# *GXSW*
# Switching Instruments

# GX7016
## GENASYS Switching Subsystem
## *User's Guide*

Last Updated: February 26, 2015

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 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

Marvin Test Solutions products are warranted against defects in materials and workmanship for a period of 12 months. 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 questions 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 8:30 AM and 5:30 PM Pacific Standard Time.

Our address (check our website for latest address): Marvin Test Solutions, 1770 Kettering, Irvine, CA 92614, USA.

## 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 Inc. |
| C++ Builder, Delphi | Embarcadero Technologies Inc. |
| LabView, LabWindows™/CVI | National Instruments |
| Microsoft Developer Studio, Microsoft Visual C++, Microsoft Visual Basic, .NET, Windows 95, 98, NT, ME, 2000, XP, VISTA, Windows 7 and 8 | Microsoft Corporation |

All other trademarks are the property of their respective owners.

# Table of Contents

Safety and Handling....1
Warranty....1
If You Need Help....1
Disclaimer....1
Copyright....1
Trademarks....2
**Chapter 1 - Introduction....1**
Manual Scope and Organization....1
Manual Scope....1
Manual Organization....1
Conventions Used in this Manual....1
**Chapter 2 - Overview....3**
Introduction....3
Description....3
Features....3
Configuration....4
Switching Subsystem Components....4
GX7016 – PXI Chassis....5
GX6256 - Low Frequency Multiplexer PXI (LF)....6
GX6192 - High Frequency Multiplexer PXI Board (HF)....7
GX6032 - Rear Transition Module (RTM)....8
GX6009 – Switching Router Module (SR)....9
Architecture....10
GX7016....10
GX6256 (LF)....11
GX6192 (HF)....12
GX6032 (RTM)....14
GX6009 (SR)....15
Specifications....16
Virtual Panel....18
Virtual Panel - LF Slot Page....19
Virtual Panel - HF Slot Page....20
Virtual Panel - Rear Transition Module (RTM) Page....22
Virtual Panel - Switch Router (SR) Page....23
**Chapter 3 - Installation and Connections....27**

Getting Started ..... 27
Packing List ..... 27
Unpacking and Inspection ..... 27
System Requirements ..... 27
Installation of the GXSW Software ..... 28
Setup Maintenance Program ..... 28
Overview of the GXSW Software ..... 29
Installation Folders ..... 29
Configuring Your PXI System using the PXI/PCI Explorer ..... 30
Board Installation ..... 31
Before you Begin ..... 31
Electric Static Discharge (ESD) Precautions ..... 31
Installing a PXI Board ..... 31
Plug & Play Driver Installation ..... 33
Removing a Board ..... 33
Connectors and Jumpers ..... 34
GX6256 ..... 34
GX6192 ..... 35
Connectors and Accessories ..... 36
GX6256 ..... 36
GX6192 ..... 36
GX6256 J2 Connector ..... 37
GX6256 J3 Connector ..... 39
GX6256 J7 Connector ..... 41
GX6192 J2 Connector ..... 42
GX6192 J3 Connector ..... 44
GX6192 P1 Connector ..... 46
GX6192 P2 Connector ..... 46
**Chapter 4 - SwitchEasy Software ..... 47**
SwitchEasy Overview ..... 47
SwitchEasy Panel ..... 48
Status Page ..... 48
Pin Maps Page ..... 50
Exclusions Page ..... 51
SwitchEasy Base Commands ..... 52
SwitchEasy Command Reference ..... 52

SwitchEasy Connect/Disconnect Commands ..... 55
SwitchEasy Commands Conventions ..... 56
Programming Examples ..... 56
Mapping pins and pin groups ..... 56
Making a connection ..... 56
Disconnecting ..... 57
**Chapter 5 - Function Reference ..... 59**
Introduction ..... 59
GX6256/GX6192/GX6009/GX6032 Functions ..... 60
Gx6256BITAbort ..... 62
Gx6256BITGetStatus ..... 63
Gx6256BITRun ..... 64
Gx6256BleedRelayClose ..... 65
Gx6256BleedRelayGetState ..... 66
Gx6256BleedRelayOpen ..... 68
Gx6256GetBoardSummary ..... 69
Gx6256GetBoardType ..... 70
Gx6256GetRelayCount ..... 71
Gx6256GetRelayStates ..... 72
Gx6256Initialize ..... 73
Gx6256InitializeVisa ..... 75
Gx6256MatrixClose ..... 76
Gx6256MatrixGetChannel ..... 78
Gx6256MatrixOpen ..... 80
Gx6256MatrixInputSelectorClose ..... 82
Gx6256MatrixInputSelectorGetChannel ..... 83
Gx6256MatrixInputSelectorOpen ..... 84
Gx6256MatrixReset ..... 85
Gx6256MuxClose ..... 86
Gx6256MuxGetChannel ..... 88
Gx6256MuxOpen ..... 90
Gx6256MuxReset ..... 92
Gx6256MuxGetSelectorMode ..... 93
Gx6256MuxSetSelectorMode ..... 95
Gx6256Panel ..... 97
Gx6256PioGetOutputData ..... 98

Gx6256PioGetOutputEnable ..... 99
Gx6256PioReadInput ..... 100
Gx6256PioSetOutputData ..... 101
Gx6256PioSetOutputEnable ..... 102
Gx6256Reset ..... 103
Gx6256RtmMatrixClose ..... 104
Gx6256RtmMatrixGetChannel ..... 105
Gx6256RtmMatrixOpen ..... 106
Gx6256SrBusClose ..... 107
Gx6256SrBusGetChannel ..... 108
Gx6256SrBusOpen ..... 109
GxSWGetErrorString ..... 110
**Index ..... 115**

