GXSW

Switching Instruments

GX6196

96 Channel DPST Relay and Control

6U PXI Carrier Card

User's Guide

Last Updated: April 3, 2017

MARVIN TEST SOLUTIONS

Safety and Handling

Each product shipped by Marvin Test Solutions 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 diskette for installation. Store the original diskette 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

Marvin Test Solutions products are warranted against defects in materials and workmanship for a period of 12 months. Software products and accessories are warranted for 3 months. Unless covered by software support or maintenance agreement. Marvin Test Solutions 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 Marvin Test Solutions Customer Technical Services department via http://www.marvintest.com/magic the Marvin Test Solutions on-line support system.

If You Need Help

Visit our web site at http://www.marvintest.com for more information about Marvin Test Solutions 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 question please use the following link: http://www.marvintest.com/magic

You can also use Marvin Test Solutions technical support phone line (949) 263-2222. This service is available between 7:30 AM and 5:30 PM Pacific Standard Time.

Disclaimer

In no event shall Marvin Test Solutions 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 Marvin Test Solutions has been advised of the possibility for such damages.

Copyright

Trademarks

ATEasy®, CalEasy, DIOEasy®, DtifEasy, WaveEasy	Marvin Test Solutions (prior name is Geotest - Marvin Test Systems Inc.)
C++ Builder, Delphi	Embarcadero Technologies Inc.
LabVIEW, LabWindows™/CVI	National Instruments
Microsoft Developer Studio, Microsoft Visual C++, Microsoft Visual Basic, .NET, and Windows	Microsoft Corporation

All other trademarks are the property of their respective owners.

Table of Contents

Chapter 1 - Introduction

Manual Scope and Organization

Manual Scope

This manual provides all the information necessary for installation, operation, and maintenance of the **GX6196** PXI Relay and Control Board. The manual also covers the **GXSW** software package that includes the GX6196 driver. This manual assumes the reader has a general knowledge of PC based computers, Windows operating systems, and a general knowledge of modular test equipment.

Manual Organization

The GX6196 manual is organized in the following manner:

Chapter	Content
Chapter 1 – Introduction	Introduces the GX6196 manual and shows warning conventions used in the manual.
Chapter 2 – Overview	Provides the GX6196 list of features, description of the board, architecture, specifications and the virtual panel description and operation.
Chapter 3 –Installation and Connections	Provides instructions about how to install a GX6196 board and the GXSW software.
Chapter 4 – Functions Reference	Provides a list of the GXSW driver functions for the GX6196. 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.

Formatting Convention	Meaning
`Monospaced Text`	Examples of field syntax and programming samples.
Bold type	Words or characters you type as the manual instructs. For example: function or panel names.
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 GX6196 is a 6U PXI switch and control board that can be used to create custom switching, custom loads or specific UUT control interfaces. The board provides 96, DPST relays, 8 digital I/O lines, and a serial digital interface. All of these resources are available via (5) inter-board connectors which are used to interface the control / carrier board's resources to the user defined module (mezzanine) board. The module board provides all connections to the unit under test (UUT) via two 78 pin D-sub connectors. The GX6196 is supplied with a front panel which accepts (2) 78 pin D-sub connectors.

Features

The GX6196 features 96 DPST relays which can be individually controlled via the PXI bus and interface via four inter-board connectors to a module board assembly (user supplied). In addition, there are eight parallel digital LVTTL I/O control lines and a serial bus interface which are accessed via an additional board to board connector. Control for all of these digital interfaces is also supported via the PXI bus, providing control for circuitry located on the user defined module board. Figure 2-1 details the PXI relay and control board.

Figure 2-1: GX6196 Relay and Control Card

Specifications

The following table outlines the specifications of the GX6196:

Relay Specifications

Number of Relays	96, DPST
Relay Contact Resistance	<0.1Ω
Contact Life	1 x10e5 cycles (Rating At 30 VDC @ 2A)
Maximum Switchable Voltage	220 VDC
Maximum Switchable Current	2 A Max.
Contact Carry Current	2 A Max.
Operate Time	4 mSec (including bounce)
Release Time	4 mSec

Module Board Interface

Interface Connectors	5, 100 pin, Samtec SOLC-125-02-L-Q-A-P
ID Lines	4
Digital I/O Lines	8, LVTTL, individually programmable with read status
Serial Communication Interface	SDOut, SDIn, SClk, (4) Chip Enables; 1 MHz data rate
Power Interface	+5V, 5A; +3.3V, 5A; +12V, 1A; -12V, 1A; +1.2V, 1A (max current)

Carrier Power Requirements (Not including module board)

Operating Voltage	+3.3 VDC
Power Consumption	330 mA (max)
Relay Power	+5 V, 29 mA per relay

Physical

Size	6U PXI
Weight	18 oz.

Environmental

Temperature:	0 to +55 °C Operating -20 to +85 °C Storage
Vibration	5g at 500Hz
Shock	5g for 6ms

Virtual Panel Description

The **GXSW** software 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 **GX6196 Panel** from the **Marvin Test Solutions**, **GXSW** menu under the **Start** menu. The GX6196 virtual panel opens as shown here:

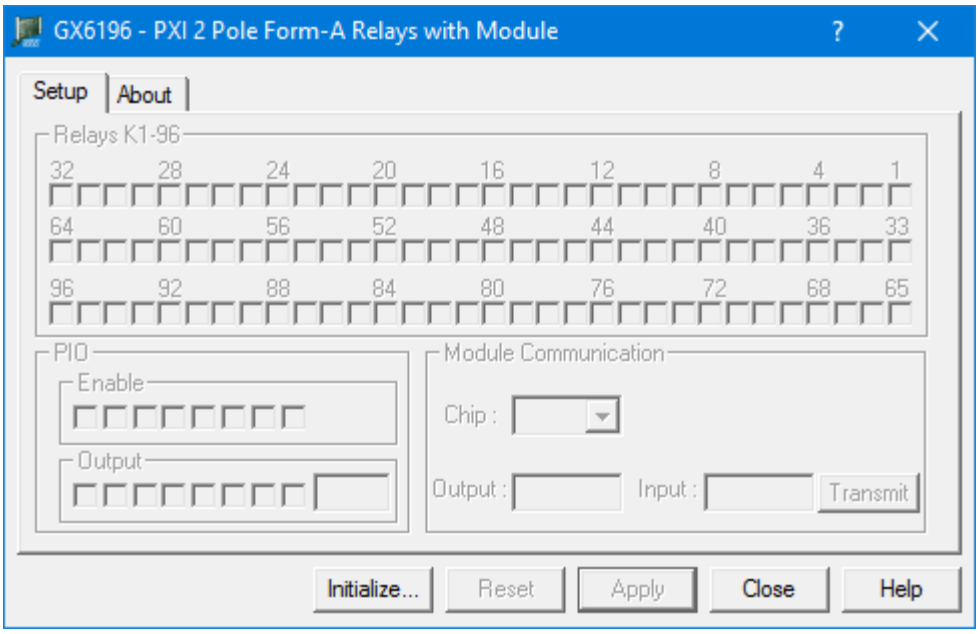


Figure 2-2: GX6196 Virtual Panel

Initialize – Opens the Initialize Dialog (see Initialize Dialog paragraph) in order to initialize the board driver. The current settings of the selected board will **not change after calling initialize**. The panel will reflect the current settings of the board 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 displays 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 board. The board settings **will not change** after initialize is called. Once initialized, the panel will reflect the current settings of the board.

