Resistance measurement (GXCAL\_FUNCTION\_MEASURE\_RESISTANCE):**

- *pdValue1*: returns the the output current settings, range is 0.0A through 0.0199A.
- *pdValue2*: Not Used.

Connect the output the board's current source and the internal ADC to the output connector. The user can then measure the output resistance by calling **GxCalMeasure** function. Calling the function with different output current settings (lower or higher values) can increase the accuracy and stability of the resistance measurements.

**Note:** Make sure that there are no components in the connected circuit that can be damaged by the 20mA (Max) current settings.

**Example**

The following example returns the function and function settings:

```
SHORT nHandle, nFunction, nStatus;
DOUBLE dValue, dValue1;
GxCalGetFunction (nHandle, &nFunction, &dValue, &dValue1, &nStatus);
```

**See Also**

**GxCalSetFunction, GxCalMeasure, GxCalGetErrorString**

## GxCALGetFunctionStoredMeasuredValue

---

### Purpose

Returns the specified function stored measured values.

### Syntax

**GxCALGetFunctionStoredMeasuredValue** (*nHandle*, *nFunction*, *dValue1*, *dValue2*, *pdMeasuredValue1*, *pdMeasuredValue2*, *pnStatus*)

### Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1034 board.
pnFunction	PSHORT	Function numbers (see comments for details): GXCAL_FUNCTION_NOT_CONNECTED: No function connected. GXCAL_FUNCTION_4WIRE: 4-wires resistors connected. GXCAL_FUNCTION_2WIRE: 2-wires resistors connected. GXCAL_FUNCTION_POSITIVE_VDC: Positive DC voltage connected. GXCAL_FUNCTION_NEGATIVE_VDC: Negative DC voltage connected. GXCAL_FUNCTION_AC: AC voltage connected.
dValue	DOUBLE	See comments.
<i>dValue1</i>	DOUBLE	See comments.
pdMeasuredValue	PDOUBLE	See comments.
<i>pdMeasuredValue1</i>	PDOUBLE	See comments.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure. See comments for details.

### Comments

Returns the specified function stored measured values are as follow:

#### Connected function: 4-wires resistors (GXCAL\_FUNCTION\_4WIRE):

- *dValue*: specified supported resistance values are: 1, 10, 100
- *pdValue1*: Not Used write as zero.
- *pdMeasuredValue*: returns the stored measured resistance value that corelates to *dValue* value.

There are three 4-wire resistors on-board of 1Ω, 10Ω, 100Ω. Actual measured values are stored in the on-board EEPROM.

- *pdMeasuredValue1*: Not Used

#### Connected function: 2-wires resistors (GXCAL\_FUNCTION\_2WIRE):

- *dValue*: specified supported resistance values are: 1, 10, 100, 1000, 10000, 100000, 1000000, 10000000
- *pdValue1*: Not Used write as zero.
- *pdMeasuredValue*: returns the stored measured resistance value that corelates to *dValue* value.

There are eight 2-wire resistors on-board of 1Ω, 10Ω, 100Ω, 1KΩ, 10KΩ, 100KΩ, 1MΩ, 10MΩ. Actual measured values are stored in the on-board EEPROM.

- *pdMeasuredValue1*: Not Used

**Connected function: Positive DC voltage (GXCAL\_FUNCTION\_POSITIVE\_VDC):**

- *pdValue* supported values are: 9.0V, 0.9V, 0.09V, 0.009V.
- *pdValue1*: Not Used.
- *pdMeasuredValue*: returns the stored measured voltage value that correlates to *dValue* value.

There are four positive DC voltages on-board of 9.0V, 0.9V, 0.09V, 0.009V. Actual measured values are stored in the on-board EEPROM.

- *pdMeasuredValue1*: Not Used

**Connected function: Negative DC voltage (GXCAL\_FUNCTION\_NEGATIVE\_VDC):**

- *pdValue* supported values are: -9.0V, -0.9V, -0.09V, -0.009V.
- *pdValue1*: Not Used.
- *pdMeasuredValue*: returns the stored measured voltage value that correlates to *dValue* value.