# Chapter 1 - Introduction

## Manual Scope and Organization

### Manual Scope

This manual provides all the information necessary for installation, operation, and maintenance of the **GX7016** chassis, and the switching boards: **GX6256**, **GX6192**, **GX6009**, and **GX6032**. These components are part of the GX7016 digital subsystem that is part of the TS-323/GENASYS test system. The manual also covers the **GXSW** software package that includes the GX6256 and GX6192 driver. In addition to the GXSW software components, this manual also described the **SwitchEasy** software that is based on the GXSW driver and provides a rule-based, pin mapping and higher level access to the switching subsystem. SwitchEasy software is described later in this manual. 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

This manual is organized in the following manner:

| Chapter | Content |
|---|---|
| Chapter 1 – Introduction | Introduces the GX7016 switching subsystem manual and shows warning conventions used in the manual. |
| Chapter 2 – Overview | Provides the GX7016 switching subsystem 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 GX6256 and GX6192 board and the GXSW software. |
| Chapter 4 –SwitchEasy Software | Provides an overview of the SwitchEasy software driver and panel |
| Chapter 5 – Functions Reference | Provides a list of the GXSW driver functions for the GX6256, GX6192, GX6009, and GX6032. 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 Marvin Test Solutions GX7016 GENASYS switching subsystem provides the following features

6U PXI chassis with integrated MAC Panel SCOUT receiver

20 slot 6U PXI chassis supports GENASYS switching subsystem modules

Supports over 4000 multiplexed, hybrid pin connections

Supports up to 288 digital channels when integrated with a digital subsystem

## Description

A component of the GENASYS platform, the GX7016 switching subsystem offers a new level of high performance, high density switching for board and system level functional test. Based on the 6U PXI architecture, the GX7016 incorporates a modular switch matrix and multiplexer architecture which supports up to 4608 multiplexed, hybrid I/O pins. Up to 128 external resources can be connected to any of the test system's receiver I/O pins via a high performance, internal 16 wire matrix bus. For digital test capability, the GX7016's switching subsystem can be connected to a digital subsystem such as Marvin Test Solutions' GX5960 digital subsystem, providing hybrid pin capability for each multiplexed I/O pin. Up to 288 digital channels are supported by the switching subsystem.

GX7016 chassis incorporates the MAC Panel 6U SCOUT receiver. The SCOUT receiver offers a reliable and high performance method to connect the switch modules to a mass interconnect receiver, minimizing the need for cable assemblies. The SCOUT receiver is a "pull-through" design, with each switch card providing a "cable-less" connection to the receiver connectors, – eliminating the need for cable harnesses and the associated reliability issues that come with cabled solutions. The result is a system interconnect design that is cost effective, reliable, and maintainable. The modular design of the SCOUT also allows for the use of a broad range of receiver connectors including high density, high current, and coaxial types. The 6U SCOUT receiver can accommodate up to 21 connector slots and over 8000 connections when fully populated.

## Features

The GX7016 is a 20-slot 6U PXI chassis that can accommodate up to 19 switching or instrument cards as well as a remote PXI bus interface such as the MXI-4. In addition to supporting all of the PXI-1 resources, the GX7016's PXI backplane provides an internal, high performance, 16 wire, and analog bus via the backplane's P5 connectors. Each of the GENASYS switching cards connects to this internal 16 wire bus, providing the ability to route signals from an external instrument to any of the receiver's interface connections.

System power for the GX7016 is provided by a 755 watt power supply. Forced-air cooling for the chassis is provided by a four (4) 79 CFM fans mounted under the card cage—providing positive airflow per the PXI specification. This configuration provides the optimum cooling for the chassis regardless of the type or number of instruments or switch cards used. Additional cooling is provided for the system power supply. This cooling configuration, in conjunction with air plenums within the chassis, provides airflow for all module slots per the PXI specification and requires no additional rack space for inlet or outlet air.

The GX7016 chassis supports the monitoring of slot temperatures and system power supply voltages as well providing the ability to program or map each PXI trigger line from one PCI segment to another. In addition, the user can program the temperature monitoring function for specific warning and shutdown limits. All user specific setups can be stored in non-volatile memory as a user configuration and can be used as the default setup for normal chassis operation.

## Configuration

The GX7016's Slot 1 is dedicated for the remote system controller. A PXI Star Trigger controller, any PXI or cPCI instrument or a GENASYS switching module can be used in slot 2. Slots 3-15 support the PXI Star Trigger, any PXI or cPCI instrument or a GENASYS switching module. Slots 16-20 accommodate a PXI or cPCI instruments without the Star Trigger or a GENASYS switching module.

## Switching Subsystem Components

The modular architecture of the GX7016 allows the switching subsystem to support a wide range of configurations and capabilities. Input analog signal routing is supported by the GX6032, a matrix switch module which is located on the rear of the chassis. This module is configured as 32 x 16 matrix and supports up to 32 inputs which can be connected to the GX7016's internal 16 wire bus. Up to (4) of these modules can be supported by the system. The base system is supplied with (1) GX6032.

Signal routing from the internal backplane to the receiver interface is supported by the GX6256 MultiMatrix™ switch card which is an extended 6U PXI module with direct connection to the receiver interface. The module combines a 16x16 matrix with (16) 2:16 multiplexers. The result is a switch card that can support up to 256 multiplexed signals from the 16 wire bus. Additionally, a secondary bus is included, providing the ability to route (2) resources to each group of 16 I/O pins within a mux group. The GX6256 also supports the ability to select an analog or digital resource (GX5960 digital subsystem) for each of the 16 inputs/ outputs.

For higher performance analog switching applications, the GX7016 can be configured with the GX6192 HF MultiMatrix™ switch cards which combines a 16x16 matrix with (16) 1x12 multiplexers, providing a total of 192 high performance I/O pins at the receiver interface. HF resources are connected directly to the GX6192 via an external connector. In addition signals can be routed to / from the HF switch card to other switch cards in the system via the system's internal 16 wire bus.

The GX7016 switching subsystem includes the following components:

GX7016 – 6U PXI chassis with special backplane used for plugging in the SR, RTM in addition to the PXI boards LF and HF

GX6009 - Switching Routers modules (SR)

GX6032 - Rear Transition modules (RTM)

GX6192 - High Frequency multiplexer PXI boards (HF)

GX6256 - Low Frequency multiplexer PXI boards (LF).