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

1. **Use Marvin Test Solutions' 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 GX6196 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 0x104(chassis 1 Slot 4). This is the slot number argument (*nSlot*) passed by the panel when calling the driver **Gx6196nitialize** function which is used to initialize the driver for the specified board.

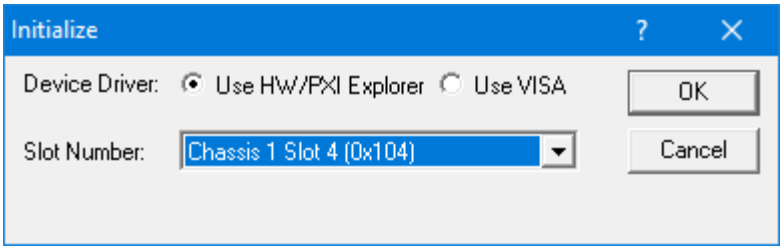


Figure 2-3: Initialize Dialog Box using Marvin Test Solutions' HW driver

2. **Use VISA** – This is a third-party device driver usually provided by National Instrument (NI-VISA). When 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 by Marvin Test Solutions **PXI/PCI Explorer**. The following figure shows PXI9::13::INSTR as the VISA resource (PCI bus 9 and Device 13). This is a VISA resource string argument (*szVisaResource*) which is passed by the panel when calling the driver **Gx6196InitializeVisa** function which initializes the driver for the specified board.

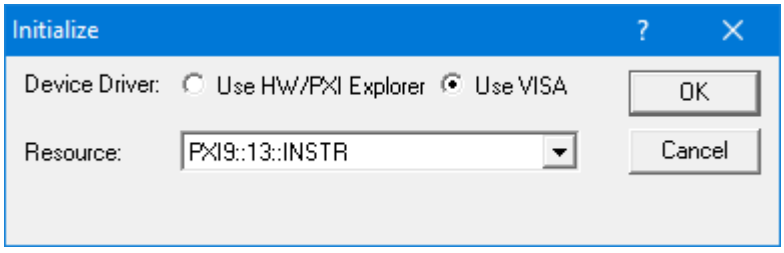


Figure 2-4: Initialize Dialog Box using VISA resources

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 **Group page** settings:

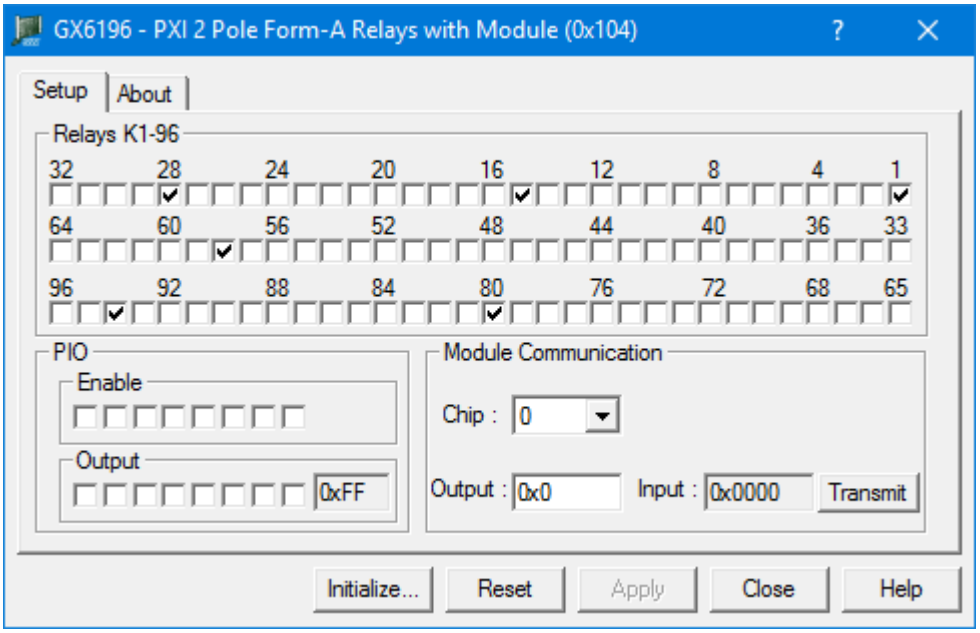


Figure 2-5: GX6196 Virtual Panel (Initialized)

The following controls are shown:

Relays K1-96 Display Area: Displays the current relay setting (closed or open). When a box in is checked, the relay is closed. Similarly, unchecked boxes mean relays are open.

PIO Enable Check Boxes: Displays the current PIO output enable setting (enabled or disabled). When a box is checked, the PIO channel output is enabled. Similarly, an unchecked box means that the PIO channel output is disabled.

PIO Output Check Boxes: Displays the current PIO output state (driving a high or low). When a box is checked, the PIO channel is outputting a TTL high. Similarly, an unchecked box means that the PIO channel is outputting a TTL low.

Module Communication Chip: Displays and selects the Chip Select line to use for serial communication to the Module.

Module Output: Displays and allows the user to enter the 16-bit word to serially transmit to the Module.

Module Input: Displays the 16-bit word received from the Module during the last serial transmission.

Transmit Button: Transmit data serially to the Module.

Virtual Panel About Page

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

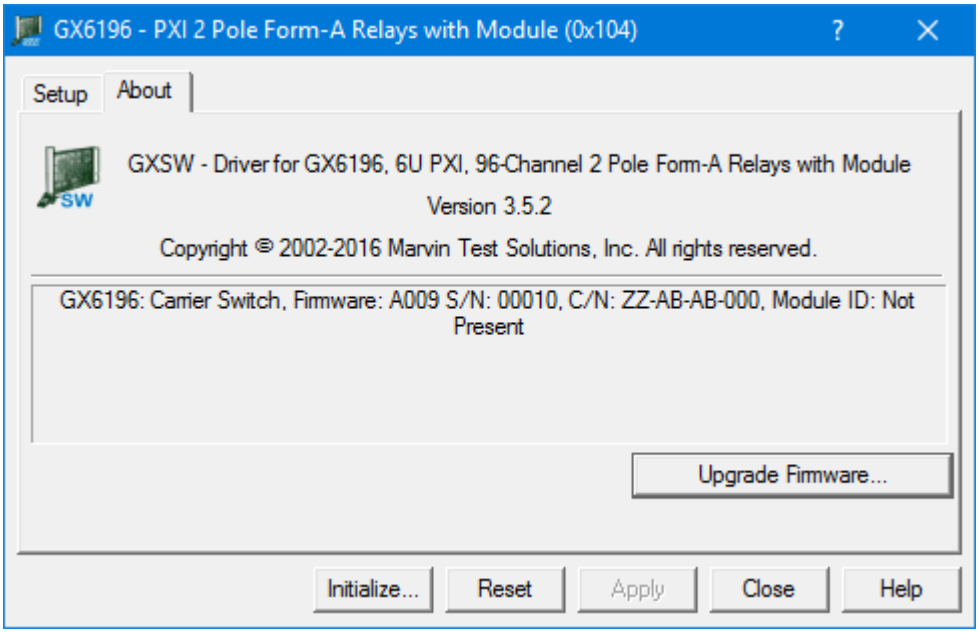


Figure 2-6:GX6196 Virtual Panel – About Page

The following controls are shown in the About page:

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

The **About** page also contains a button **Upgrade Firmware…** which is used to upgrade the board's FPGA firmware. This button maybe used only when the board requires upgrade as directed by Marvin Test Solutions 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 GX6196 board and installation instructions for the GX6196 (GXSW) 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
GX6196 Driver Installation	This Chapter
GX6196 Function Reference	Chapter 4

Packing List

All GX6196 boards have the same basic packing list, which includes:

1. GX6196 Board
2. CD that includes the GXSW software

Unpacking and Inspection

After removing the board from the shipping carton:

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

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

System Requirements

The GX6196 instrument board is designed for use with a 6U cPCI or PXI compatible chassis. The software is compatible with any computer system running Windows XP SP3, VISTA, 7, 8, and 10 (32/64 bit) and Windows 7 operating systems. In addition, Microsoft Windows Explorer version 4.0 or above is required to view the online help.