There are four negative DC voltages on-board of -9.0V, -0.9V, -0.09V, -0.009V. Actual measured values are stored in the on-board EEPROM.

- *pdMeasuredValue1*: Not Used

**Connected function: AC voltage (GXCAL\_FUNCTION\_AC):**

- *dValue* AC voltage supported values are: 9.0VAC, 0.9VAC, 0.09VAC, 0.009VAC.
- *dValue1* AC frequency supported values are: 100, 1000, 10000, 100000, 50, 60, 400.
- *pdMeasuredValue*: returns the stored measured AC voltage value that correlates to *dValue* value.
- *pdMeasuredValue1*: returns the stored measured AC frequency for the specified voltage (*dValue*) value.

The supported AC frequencies are 100Hz, 1KHz, 10KHz, 100KHz, 50Hz, 60Hz and 400Hz. Actual measured values are stored in the on-board EEPROM.

Note: This function allows the user to retrieve all the actual (measured) values for each function.

**Example**

The following example returns the function and function settings:

```
SHORT nHandle, nFunction, nStatus;
DOUBLE dValue, dValue1;
GxCalGetFunction (nHandle, &nFunction, &dValue, &dValue1, &nStatus);
```

**See Also**

**GxCalSetFunction, GxCalMeasure, GxCalGetErrorString**

## GxCalInitialize

---

### Purpose

Initializes the driver for the specified PXI slot using the HW device driver.

### Syntax

**GxCalInitialize** (*nSlot*, *pnHandle*, *pnStatus*)

### Parameters

Name	Type	Comments
nSlot	SHORT	GX1034 board slot number.
pnHandle	PSHORT	Returned Handle for a GX1034 board.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

### Comments

The Geotest HW device driver is installed with the driver and is the default device driver. The function returns a handle that for use with other Counter functions to program the board. The function does not change any of the board settings.

The specified PXI slot number is displayed by the **PXI/PCI Explorer** applet that can be opened from the Windows **Control Panel**. You may also use the label on the chassis below the PXI slot where the board is installed. The function accepts two types of slot numbers:

- A combination of chassis number (chassis # x 256) with the chassis slot number. For example 0x105 (chassis 1 slot 5).
- 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 (1-255).

### Example

The following example initializes two GX1034 boards at slot 1 and 2.

```
SHORT nHandle1, nHandle2, nStatus;
Gx6616Initialize (1, &nHandle1, &nStatus);
Gx6616Initialize (2, &nHandle2, &nStatus);
if (nHandle1==0 || nHandle2==0)
{
    printf("Unable to Initialize the boards")
    return;
}
```

### See Also

**GxCalInitializeVisa**, **GxCalGetErrorString**, **GxCalReset**

## GxCalInitializeVisa

---

### Purpose

Initializes the driver for the specified PXI slot using the default VISA provider.

### Syntax

**GxCalInitializeVisa** (*szVisaResource*, *pnHandle*, *pnStatus*)

### Parameters

Name	Type	Comments
<i>szVisaResource</i>	LPCTSTR	String identifying the location of the specified board in order to establish a session.
<i>pnHandle</i>	PSHORT	Returned Handle (session identifier) that can be used to call any other operations of that resource
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, 1 on failure.

### Comments

The **GxCalInitializeVisa** 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 Counter boards at PXI bus 5 and device 11.

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

### See Also

**GxCalInitialize**, **GxCalGetErrorString**, **GxCalReset**

## GxCalPanel

---

### Purpose

Opens a virtual panel used to interactively control the counter.

### Syntax

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

### Parameters

Name	Type	Comments
<i>pnHandle</i>	PSHORT	Handle to a GX1034 board.
<i>hwndParent</i>	HWND	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 window and 1 for modal window.
<i>phwndPanel</i>	HWND	Returned window handle for the panel.
<i>pnStatus</i>	PSHORT	Returned status: 0 on success, negative number on failure.

### Comments

The function is used to create the panel window. The panel window may be open 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 closes. In that case, the *pnHandle* may return the handle created by the user using the panel Initialize dialog. This handle may be used when calling other counter functions.