In addition to the GX6256 and GX6192 switching function, they are also used control an RTM and an SR that are mounted to backplane since these card do not sit on the PXI bus.

## GX7016 – PXI Chassis

The GX7016 switching sub system includes a PXI chassis with a special backplane that includes Switching Routers (SR GX6009), Rear Transition Modules (RTM GX6032), GX6192 (HF) and GX6256s (LF). The GX6256 and GX6192 can control an RTM and an SR that are mounted to backplane.


**Figure 2-1: GX7016 Genasys Switching Chassis**

## GX6256 - Low Frequency Multiplexer PXI (LF)

The GX6256 Low Frequency multiplexer 6U PXI card has sixteen 2x16 switching groups connected to 256 channels tied to selectors. Each channel has a primary path and a secondary path. The primary path is designed for optimum performance while the secondary path is intended to be used for DC signal.

The board consists of a multiplexer, matrix and selectors. The GX6256 connects to 16 analog signals through the backplane and to 16 digital signals through a connector in the bottom. Input selectors select the analog or digital input for each multiplex group. Each group routs this signal to one of sixteen channels that are connected to the front connectors.

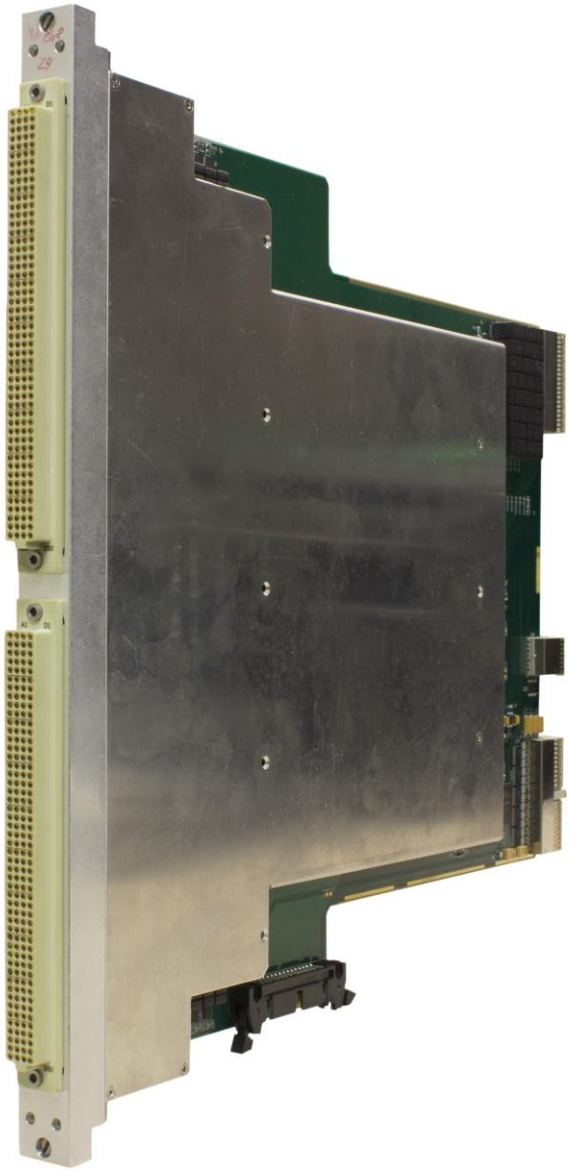

**Figure 2-2: GX6256 Low Frequency Mux and Matrix board**

## GX6192 - High Frequency Multiplexer PXI Board (HF)

The GX6192 is a High Frequency 6U PXI card derivative of the GX6256 with several changes in architecture. The GX6192 provides the user with the ability to connect and interface 16 resources to 192 UUT connections. The board consists of sixteen 1x12 multiplexer groups, eight 4x8 matrices, sixteen 2:2 selectors, and sixteen 1:4 input selectors. The GX6192 connects to 16 analog signals bus through a connector in the back. Each multiplexer group routes its signals to one of sixteen channels that are connected to the front connectors.