Installation of the GXSW Software

Before installing the board, it is recommended that you install the GXSW software as described in this section. To install the GXSW software, follow the instruction described below:

1. Insert the Marvin Test Solutions CD-ROM and locate the **GXSW.EXE** setup program. If your computer's Auto Run is configured, when inserting the CD, a browser will show several options. Select the Marvin Test Solutions 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 Marvin Test Solutions' web site (www.marvintest.com).
2. Run the GXSW setup and follow the instruction on the Setup screen to install the GXSW driver.

   Note: When installing under Windows, you may be required to restart the setup after logging-in as a user with Administrator privileges. This is required in-order to upgrade your system with newer Windows components and to install kernel-mode device drivers (HW driver) which are required by the GXSW 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 GXSW is to be installed. Either click Browse to set up a new folder, or click Next to accept the default entry. The default entry for 32 bit machines is **C:\Program Files\Marvin Test Solutions\GxSw**, and for 64 bit Windows **C:\Program Files (x86)\Marvin Test Solutions\GxSw**.
5. Select the type of Setup you wish and click **Next.** You can choose between **Typical**, **Run-Time** and **Custom** setups types. The **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. The **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 completion 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 lets you control the board interactively. The panel program can be started by selecting it from the Start, Programs, GXSW menu located in the Windows Taskbar.

Setup Maintenance Program

You can run the Setup again after GXSW 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 GXSW installation. The following options are available in Maintenance mode:

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

Select one of the options and click **Next** and follow the instruction on the screen until Setup is complete.

Overview of the GXSW Software

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

- **GXSW Panel** – Configures and controls the GX6196 various features via an interactive user interface.
- **GXSW driver** - A DLL based function library (GXSW.DLL (32 bit) and GXSW64.DLL (64 bit), located in the Windows System folder) used to program and control the board.
- **Programming files and examples** – Interface files and libraries for support of various programming tools such as C#, C++, VB, VB6, and LabVIEW. A complete list of files and development tools supported by the driver is included in subsequent sections of this manual.
- **Documentation** – On-Line help and User's Guide for the GX6196 board, GXSW driver and panel.

HW driver and PXI/PCI Explorer applet – HW driver allows the GXSW driver to access and program the supported boards. The explorer applet configures 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 and is used by Marvin Test Solutions instruments HW driver and VISA. The applet can be used to assign chassis numbers, Legacy Slot numbers and instrument alias names. The HW driver is installed and shared with all Marvin Test Solutions products to support accessing the PC resources. Similar to HW driver, VISA provides a standard way for instrument manufacturers and users to write and use instruments drivers. VISA is a standard maintained by the VXI Plug & Play System Alliance and the PXI Systems Alliance organizations (http://www.ivifoundation.org, http://www.pxisa.org/). The VISA resource manager such as National Instruments **Measurement & Automation** (NI-MAX) displays and configures instruments and their address (similar to Marvin Test Solutions' PXI/PCI Explorer). The GXSW driver can work with either HW or VISA to control an access the supported boards.

Installation Folders

The GX6169 driver files are installed in the default folder **C:\Program Files\Marvin Test Solutions\GxSw**, and For 64 bit machines they are installed in **C:\Program Files (x86)\Marvin Test Solutions\GxSw**.

You can change the default GxSw folder to one of your choosing at the time of installation.

During the installation, GxSw Setup creates and copies files to the following folders:

Name	Purpose / Contents
…\Marvin Test Solutions\GXSW	The GXSW folder. Contains panel programs, programming libraries, interface files and examples, on-line help files and other documentation.
…\Marvin Test Solutions\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 folder. GXSW Driver and example are copied to this directory only if ATEasy is installed to your machine.
…\Windows\System32, or …\Windows\SysWOW64 when running 64 bit Windows	Windows System directory. Contains the GXSW DLL driver, HW driver shared files and some upgraded system components, such as the HTML help viewer, etc.

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, **Marvin Test Solutions, HW, PXI/PCI Explorer**.
2. Identify Chassis and Controllers. After the PXI/PCI Explorer is started, it will scan your system for changes and will display the current configuration. The PXI/PCI Explorer automatically detects systems that have Marvin Test Solutions 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 which are used by these commands.
3. **Change chassis numbers, PXI devices Legacy Slot numbering and PXI devices Alias names**. These are optional steps and can be performed if you would like your chassis to have different numbers. Legacy slots numbers are used by older Marvin Test Solutions or VISA drivers. Alias names can provide a way to address a PXI device using a logical name (e.g. "DMM1"). For more information regarding slot numbers and alias names, see the **Gx6196Initialize** and **Gx6196InitializeVisa** functions.
4. Save your 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 your changes. The PXI/Explorer will prompt you to save the changes if changes were made or detected (an asterisk sign ' *' in the caption indicated changes).

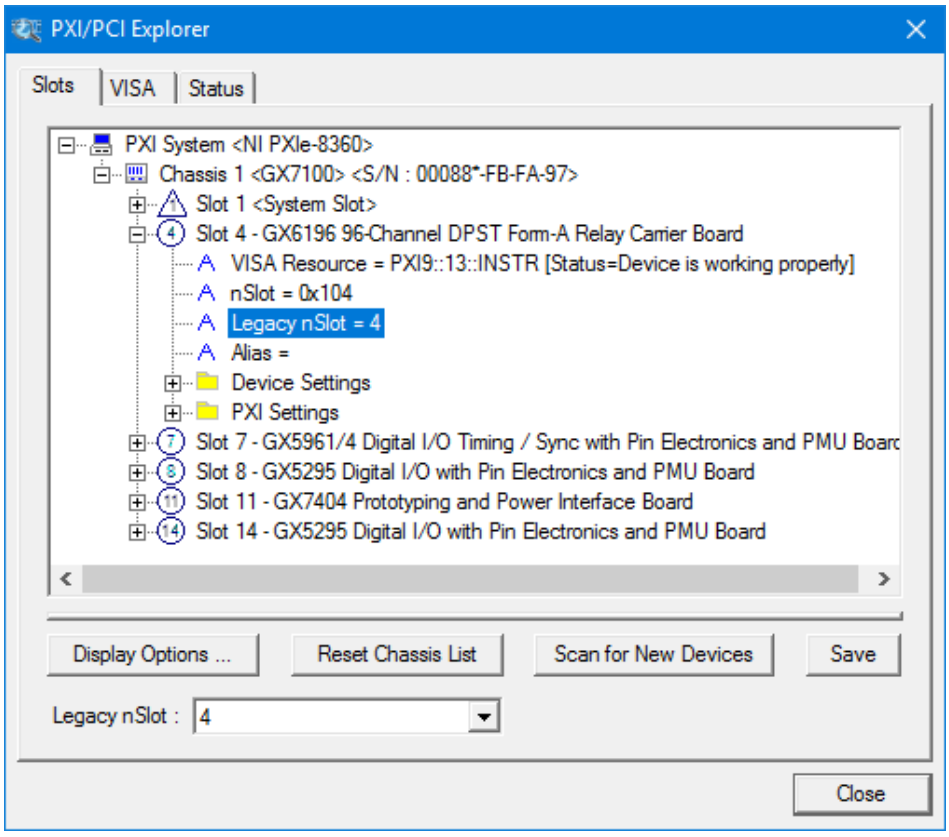


Figure 3-1: PXI/PCI Explorer

Board Installation

Before you Begin

- Install the GXSW 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 section in this chapter) are present.