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

### Example

The following example opens the panel in modal mode:

```
DWORD dwPanel;
SHORT nHandle=0, nStatus;

GxCalPanel(&nHandle, 0, 1, &dwPanel, &nStatus);
```

### See Also

**GxCalInitialize**, **GxCalGetErrorString**

## GxCalMeasure

---

### Purpose

Measure the output voltage or resistance according to the specified voltage range and connected function.

### Syntax

**GxCalMeasure** (*nHandle*, *nVoltageRange*, *pdVoltage*, *pdValue*, *pnStatus*)

### Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1034 board.
nVoltageRange	SHORT	Voltage range should be set as follows: 3. GXCAL_MEASURE_VOLTAGE_RANGE_MINUS_10_TO_10: voltage range is from -10V to +10V with resolution of 0.3mV.
pdValue	PDOUBLE	Returned measurement.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

### Comments

The function measures the output voltage or resistance according to the specified voltage range only if the board is set to either GXCAL\_FUNCTION\_MEASURE\_VDC function or GXCAL\_FUNCTION\_MEASURE\_RESISTANCE function otherwise the function returns an error.

If the board was set to GXCAL\_FUNCTION\_MEASURE\_VDC function then the function returns the measured voltage between pins 2 and 6 on the output DIN connector in volts. The user can specify the voltage range of the measurement allowing for better resolution for small voltage amplitudes.

If the board was set to GXCAL\_FUNCTION\_MEASURE\_RESISTANCE function then the function returns the measured resistance between pins 2 and 6 on the output DIN connector in OHms. The user can specify different voltage range allowing for better resolution for high resistance values, as well as change the output current settings by calling the GXCAL\_FUNCTION\_MEASURE\_RESISTANCE function.

### Example

The following example sets the function to measure VDC and then take a measurement the output voltage using 0-10V range:

```
SHORT nHandle, nStatus;
DOUBLE dVoltage;

GxCalSetFunction (nHandle, GXCAL_FUNCTION_MEASURE_VDC, 0, 0, &nStatus);
GxCalMeasure (nHandle, 0, TRUE, GXCAL_MEASURE_VOLTAGE_RANGE_MINUS_10_TO_10, &dVoltage, &nStatus);
```

### See Also

**GxCalReset, GxCalSetFunction, GxCalGetFunction, GxCalGetErrorString**

## **GxCalReset**

---

### **Purpose**

Resets the counter board to its default settings.

### **Syntax**

**GxCalReset** (*nHandle*, *pnStatus*)

### **Parameters**

<b>Name</b>	<b>Type</b>	<b>Comments</b>
nHandle	SHORT	Handle to a GX1034 board.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

### **Comments**

After calling reset, no function is connected to the output connector.

### **Example**

The following example initializes and resets the counter board:

```
GxCalInitialize (1, &nHandle, &nStatus);  
GxCalReset (nHandle, &nStatus);
```

### **See Also**

**GxCalInitialize**, **GxCalGetErrorString**

## GxCalSetFunction

---

### Purpose

Returns the connected function and the function settings.

### Syntax

**GxCalGetFunction** (*nHandle*, *pnFunction*, *pdValue1*, *pdValue2*, *pnStatus*)

### Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1034 board.
nFunction	SHORT	Connected function number (see comments for details): GXCAL_FUNCTION_NOT_CONNECTED: No function connected. GXCAL_FUNCTION_4WIRE: 4-wires resistors is connected. GXCAL_FUNCTION_2WIRE: 2-wires resistors is connected. GXCAL_FUNCTION_POSITIVE_VDC: Positive DC voltage is connected. GXCAL_FUNCTION_NEGATIVE_VDC: Negative DC voltage is connected. GXCAL_FUNCTION_AC: AC voltage is connected. GXCAL_FUNCTION_TTL_CLOCK: TTL clock is connected. GXCAL_FUNCTION_CURRENT_SOURCE: Current source is connected GXCAL_FUNCTION_MEASURE_VDC: Output voltage measuremnt is connected. GXCAL_FUNCTION_MEASURE_RESISTANCE: Cutput voltage measuremnt and Current sourcea re connected.
dValue1	DOUBLE	See comments.
<i>dValue2</i>	DOUBLE	See comments.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure. See comments for details.