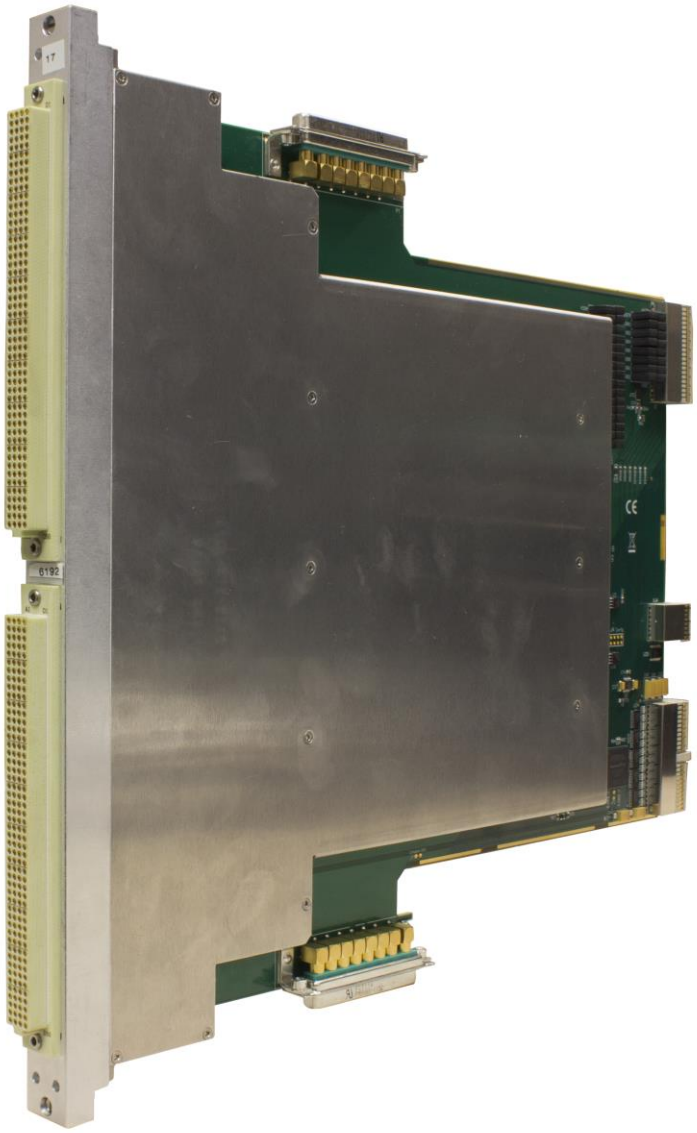

**Figure 2-3: GX6192 High Frequency Mux and Matrix board**

## GX6032 - Rear Transition Module (RTM)

The GX6032 Rear Transition Module (RTM) connects to the back side of the GX7016 backplane. The RTM is designed to connect up to 32 external instrument inputs to the backplane's 16 global bus lines via a 32x16 matrix. The RTM is controlled by an accompanying GX6256 or GX6192 board which is connected to the PXI slot adjacent to the RTM.

**Figure 2-4: GX6032 Rear Transition Module (RTM)**

## GX6009 – Switching Router Module (SR)

The GX6009 Switching Router module (SR) connects up to 9 switching cards (GX6256 or GX6192) together across the backplane via the global us lines. The GX7016 switching subsystem contains two GX6009 switching routers which can be connected together in order to connect the global bus across up to 18 slots.

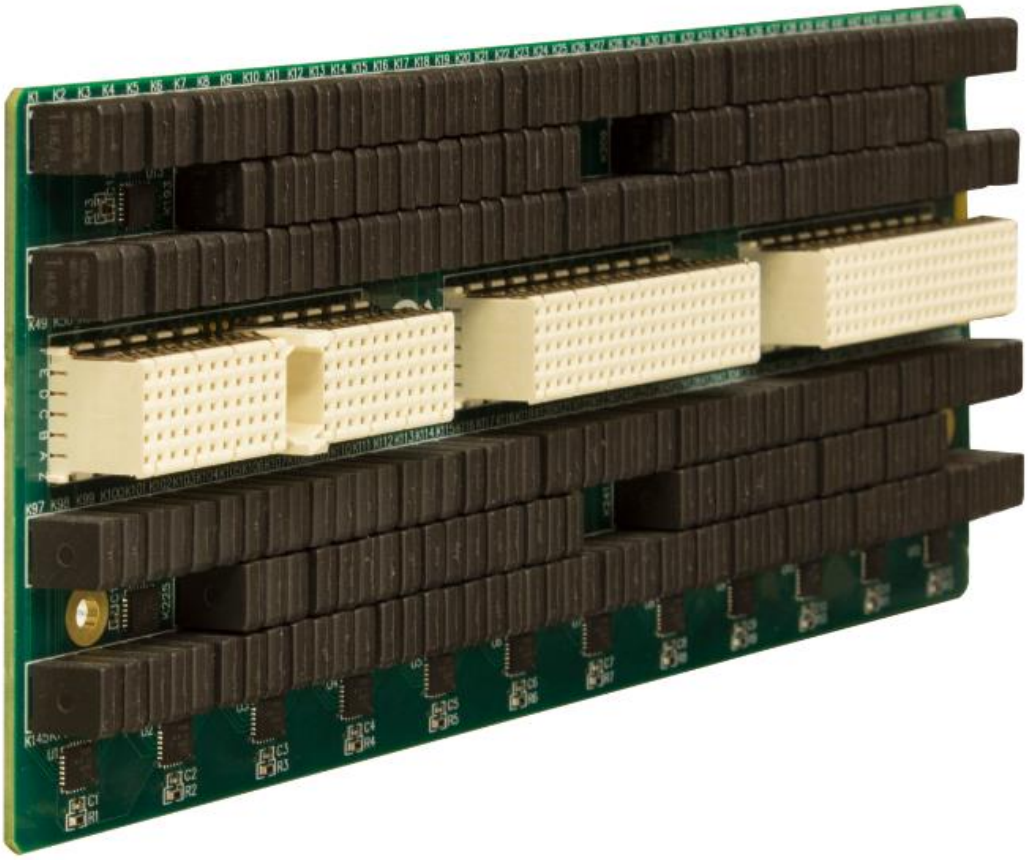

**Figure 2-5: GX6009 Switch Router (SR)**

## Architecture

### GX7016

The Genasys switching sub-system has four main components: backplane, MultiMatrix™ switching cards, Switching Routers (SRs), and Rear Transition Modules (RTMs). Figure 2-6 describes the block diagram of the Genasys switching sub-system. Each component is shown in a different color.