Electric Static Discharge (ESD) Precautions

To reduce the risk of damage to the GX6196 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 GXSW Driver as described 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).

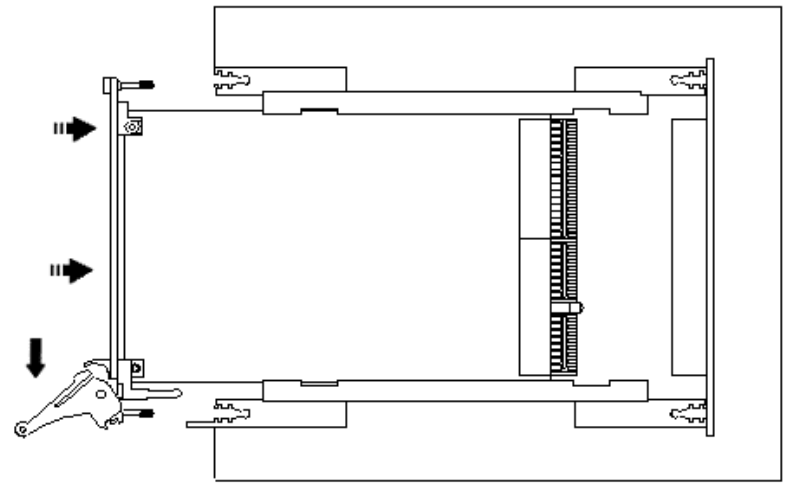


Figure 3-2: Ejector handles position during module insertion

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

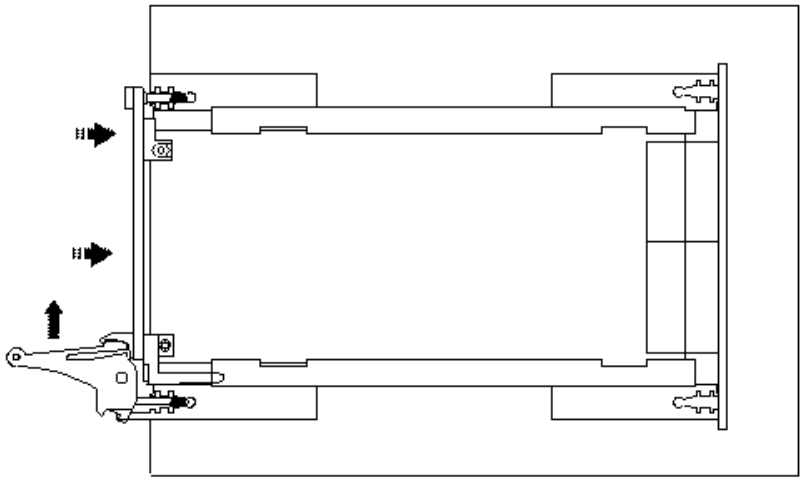


Figure 3-3: Ejector handles position after module insertion

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 notifies the user that a new board was found using the **New Hardware Found** wizard after restarting the system with the new board.

If another Marvin Test Solutions board software package was already installed, Windows will suggest using the driver information file: HW.INF. The file is located in your C:\Program Files\Marvin Test Solutions\HW folder, or on 64 bit machines C:\Program Files (x86)\Marvin Test Solutions\HW. 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 GXSW driver was not installed prior to the board installation), you may install the GXSW 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\Marvin Test Solutions\HW**, or on 64 bit systems the HW.INF file is located in **C:\Program File (x86)\Marvin Test Solutions\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 GXSW 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 GXSW driver.

Connectors and Jumpers

Figure 3-4 shows the GX6196 board connectors which provide connection to the user defined module board:

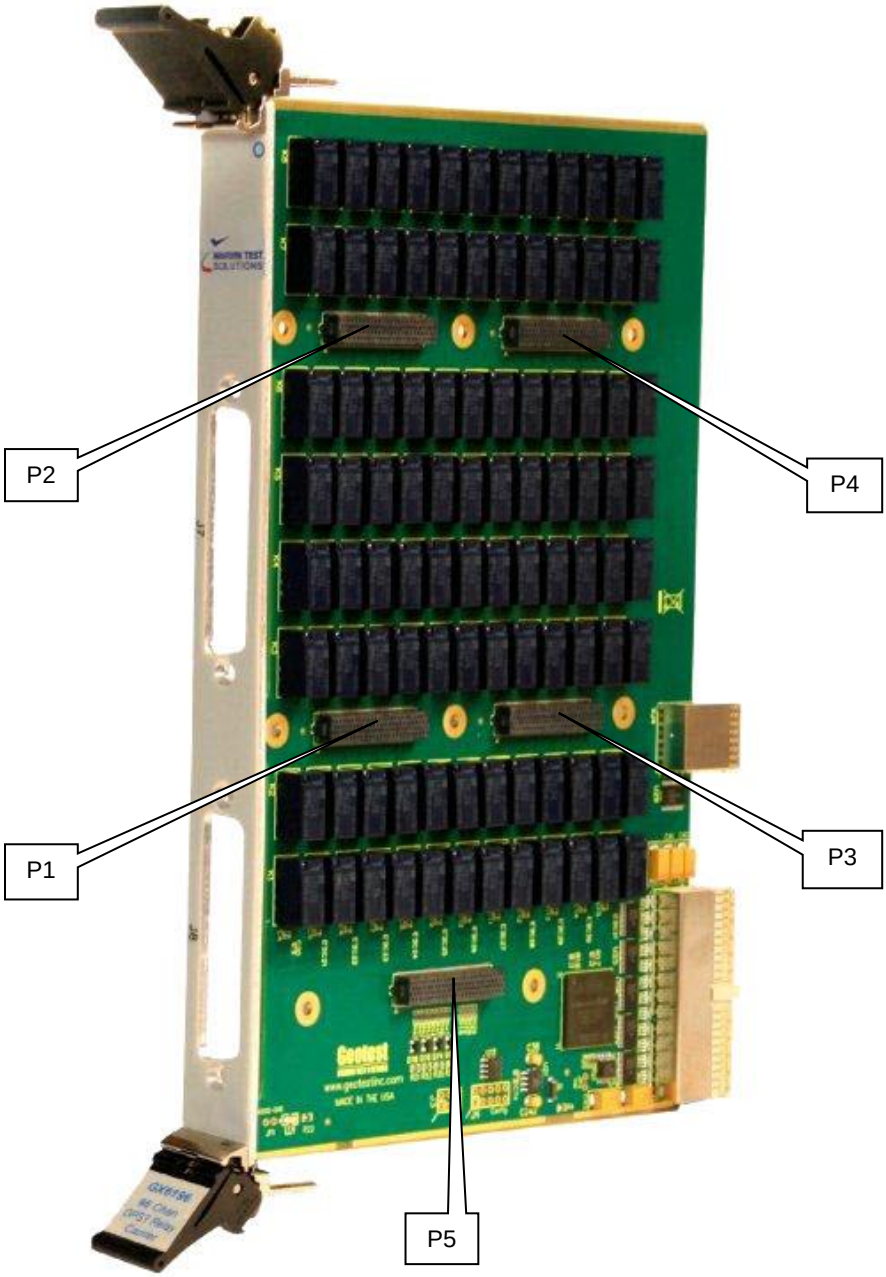


Figure 3-4: GX6196 Connectors and Jumpers

Module Board

Identification Lines

The ID[n] lines are for signals that are used for identification of the module board. Each line has a pull-up and each module card should tie a unique combination of these lines to ground. There are 15 combinations available. The value of 0xF is reserved for indicating no module card is installed.