### Comments

Connected function is as follow:

#### Connected function: 4-wires resistors (GXCAL\_FUNCTION\_4WIRE):

- *dValue1*: supported values are: 1, 10, 100

There are three 4-wire resistors on-board of 1Ω, 10Ω, 100Ω. Actual measured values are stored in the on-board EEPROM and can be retrieve by calling the **GxCalGetFunctionMeasuredValue** function.

- *dValue2*: Not Used.

#### Connected function: 2-wires resistors (GXCAL\_FUNCTION\_2WIRE):

- *dValue1*: supported values are: 1, 10, 100, 1000, 10000, 100000, 1000000, 10000000.

There are eight 2-wire resistors on-board of 1Ω, 10Ω, 100Ω, 1KΩ, 10KΩ, 100KΩ, 1MΩ, 10MΩ. Actual measured values are stored in the on-board EEPROM and can be retrieved by calling the **GxCalGetFunctionMeasuredValue** function.

- *dValue2*: Not Used.

#### Connected function: Positive DC voltage (GXCAL\_FUNCTION\_POSITIVE\_VDC):

- *dValue1*: supported values are: 9.0V, 0.9V, 0.09V, 0.009V.

There are four positive DC voltages on-board of 9.0V, 0.9V, 0.09V, 0.009V, actual measured values are stored in the on-board EEPROM and can be retrieved by calling the **GxCalGetFunctionMeasuredValue** function.

- *dValue2*: Not Used.

**Connected function: Negative DC voltage (GXCAL\_FUNCTION\_NEGATIVE\_VDC):**

- *dValue1*: supported values are: -9.0V, -0.9V, -0.09V, -0.009V.

There are four negative DC voltages on-board of -9.0V, -0.9V, -0.09V, -0.009V, actual measured values are stored in the on-board EEPROM and can be retrieved by calling the **GxCalGetFunctionMeasuredValue** function.

- *dValue2*: Not Used.

**Connected function: AC voltage (GXCAL\_FUNCTION\_AC):**

- *dValue1*: VAC (nominal) voltage supported values are: 3..5VAC, 0.35VAC, 0.035VAC, 0.0035VAC.

There are four AC voltages on-board of 3..5VAC, 0.35VAC, 0.035VAC and 0.0035VAC, actual measured values are stored in the on-board EEPROM and can be retrieved by calling the **GxCalGetFunctionMeasuredValue** function.

- *dValue2*: VAC frequency supported values are: 100, 1000, 10000, 100000, 50, 60, 400.

The supported AC frequencies are 100Hz, 1KHz, 10KHz, 100KHz, 50Hz, 60Hz and 400Hz, actual measured values are stored in the on-board EEPROM and can be retrieved by calling the **GxCalGetFunctionMeasuredValue** function.

**Connected function: TTL clock (GXCAL\_FUNCTION\_TTL\_CLOCK):**

- *dValue1*: TTL clock divider value, supported values are integers 1 through 10000000.

The TTL clock frequency is set by dividing an on-board 10MHz time base with integer values between 1 and 10e6 with a Duty Cycle of 50%.

- *dValue2*: Not Used.

**Connected function: Current Source (GXCAL\_FUNCTION\_CURRENT\_SOURCE):**

- *dValue1*: output current settings, range is 0.0A through 0.0199A.
- *dValue2*: Not Used.

Connect to the output the board's current source.

**Note:** Make sure that there are no components in the connected circuit that can be damaged by the 20mA (Max) current settings.

**Connected function: Output voltage measurement (GXCAL\_FUNCTION\_MEASURE\_VDC):**

Connect the board's internal ADC to the output connector. Voltage measurements can then be done by calling **GxCalMeasure**.

- *dValue1*: Not Used.
- *dValue2*: Not Used.

**Connected function: Resistance measurement (GXCAL\_FUNCTION\_MEASURE\_RESISTANCE):**

- *dValue1*: output current settings, range is 0.0A through 0.0199A.
- *dValue2*: Not Used.

Connect the output the board's current source and the internal ADC to the output connector. The user can then measure the output resistance by calling **GxCalMeasure** function. Calling the function with different output current settings (lower or higher values) can increase the accuracy and stability of the resistance measurements.