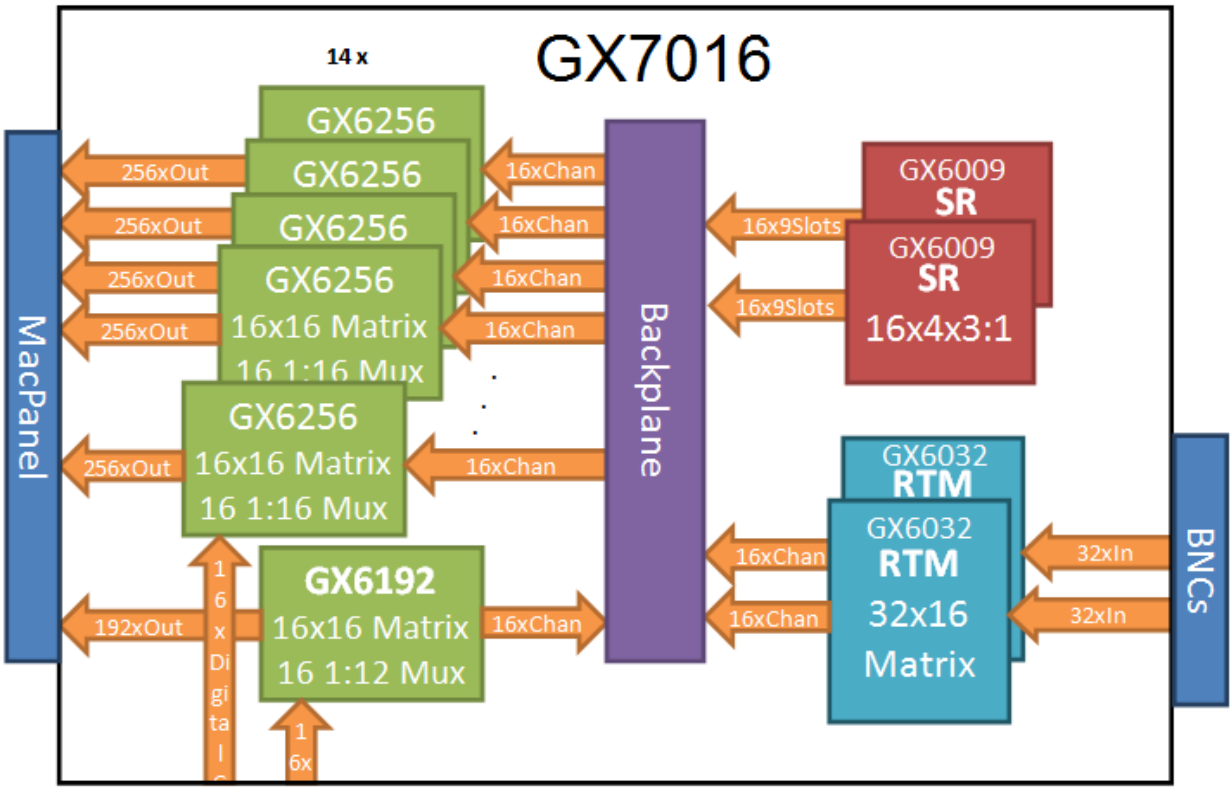


**Figure 2-6: GX7016 Genasys Switching Sub-System Block Diagram**

The MultiMatrix™ switching cards are used to connect the UUT to the system resources through a MacPanel interface. These cards combine multiplexer and matrix architectures to route high frequency (HF), low frequency (LF), and digital signals to and from the UUT. There are two types of cards. The GX6192 HF Switching and the GX6256 LF Switching. Up to 18 cards can be install in a system. Nominal configuration is one GX6192 and fourteen GX6256 cards.

The switching routers are routing the 16 between the different slots while keeping stubs length to a minimum. Each SR supports nine slots; 2-10 and 11-19.

The rear transition modules connect the LF system resources to the switching sub-system. Each RTM supports 32 input channels that it can route to any of the 16 global bus signals.

The backplane is a special PXI backplane with provision to support up to four RTMs, two SRs and up to 18 MultiMatrix™ switching cards.

### GX6256 (LF)

The GX6256 comprises of sixteen 1:4 input selectors, eight 4x8 matrices, sixteen 3:2 selectors, and sixteen 2x16 multiplexer groups connected to 256 channels. The switch matrix is essentially a 16x16 matrix implemented as eight 4x8 matrices to reduce stubs length. The row selector and input selector help in facilitating this architecture.

In order to reduce stubs length, the actual implementation divides the group to two sets of eight channels each. Channels 1-8 are the 'Lo' channels and 9-16 are the 'Hi' channels. Another feature implemented in the design is a Build-In Test circuit (BIT). By connecting a pull-up and a level sensor to one input of a pair, and a pull down to the other input, the activation of all the groups' relays can be verified.

Figure 2-7 describes the switching sections of the GX6256. Each section is shown in a different color.

There are eight 4x8 switch matrices. The matrices columns are routed to the input channels and the rows are routed to multiplexer groups via the selectors.

The input selector switches an Input channel from the global switching bus in the backplane to one of four groups of rows: A-D, E-H, I-L, M-P. Figure 4 includes a logic schematic of an input selector.

The row selector switches a pair of multiplexer groups to one of two sets of columns: 1-8 or 9-16 or to a digital input. Figure 2-8 includes a logic schematic of a row selector. The row selector can also connect the row to the BIT circuit.

The multiplexer has 16 switching groups, A-P. Each group has 16 output channels. In order to reduce stubs length, the actual implementation divides the group to two sets of eight channels each. Channels 1-8 are the 'Lo' channels and 9-16 are the 'Hi' channels.