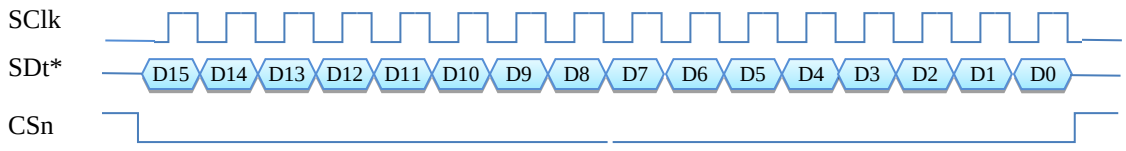
Digital I/O Lines

The DIO[n] lines are eight signals that are general purpose LVTTL discrete I/O. Each line can be programmed to drive high, drive low or hi-z. The status of each line can be read at any time.

Serial Link Interface

The serial control link is similar in functionality to the SPI interface bus. There are seven Sxxxx signals used for a serial communication link to the module or module board. SClk is the serial clock signal driven by the GX6196. SDOut carries the serial data from the module board to the GX6196. SDIn carries the serial data from the GX6196 to the module board. There are four SCSn[n] active low Chip Select signals which are used to select which device on the module board is communicating with the serial bus.

The GX6196 supports automatic serial transaction of 16 bits at a1MHz rate via the function call **Gx6196SendReceiveModule** . Figure 3-5 details the timing of the serial interface.



*SDt-Serial data. SDOut or SDIn.

Figure 3-5: Serial Interface Timing Diagram

Power

Power is drawn from the PXI backplane so power requirement of other cards may affect the maximum current available for the module board.

Voltage	Nominal Current	Maximum Current
+5V	1A	5A
+3.3V	2A	5A
+12V	0.5A	1A
-12V	0.25A	1A
+1.2V	0.5A	1A

Table 3-1: Power requirements from backplane

Dimensions

Figure 3-6 provides an outline drawing of the module board. For users designing their own module board, Marvin Test Solutions can supply an electronic file detailing this board's outline and critical dimensions:

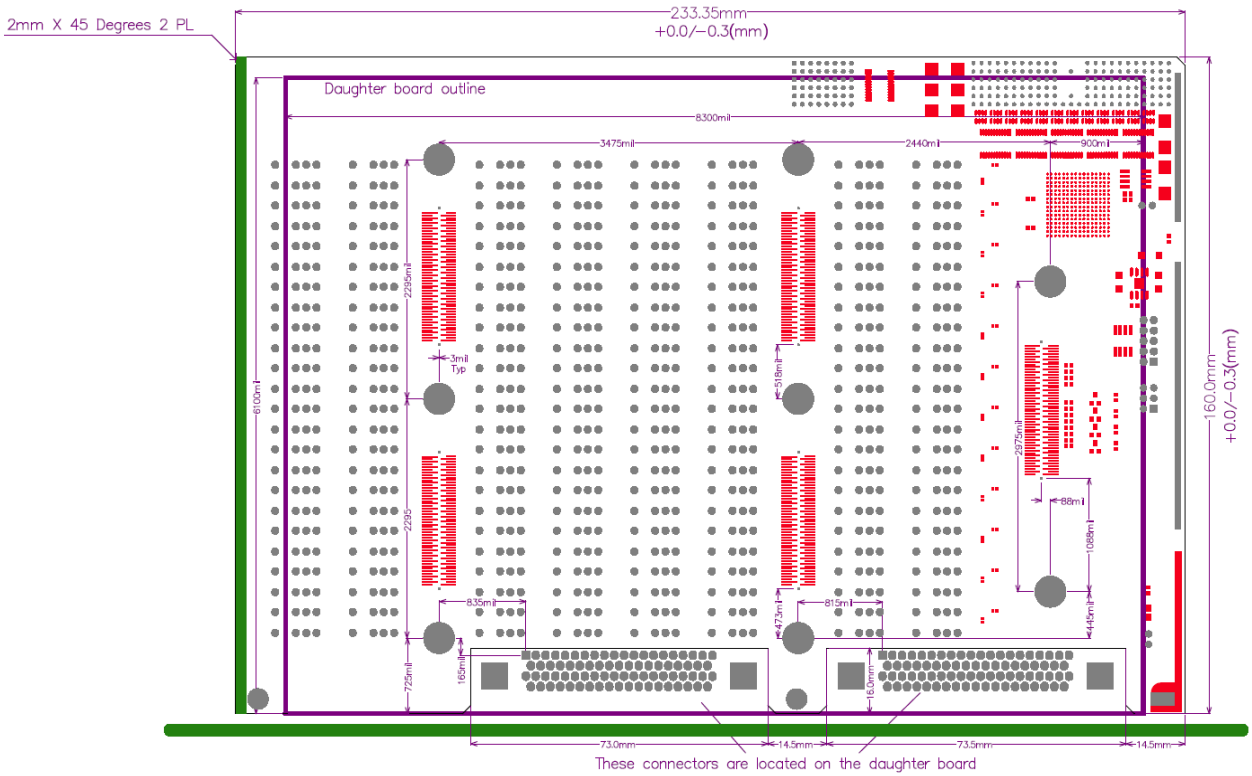


Figure 3-6: GX6196 Line Drawing of Board Layout

P1 - P5 Connectors – Module Connectors

Tables 3-2 to 3-5 list the pin assignments for connectors P1-P4. These connectors are connected to the relays contacts, providing the user with full access to all 96 relays and providing the means to create specific switch topologies. Additionally, by incorporating specific loads on the module board, a custom switched load board can be created. Each relay has two switches, A and B, and each switch has two pins. For example, K12 switch A (K12-A) connects between P1-C7 and P1-D8.