**Note:** Make sure that there are no components in the connected circuit that can be damaged by the 20mA (Max) current settings.

**Example**

The following example sets the function to be 4-wires resistance of 100:

```
SHORT nHandle, nStatus;
DOUBLE dValue, dValue1;
GxCalSetFunction (nHandle, GXCAL_FUNCTION_4WIRE, 100, 0, &nStatus);
```

**See Also**

**GxCalGetFunction, GxCalMeasure, GxCalGetErrorString**

## GxCALSetRefClockToPxiRefClockState

---

### Purpose

Sets the connection state of the on board reference Clock to the PXI reference Clock.

### Syntax

**GxCALSetRefClockToPxiRefClockState** (*nHandle*, *nState*, *pnStatus*)

### Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX1034 board.
nState	SHORT	Counter Reference Clock to PXI Reference Clock state: 0 GXCAL_REF_CLOCK_TO_PXI_REF_CLOCK_OFF: Disconnect the on board reference Clock from the PXI Reference Clock. 1 GXCAL_REF_CLOCK_TO_PXI_REF_CLOCK_ON: Connect the on board reference Clock to the PXI Reference Clock.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

### Comments

The PXI chassis has a 10 MHz clock line (PXI\_CLK10) in its backplane that allows devices plugged into the non-star controller slots (slots 3 and higher) to phase-lock their oscillators to the backplane clock. A single backplane clock can control this way the drift in the oscillators on all the devices in the non-star controller slots. The GX1034 board, when plugged into the star-controller slot (slot 2) can replace the PXI backplane clock with its much more stable oscillator. Alternatively, when plugged into one of the non-star controller slots (slots 3 and higher), the GX1034 board can be programmed that its clock source will use the PXI backplane clock see the GxCALSetClockSource function for details.

This call is valid only if the GX1034 board is plugged into the star-controller slot (slot 2) of the PXI chassis.

### Example

The following example enables the connection from the Counter Reference clock to the PXI Reference clock:

```
GxCALSetRefClockToPxiRefClockState (nHandle, GXCAL_REF_CLOCK_TO_PXI_REF_CLOCK_ON, &nStatus);
```

### See Also

**GxCALGetRefClockToPxiRefClockState**, **GxCALGetErrorString**

## GxCalGetErrorString

---

### Purpose

Returns the error string associated with the specified error number.

### Syntax

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

### Parameters

Name	Type	Comments
nError	SHORT	Error number.
pszMsg	PSTR	Buffer to the returned error string.
nErrorMaxLen	SHORT	The size of the error string buffer.
pnStatus	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

0	No error has occurred
-1	Unable to open the HW driver. Check if HW is properly installed
-2	Board does not exist in this slot/base address
-3	Different board exist in the specified PCI slot/base address
-4	PCI slot not configured properly. You may configure using the PciExplorer from the Windows Control Panel
-5	Unable to register the PCI device
-6	Unable to allocate system resource for the device
-7	Unable to allocate memory
-8	Unable to create panel
-9	Unable to create Windows timer
-10	Bad or Wrong board EEPROM
-11	Not in calibration mode
-12	Board is not calibrated
-13	Function is not supported by the specified board

### General Parameter Errors

-20	Invalid or unknown error number
-21	Invalid parameter
-22	Illegal slot number
-23	Illegal board handle
-24	Illegal string length
-25	Illegal operation mode

### Parameter Errors

-40	Standard value not supported
-41	Operation generated Timeout.
-42	Invalid current value, values are 0 to 0.0020.
-43	Invalid measure voltage channel, only 0 allowed.
-44	Invalid measure voltage mode, only differential mode is allowed.
-45	Invalid measure voltage range.

**In-System Calibration Errors**

- 50 Error: In-System Calibration was not stated.
- 51 Error: In-System Calibration invalid device.
- 52 Error: In-System Calibration invalid device data index.
- 53 Error: In-System Calibration device not in calibration mode.
- 54 Error: In-System Calibration invalid device data.
- 55 Error: In-System Calibration 10MHz time base.