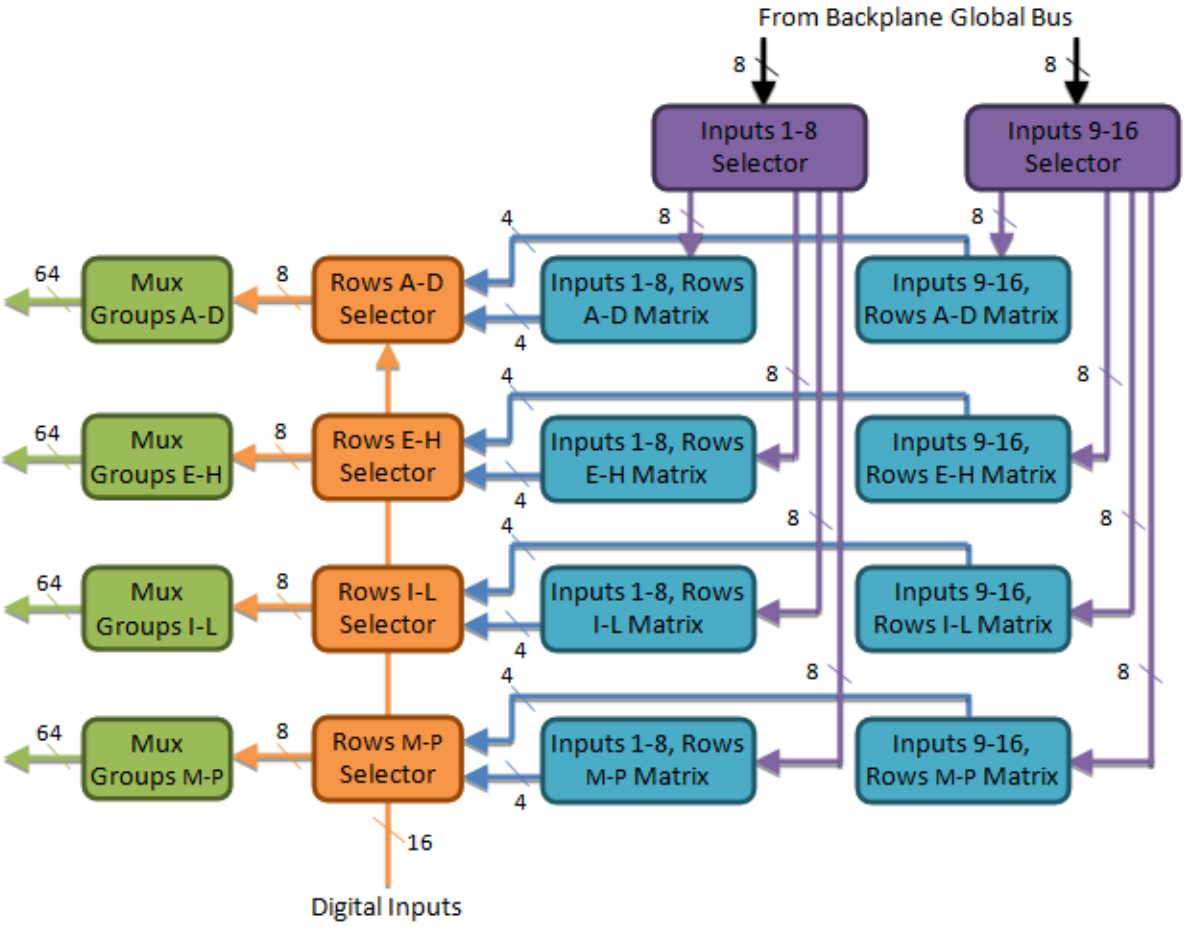

**Figure 2-7: Block Diagram of the Switching Sections of the GX6256**

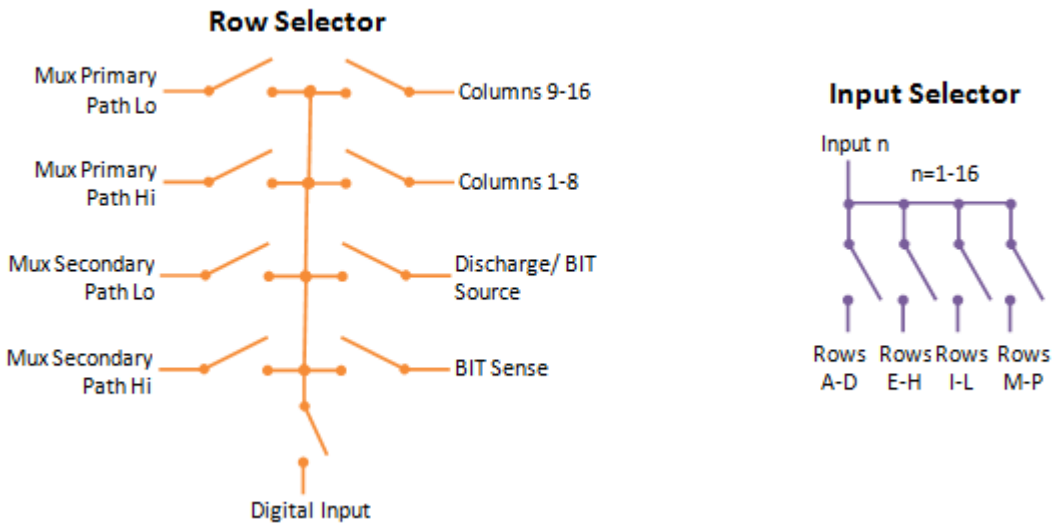


**Figure 2-8: Schematic Diagram for Selectors**

### GX6192 (HF)

The GX6192 comprises of sixteen 1:4 input selectors, eight 4x8 matrices, sixteen 2:2 selectors, and sixteen 1x12 multiplexer groups connected to 192 channels. The switch matrix is essentially a 16x16 matrix implemented as eight 4x8 matrices to reduce stubs length. The selector and input selector help in facilitating this architecture.

Figure 2-9 describes the switching sections of the GX6192. Each section is shown in a different color.

There are eight 4x8 switch matrices. The matrices columns are routed to the input channels and the rows are routed to multiplexer groups via the selectors.

The input selector switches an Input channel to one of four groups of rows: A-D, E-H, I-L, M-P. Figure 2-9 includes a logic schematic of an input selector.

The row selector switches a multiplexer group to one of two sets of columns: 1-8 or 9-16. Figure 2-10 includes a logic schematic of a row selector. The row selector can also connect the row to the corresponding global bus line in the backplane.

The multiplexer has 16 switching groups, A-P. Each group has 12 output channels. In order to reduce stubs length, the actual implementation divides the group to two sets of six channels each. Channels 1-6 are the 'Lo' channels and 7-12 are the 'Hi' channels.