Row/Column	A	B	C	D
1	K1-A	GND	K3-A	GND
2	K1-B	K1-A	K3-B	K3-A
3	K2-A	K1-B	K4-A	K3-B
4	K2-B	K2-A	K4-B	K4-A
5	K9-A	K2-B	K11-A	K4-B
6	K9-B	K9-A	K11-B	K11-A
7	K10-A	K9-B	K12-A	K11-B
8	K10-B	K10-A	K12-B	K12-A
9	K17-A	K10-B	K19-A	K12-B
10	K17-B	K17-A	K19-B	K19-A
11	K18-A	K17-B	K20-A	K19-B
12	K18-B	K18-A	K20-B	K20-A
13	K25-A	K18-B	K27-A	K20-B
14	K25-B	K25-A	K27-B	K27-A
15	K26-A	K25-B	K28-A	K27-B
16	K26-B	K26-A	K28-B	K28-A
17	K33-A	K26-B	K35-A	K28-B
18	K33-B	K33-A	K35-B	K35-A
19	K34-A	K33-B	K36-A	K35-B
20	K34-B	K34-A	K36-B	K36-A
21	K41-A	K34-B	K43-A	K36-B
22	K41-B	K41-A	K43-B	K43-A
23	K42-A	K41-B	K44-A	K43-B
24	K42-B	K42-A	K44-B	K44-A
25	GND	K42-B	GND	K44-B

Table 3-2: P1 Pin Assignments

Row/Column	A	B	C	D
1	K5-A	GND	K7-A	GND
2	K5-B	K5-A	K7-B	K7-A
3	K6-A	K5-B	K8-A	K7-B
4	K6-B	K6-A	K8-B	K8-A
5	K13-A	K6-B	K15-A	K8-B
6	K13-B	K13-A	K15-B	K15-A
7	K14-A	K13-B	K16-A	K15-B
8	K14-B	K14-A	K16-B	K16-A
9	K21-A	K14-B	K23-A	K16-B
10	K21-B	K21-A	K23-B	K23-A
11	K22-A	K21-B	K24-A	K23-B
12	K22-B	K22-A	K24-B	K24-A
13	K29-A	K22-B	K31-A	K24-B
14	K29-B	K29-A	K31-B	K31-A
15	K30-A	K29-B	K32-A	K31-B
16	K30-B	K30-A	K32-B	K32-A
17	K37-A	K30-B	K39-A	K32-B
18	K37-B	K37-A	K39-B	K39-A
19	K38-A	K37-B	K40-A	K39-B
20	K38-B	K38-A	K40-B	K40-A
21	K45-A	K38-B	K47-A	K40-B
22	K45-B	K45-A	K47-B	K47-A
23	K46-A	K45-B	K48-A	K47-B
24	K46-B	K46-A	K48-B	K48-A
25	GND	K46-B	GND	K48-B

Table 3-3: P2 Pin Assignments

Row/Column	A	B	C	D
1	K49-A	GND	K51-A	GND
2	K49-B	K49-A	K51-B	K51-A
3	K50-A	K49-B	K52-A	K51-B
4	K50-B	K50-A	K52-B	K52-A
5	K57-A	K50-B	K59-A	K52-B
6	K57-B	K57-A	K59-B	K59-A
7	K58-A	K57-B	K60-A	K59-B
8	K58-B	K58-A	K60-B	K60-A
9	K65-A	K58-B	K67-A	K60-B
10	K65-B	K65-A	K67-B	K67-A
11	K66-A	K65-B	K68-A	K67-B
12	K66-B	K66-A	K68-B	K68-A
13	K73-A	K66-B	K75-A	K68-B
14	K73-B	K73-A	K75-B	K75-A
15	K74-A	K73-B	K76-A	K75-B
16	K74-B	K74-A	K76-B	K76-A
17	K81-A	K74-B	K83-A	K76-B
18	K81-B	K81-A	K83-B	K83-A
19	K82-A	K81-B	K84-A	K83-B
20	K82-B	K82-A	K84-B	K84-A
21	K89-A	K82-B	K91-A	K84-B
22	K89-B	K89-A	K91-B	K91-A
23	K90-A	K89-B	K92-A	K91-B
24	K90-B	K90-A	K92-B	K92-A
25	GND	K90-B	GND	K92-B

Table 3-4: P3 Pin Assignments

Row/Column	A	B	C	D
1	K53-A	GND	K55-A	GND
2	K53-B	K53-A	K55-B	K55-A
3	K54-A	K53-B	K56-A	K55-B
4	K54-B	K54-A	K56-B	K56-A
5	K61-A	K54-B	K63-A	K56-B
6	K61-B	K61-A	K63-B	K63-A
7	K62-A	K61-B	K64-A	K63-B
8	K62-B	K62-A	K64-B	K64-A
9	K69-A	K62-B	K71-A	K64-B
10	K69-B	K69-A	K71-B	K71-A
11	K70-A	K69-B	K72-A	K71-B
12	K70-B	K70-A	K72-B	K72-A
13	K77-A	K70-B	K79-A	K72-B
14	K77-B	K77-A	K79-B	K79-A
15	K78-A	K77-B	K80-A	K79-B
16	K78-B	K78-A	K80-B	K80-A
17	K85-A	K78-B	K87-A	K80-B
18	K85-B	K85-A	K87-B	K87-A
19	K86-A	K85-B	K88-A	K87-B
20	K86-B	K86-A	K88-B	K88-A
21	K93-A	K86-B	K95-A	K88-B
22	K93-B	K93-A	K95-B	K95-A
23	K94-A	K93-B	K96-A	K95-B
24	K94-B	K94-A	K96-B	K96-A
25	GND	K94-B	GND	K96-B

Table 3-5: P4 Pin Assignments

Table 3-6 lists the pin assignment of P5. This connector is used for providing power and control to the module board. The IDn lines are for signals that are used for identification of the module board. The DIOn lines are eight digital signals which provide that are general purpose discrete I/O control. The four Sxxxx signals are used for a serial communication link to the module board.

Row/Column	A	B	C	D
1	GND	GND	GND	GND
2				
3	+5V	+5V		
4	+5V	+5V		
5				
6				
7		DIO0		
8		DIO1		
9		DIO2	+3.3V	+3.3V
10		DIO3	+3.3V	+3.3V
11		DIO4		
12		DIO5		
13		DIO6		
14		DIO7		
15				
16				
17				
18	ID0			
19	ID1		SClk	
20	ID2		SCs	
21	ID3		SDIn	
22			SDOut	
23		-12V		
24		+12V		+1.2V
25	GND	GND	GND	GND

Table 3-6: P5 Pin Assignments

Chapter 4 - Functions Reference

Introduction

The GX6196 driver functions reference chapter is organized in alphabetical order. Each function is presented starting with the syntax of the function, a short description of the function parameters description and type followed by a Comments, an Example (written in C), and a See Also sections.

All function parameters follow the same rules:

- Strings are ASCIIZ (null or zero character terminated).
- Most function's first parameter is *nHandle* (16-bit integer). This parameter is required for operating the board and is returned by the **Gx6196Initialize** or the **Gx6196InitializeVisa** functions. The *nHandle* is used to identify the board when calling a function for programming and controlling the operation of that board.
- All functions return a status with the last parameter named *pnStatus*. The *pnStatus* is zero if the function was successful, or less than a zero on error. The description of the error is available using the **GxSWGetErrorString** function or by using a predefined constant, defined in the driver interface files: GXSW.H, GXSW.BAS, GXSW.PAS or GX6196.DRV.
- Parameter name are prefixed as follows:

Prefix	Type	Example
a	Array, prefix this before the simple type.	*anArray* (Array of Short)
n	Short (signed 16-bit)	*nMode*
d	Double - 8 bytes floating point	*dReading*
dw	Double word (unsigned 32-bit)	*dwTimeout*
l	Long (signed 32-bit)	*lBits*
p	Pointer. Usually used to return a value. Prefix this before the simple type.	*pnStatus*
sz	Null (zero value character) terminated string	*szMsg*
w	Unsigned short (unsigned 16-bit)	*wParam*
hwnd	Window handle (32-bit integer).	*hwndPanel*

Table 4-1: Parameter Prefixes

GX6196 Functions

The following list is a summary of functions available for the GX6196:

Driver Functions	Description
Gx6196Close	Closes a relay specified a channel number
Gx6196GetBoardSummary	Returns the board summary.
Gx6196GetChannel	Return the state of the specified by a channel number.
Gx6196GetPio	Returns PIO output states
Gx6196GetPioOutputEnable	Returns the PIO channel output enable states
Gx6196Initialize	Initializes the driver for the board at the specified slot number. The function returns a handle that can be used with other GX6196 functions to program the board
Gx6196InitializeVisa	Initializes the driver for the specified slot using VISA. The function returns a handle that can be used with other GX6196 functions to program the board.
Gx6196Open	Opens a relay specified by channel number.
Gx6196Panel	Opens a virtual panel used to interactively control the GX6196 board.
Gx6196ReadPio	Read input levels of the 8 PIO channels.
Gx6196Reset	Opens all the board relays.
Gx6196SendReceiveModule	Communicates serially with onboard Module (output and input)
Gx6196SetPioOutputEnable	Sets the PIO channel output enable states
Gx6196WritePio	Write output levels of the 8 PIO channels.
GxSWGetErrorString	Returns the error string associated with the specified error number.