**Built In Test error/warning**

- 60 Built In test: Self-test adaptor is not connected.
- 61 Built In test: error closing a relay.

**Example**

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

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

# Index

- .
- .NET ..... ii
- :
- : GX1034 Front Connectors.....22
- A**
- Applications** .....4
- Architecture ..... 1, 8
- ATEasy .....ii, 24, 25, 27, 28
- B**
- Block Diagram.....8
- Board Description..... 1, 5
- Board Handle.....30
- Borland .....ii, 24, 27, 28
- Borland-Delphi .....28
- C**
- C/C++ .....24, 25, 27
- C++ .....27
- Connector .....22
- Connectors.....21, 22
- Corrupt files.....26
- D**
- DB-9 Connector** .....23
- Delphi .....ii, 24, 27, 28
- Directories .....24
- Distributing.....31
- Driver
  - Directory.....24
  - Files .....24
- Driver Version .....30
- E**
- Error Handling.....30
- Error-Handling .....30
- Example.....25
- F**
- Functions Reference .....35
- G**
- GX1034 ..... 1
- GX1034 ..... 1
- GX1034** .....5
- GXCAL ..... 1, 16
  - Driver-Description.....27
  - Header-file .....27
  - Help-File-Description .....25
  - Panel-File-Description.....24
  - Supported-Development-Tools.....27
- GxCal Functions .....36
- GXCAL.BAS.....24, 27
- GXCAL.DLL.....24, 27, 28
- GXCAL.EXE.....16, 31
- GXCAL.H .....24, 27
- GXCAL.LIB .....24, 27
- GXCAL.PAS .....24, 28
- GXCALBC.LIB.....24, 27
- GxCalGetBoardSummary .....37
- GxCalGetBoardTemperature.....38
- GxCalGetDriverSummary .....29, 30, 40
- GxCalGetErrorString .....30, 35, 55
- GxCalGetFunction.....41
- GxCalGetFunctionStoredMeasuredValue44
- GxCalGetRefClockToPxiRefClockState 39
- GxCalInitialize ..... 12, 29, 30, 35, 46
- GxCalInitializeVisa ..... 12, 29, 30, 47
- GxCalMeasure.....49
- GxCalPanel.....31, 48
- GXCALPANEL.EXE.....24, 31
- GxCalReset.....30, 50
- GxCalSetFunction .....51
- GxCalSetRefClockToPxiRefClockState 54
- H**
- Handle ..... 19, 20, 29, 30

HW .....	16, 20, 24, 27, 31
<b>I</b>	
Installation Directories .....	24
Installation: .....	19, 21
<b>L</b>	
LabView .....	24, 26
Listing .....	32
<b>N</b>	
<i>nHandle</i> .....	28, 30
<b>O</b>	
OnError .....	28
<b>P</b>	
Panel .....	11, 14, 24, 26, 29, 31, 48
Pascal .....	24, 27, 28
PCI .....	24
Plug & Play .....	20
<i>pnStatus</i> .....	30, 35
Program-File-Descriptions .....	24
Programming	
Borland-Delphi .....	28
Error-Handling .....	30
Panel-Program .....	31
Programming the Board .....	27
PXI .....	15, 18, 19, 20, 21, 29
PXI/PCI Explorer .....	12, 18, 29, 30, 46, 47
PXIeSYS.INI .....	12
PXISYS.INI .....	12
<b>R</b>	
README.TXT .....	24, 25
Readme-File .....	25
Reset .....	30
<b>S</b>	
Sample .....	32
Sample Programs .....	32
Setup .....	16, 24, 26
Setup Maintenance .....	26
Setup-and-Installation .....	15
Slot .....	12, 19, 21, 29
Specifications .....	1, 9
System	
Directory .....	24
System-Requirements .....	15
<b>U</b>	
Using the GXCAL driver functions .....	29
<b>V</b>	
Virtual Panel. 11, 12, 13, 14, 16, 24, 29, 48	
About Page .....	14
Initialize Dialog .....	12
Setup Page .....	13
Virtual Panel Description .....	1
VISA .....	12, 29, 30, 36, 46, 47
Visual Basic .....	ii, 27
Visual C++ .....	ii, 24, 27