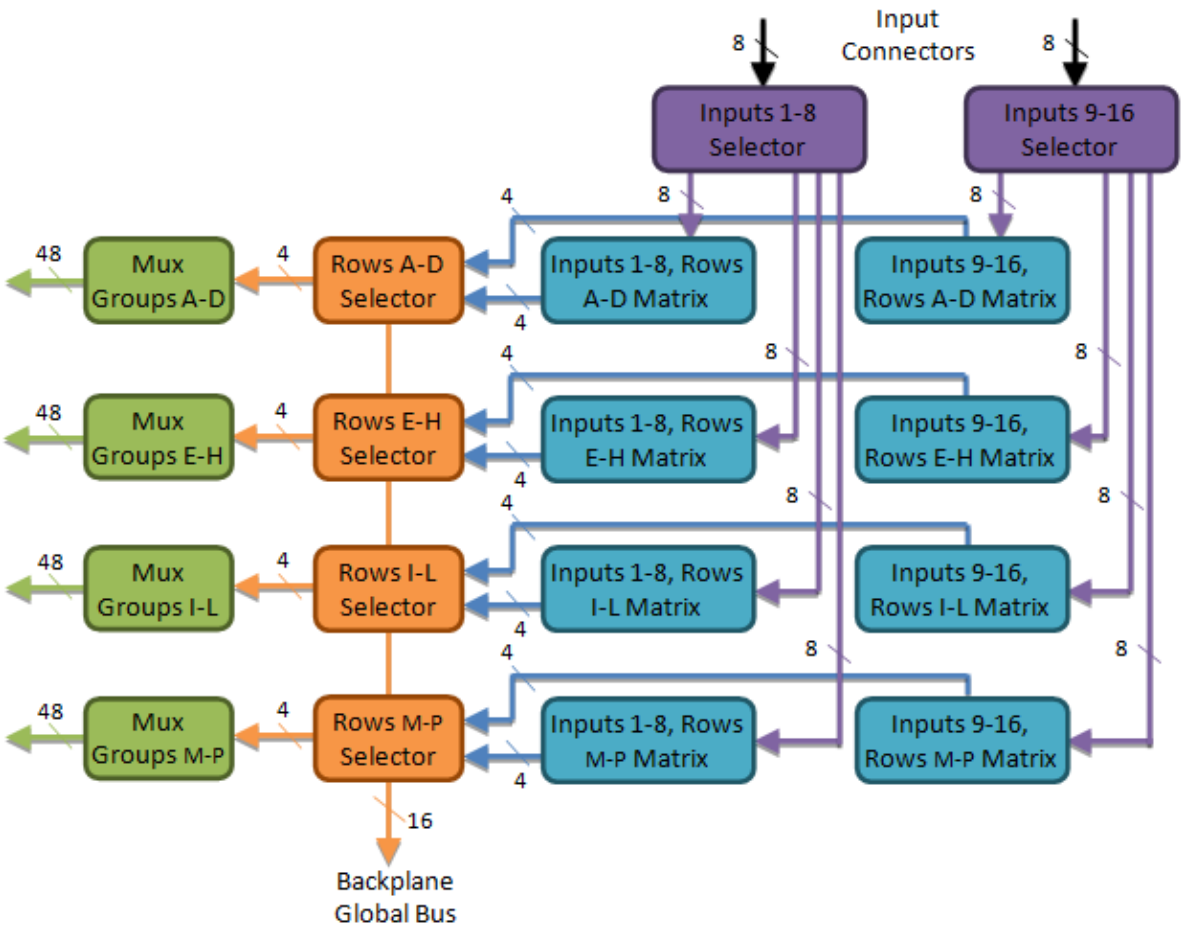


**Figure 2-9: Diagram of the Switching Sections of the GX6192**

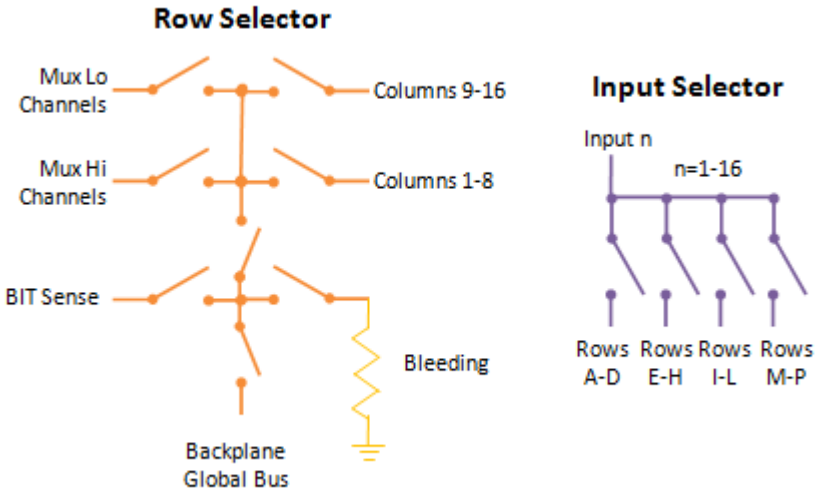


**Figure 2-10: Schematic Diagram for Selectors**

### GX6032 (RTM)

The RTM comprises of 32 1:2 input selectors, 16 4x8 matrices, and 16 1:8 selectors. The switch matrix is essentially a 32x16 matrix implemented as 16 4x8 matrices to reduce stubs length. The row selector and input selector help in facilitating this architecture. Figure 2-11 describes the switching sections of the GX6032. Each section is shown in a different color.

There are 16 4x8 switch matrices. The matrices columns are routed to the input channels and the rows are routed to the global bus via the selectors.

The input selector switches an Input channel from a BNC connector to one of two groups of columns: 1-8, 9-16. These numbers are related to the global bus signals. Figure 2-12 includes a logic schematic of an input selector.

The column selector switches a global bus signal to one of eight sets of rows: 1-4, 5-8, 9-12, 13-16, 17-20, 21-24, 25-28 or 29-32. These numbers are related to the input channels. Figure 2-12 includes a logic schematic of a row selector.