Gx6196Close

Purpose

Closes a relay specified by a channel number.

Syntax

Gx6196Close (*nHandle, nChannel, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
nChannel	SHORT	Specified Relay to close (0-95).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Each relay (K1to K96) can be closed using this function. Note that the relays marked K1 to K96 are addressed as Channel 0 to 95.

Example

The following example closes relay 95:

```
Gx6196Close(nHandle, 95, &nStatus);
```

See Also

Gx6196GetChannel, Gx6196Open, GxSWGetErrorString

Gx6196GetBoardSummary

Purpose

Returns the board summary.

Syntax

Gx6196GetBoardSummary (*nHandle, szSummary, nSumMaxLen, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
szSummary	PSTR	Buffer to contain the returned board info (null terminated) string.
nSumMaxLen	SHORT	Size of the buffer to contain the board info string.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The GX6196 summary string provides the following data from in the order shown:

- Instrument Name (e.g., GX6196)
- FPGA version (e.g. 0xA002)
- Serial Number (e.g. 61960210)

Example

For example, the returned string looks like the following:

```
char sz[512];

Gt6196GetBoardSummary(nHandle, sz, 512, &nStatus);
```

sz will return:

```
“GX6196", "Carrier Switch, FPGA-Version:0xA003, S/N 61961234”
```

See Also

GxSWGetDriverSummary, GxSWGetErrorString

Gx6196GetChannel

Purpose

Return the state of a relay.

Syntax

Gx6196GetChannel (*nHandle, nChannel, pnState, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
nChannel	SHORT	Specified Relay to get state (0-95).
pnState	PSHORT	Returned state of the relay. 1 for close and 0 for open.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Returns the state of a relay (K1 to K96). Note that the relays marked K1 to K96 are addressed as Channel 0 to 95.

Example

The following example checks whether relay 95 is closed:

```
Gx6196GetChannel(nHandle, 95, &nState, &nStatus);
if (nState==1)
    printf("Relay 95 is closed");
else
    printf("Relay 95 is open");
```

See Also

Gx6196Close, Gx6196Open, GxSWGetErrorString

Gx6196GetPio

Purpose

Returns PIO output register data.

Syntax

Gx6196GetPio (*nHandle, pucData, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
pucData	PBYTE	Returned 8 bits (0-255), each corresponds to a PIO channel. A bit that is set to 1 indicates that the PIO channel is set to output a high level. A bit that is set to 0 indicates that the PIO channel is set to output a low level. Low bit (0x1) indicates that channel 0 is set to ouput a high level and high bit 0x80 indicates that channel 7 is set to output a high level.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

This function returns data from the output register (set by **Gx6196WritePio**).

Example

The following example reads back the PIO channels output register state:

```
BYTE ucData;

Gx6196WritePio(nHandle, 0xCD, &nStatus);
Gx6196GetPio(nHandle, &ucData, &nStatus);
if(ucData==0xCD)
   printf("Output register is reading back correctly");
```

See Also

Gx6196WritePio, Gx6196ReadPio, GxSWGetErrorString

Gx6196GetPioOutputEnable

Purpose

Gets the PIO channels output enable states.

Syntax

Gx6196GetPioOutputEnable (*nHandle, pucOutputEnable, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
pucOutputEnable	PBYTE	Returned 8 bits (0-255), each corresponds to a PIO channel. A bit that is set to 1 indicates that output is enabled. A bit that is set to 0 indicates that the output is disabled. Low bit (0x1) indicates that channel 0 is enabled and high bit 0x80 indicates that channel 7 output is enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Each of the 8 PIO channels has an output enable setting. A PIO channel will only be able to drive a voltage level when the channel's output enable is set.

Example

The following example gets all PIO output states:

```
BYTE ucOutputEnable;

Gx6196GetPioOutputEnable (nHandle, &ucOutputEnable, &nStatus);
if (ucOutputEnable & 0x2)
    printf("Channel 1 output is enabled");
```

See Also

Gx6196SetPioOutputEnable, Gx6196WritePio, Gx6196ReadPio, GxSWGetErrorString

Gx6196Initialize

Purpose

Initializes the driver for the board at the specified slot number. The function returns a handle that can be used with other GX6196 functions to program the board.

Syntax

Gx6196Initialize (*nSlot, pnHandle, pnStatus*)

Parameters

Name	Type	Comments
nSlot	SHORT	GX6196 board slot number on the PXI bus.
pnHandle	PSHORT	Returned handle for the board. The handle is set to zero on error and <> 0 on success.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The **Gx6196Initialize** function verifies whether or not the GX6196 board exists in the specified PXI slot. The function does not change any of the board settings. The function uses the HW driver to access and program the board.

The Marvin Test Solutions 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).

The returned handle *pnHandle* is used to identify the specified board with other GX6196 functions.

Example

The following example initializes two GX6196 boards at slot 1 and 2.

```
SHORT nHandle1, nHandle2, nStatus;
Gx6196Initilize (1, &nHandle1, &nStatus);
Gx6196Initilize (2, &nHandle2, &nStatus);
if (nHandle1==0 || nHandle2==0)
    {printf("Unable to Initialize the board")
    return;
}
```

See Also

Gx6196Reset, **GxSWGetErrorString**

Gx6196InitializeVisa

Purpose

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

Syntax

Gx6196InitializeVisa (*szVisaResource, pnHandle, pnStatus*)

Parameters

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

Comments

The **Gx6196InitializeVisa** 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 Marvin Test Solutions PXI/PCI Explorer as shown in the prior figure. The VISA resource string can be specified in several ways as follows:

- Using chassis, slot, for example: “PXI0::CHASSIS1::SLOT5”
- Using the PCI Bus/Device combination, for example: “PXI9::13::INSTR” (bus 9, device 9).
- Using alias, for example: “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 GX6196 boards at PXI bus 5 and device 11.

```
SHORT nHandle, nStatus;
Gx6196InitializeVisa (“PXI5::11::INSTR”, &nHandle, &nStatus);
if (nHandle==0)
{
    printf("Unable to Initialize the board")
    return;
}
```

See Also

Gx6196Initialize, Gx6196Reset, **GxSWGetErrorString**

Gx6196Open

Purpose

Opens a relay specified by a channel number.

Syntax

Gx6196Open (*nHandle*, *nChannel*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
nChannel	SHORT	Specified Relay to open (0-95).
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Each relay (K1to K96) can be opened using this function. Note that the relays marked K1 to K96 are addressed as Channel 0 to 95.

Example

The following example opens relay 95:

```
Gx6196Open(nHandle, 95, &nStatus);
```

See Also

Gx6196GetChannel, Gx6196Close, GxSWGetErrorString

Gx6196Panel

Purpose

Opens a virtual panel used to interactively control the GX6196 board.

Syntax

Gx6196Panel (*pnHandle, hwndParent, nMode, phwndPanel, pnStatus*)

Parameters

Name	Type	Comments
pnHandle	PSHORT	Handle to a GX6196 board. This number may be zero if the board is to be initialized by the panel window.
hwndParent	DWORD	Sets the panel parent window handle. A value of 0 sets the desktop as the parent window.
nMode	SHORT	The mode in which the panel main window is created. 0 for modeless and 1 for modal window.
phwndPanel	PDWORD	Returned window handle for the panel (for modeless panel only).
pnStatus	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 user closed the window. 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 GX6196 functions.

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

Example

The following example opens the panel in modal mode:

```
DWORD dwPanel;
SHORT nHandle=0, nStatus;
...
Gx6196Panel (&nHandle, 0, 1, &dwPanel, &nStatus);
```

See Also

Gx6196Initialize, **GxSWGetErrorString**

Gx6196ReadPio

Purpose

Read from PIO channels.

Syntax

Gx6196ReadPio (*nHandle, pucData, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
pucData	PBYTE	Returned 8 bits (0-255), each corresponds to a PIO channel. If a bit is set to 1, it indicates that the input to the PIO channel is high. If a bit is set to 0, it indicates that the input to the PIO channel is low.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

A channel's output must be enabled before it can drive a logic level. This function reads back the actual logic levels present on the PIO pins.

Example

The following example reads back the PIO channels input logic levels:

```
BYTE ucPioReadBack;
Gx6196ReadPio(nHandle, &ucPioReadBack, &nStatus);
if (ucPioReadBack & 0x2)
    printf("Channel 1 output is high");
```

See Also

Gx6196ReadPio, Gx6196GetPio, GxSWGetErrorString

Gx6196Reset

Purpose

Resets board.

Syntax

Gx6196Reset (*nHandle*, *pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

The function is typically called after calling the **Gx6196Initialize** function.

This function opens all relays, sets all PIO outputs to 0 and disables all PIO outputs.

Example

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

```
Gx6196Initilize (1, &nHandle, &nStatus);
if (nStatus < 0)
    return nStatus;// return error
Gx6196Reset (nHandle, &nStatus);
```

See Also

Gx6196Initialize, **GxSWGetErrorString**

Gx6196SendReceiveModule

Purpose

Communicates serially with the onboard Module by sending and receiving a word

Syntax

Gx6196SendReceiveModule (*nHandle, nChipSelect, awDataOut, pawDataIn, lSize, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
nChipSelect	SHORT	Specified chip select to use for communication to onboard Module (0-3)
awDataOut	PWORD	Array of words to send to Module
pawDataIn	PWORD	Array of words to receive from Module
lSize	LONG	Number of elements in awDataOut and pawDataIn arrays
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Each communication with the Module will result in one or more words being sent and then received. When each word is sent to the Module, an accompanying word is read back from the Module.

Example

The following example sends and receives 2 words from the Module using Chip Select 3:

WORD awDataOut[2];

WORD awDataIn[2];

awDataOut[0]=0x1234;

awDataOut[1]=0x5678;

```
Gx6196SendReceiveModule (nHandle, 3, awDataOut, awDataIn, 2, &nStatus);

if (awDataIn[0]==0x1234 && adwDataIn[1]=0x5678)
   printf("The two words were sent back correctly");
```

See Also

GxSWGetErrorString

Gx6196SetPioOutputEnable

Purpose

Sets the PIO channels output enable states.

Syntax

Gx6196SetPioOutputEnable (*nHandle, ucOutputEnable, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
ucOutputEnable	BYTE	8 bits (0-255), each corresponds to a PIO channel. A bit that is set to 1 indicates that output is enabled. A bit that is set to 0 indicates that the output is disabled. Low bit (0x1) indicates that channel 0 is enabled and high bit 0x80 indicates that channel 7 output is enabled.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

Each of the 8 PIO channels has an output enable setting. A PIO channel will only be able to drive a voltage level when the channel's output enable is set.

Example

The following example sets all PIO output enables and then writes 0x9 to the PIO, enabling PIO channel 0 and 3 and then setting these channels to high:

```
Gx6196SetPioOutputEnable (nHandle, 0x09, &nStatus);
Gx6196WritePio (nHandle, 0x9, &nStatus);
```

See Also

Gx6196GetPioOuputEnable, Gx6196WritePio, Gx6196ReadPio, GxSWGetErrorString

Gx6196WritePio

Purpose

Write data to the PIO output register.

Syntax

Gx6196WritePio (*nHandle, ucData, pnStatus*)

Parameters

Name	Type	Comments
nHandle	SHORT	Handle to a GX6196 board.
ucData	BYTE	8 bits (0-255), each corresponds to a PIO channel. A bit that is set to 1 indicates that the PIO channel is set to output a high level. A bit that is set to 0 indicates that the PIO channel is set to output a low level. Low bit (0x1) indicates that channel 0 is set to output a high level and high bit 0x80 indicates that channel 7 is set to output a high level.
pnStatus	PSHORT	Returned status: 0 on success, negative number on failure.

Comments

A channel's output must be enabled before it can drive a logic level.

Example

The following example enables the output on all PIO channels and then set PIO channels 0-2 to high and set channel 3-7 to low:

```
Gx6196SetPiotOutputEnable(Handle, 0xFF, &nStatus);
Gx6196WritePio(nHandle, 0x7, &nStatus);
```

See Also

Gx6196ReadPio, Gx6196GetPio, GxSWGetErrorString

GxSWGetErrorString

Purpose

Returns the error string associated with the specified error number.

Syntax

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

Parameters

Name	Type	Comments
nError	SHORT	Error number as returned by the *pnStatus* of any GXSW function. See table below for possible error numbers values. The error number should be a negative number, otherwise the function returns the "No error has occurred" string.
pszMsg	LPSTR	Buffer containing the returned error string (null terminated string).
nErrorMaxLen	SHORT	Size of the buffer *pszMsg*.
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.

This function returns error value or 0 on success.

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

Resource Errors

-1 Board does not exist in this slot

-2 Unable to open the HW device/Service

-3 Different board exist in the specified PCI slot

-4 PCI slot not configured properly. You may configure it by using the **PCIExplorer** from the Control Panel

-5 Unable to register the PCI device

-6 Unable to allocate system resource or memory for the PCI device

-7 Too many boards

-8 Unable to create panel

-9 Unable to create a Windows timer

Parameter Errors

-20 Invalid parameter

-21 Invalid PCI slot number

-22 Invalid board handle

-23 Invalid channel number

-24 Invalid bus number

-25 Invalid mode

-26 Invalid group number

-27 Invalid string length

-28 Invalid row number

-29 Invalid column number

Board Errors/Warnings

-50 BIT error: Adapter not connected

-51 BIT error: Comparator Error

-52 BIT error: Unable to open/close a relay in Group x, Row y Column z

-53 VI in Error

-54 Memory mapping error

Miscellaneous Errors

-99 Invalid or unknown error number

Example

The following example initializes the board at slot 3. If the initialization failed the following error string is printed:

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

GX6196Initialize(3, &Handle, &Status);
if (nStatus<0)
{   GxSWGetErrorString(nStatus, sz, sizeof sz, &nStatus);
    printf(sz);// print the error string return;
}
```

Index

V