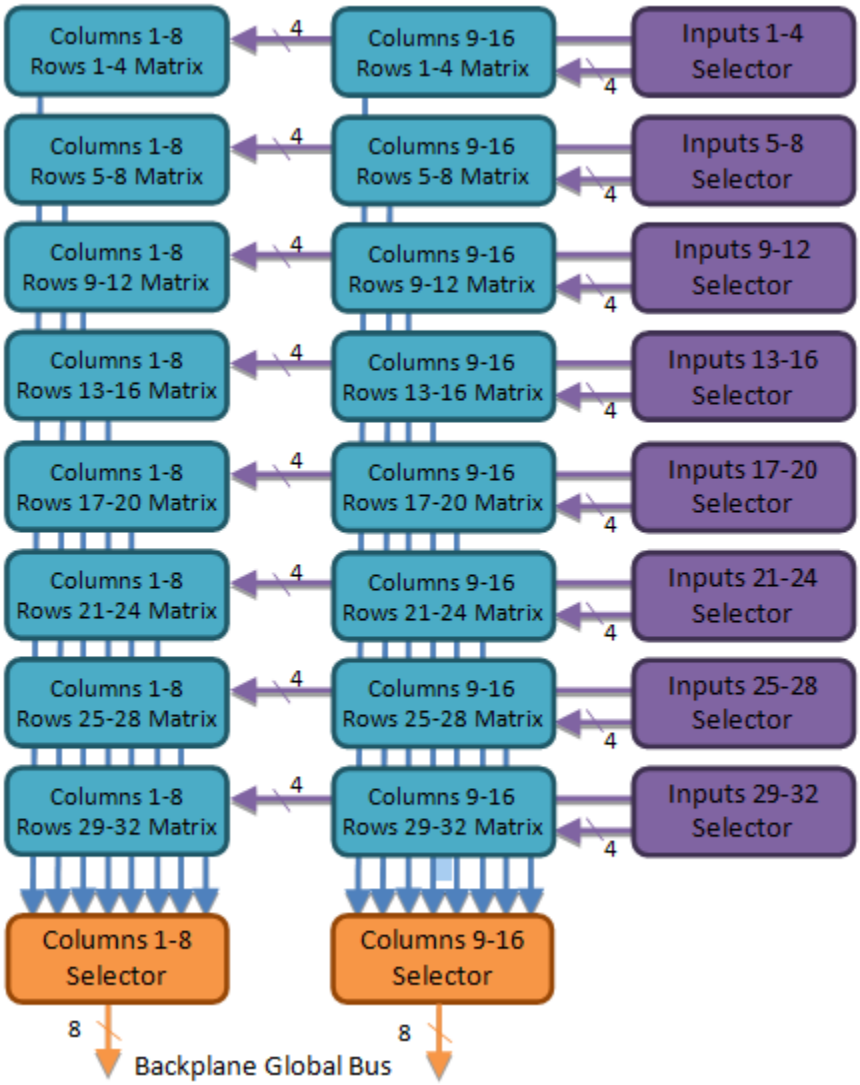


**Figure 2-11: Block Diagram of the Switching Sections of the RTM**

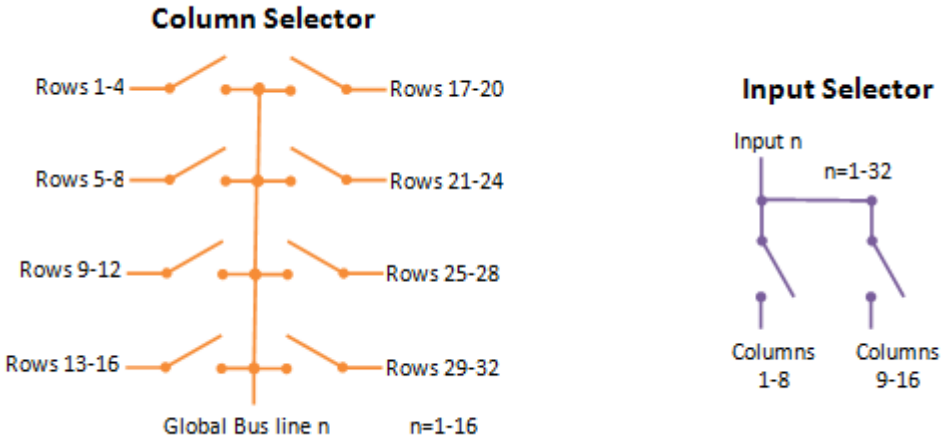


**Figure 2-12: Schematic Diagram for RTM Selectors**

### GX6009 (SR)

The Switching router routes the 16 lines of the global bus between the slots. Each one of the bus lines routing consists of four switching groups, with four relays each. Three groups are slot switching groups and one is the common switching group. Figure 2-13 shows a schematic of one bus line implementation in the SR. Three of the relays in a slot group are connected to slots and the fourth is connected to the common group. Three of the relays in the common group are connected to slot groups and the fourth is connected to the other SR board.

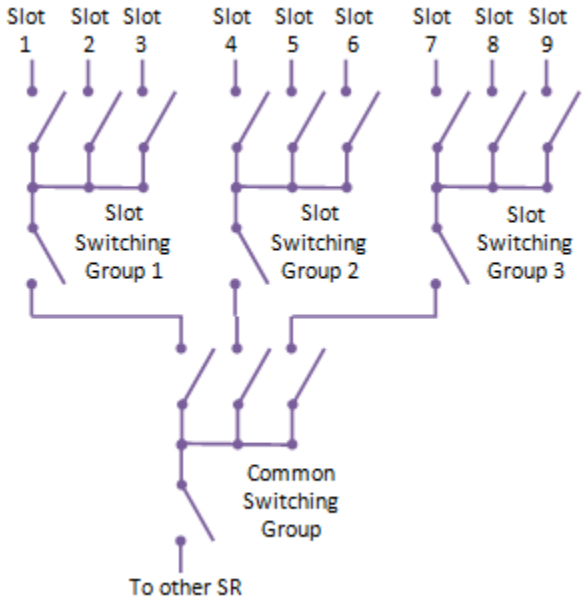


**Figure 2-13: Schematic Diagram for One Bus Signal of the Switching Router**

## Specifications

The following table outlines the specifications of the GX6256/GX6192/GX6032/GX6009 switching cards:

| **Switching Specifications** | |
|---|---|
| GX6032 Switch Card | Configuration: 32 x16<br>Inputs: 32 to 128 (4 switch cards) |
| GX6256 LF Switch Card | Configuration: 16 x16 matrix with 2 to 1 selector, (16) 2:16 multiplexers<br>Inputs: 16 internal analog bus and 16 digital<br>Outputs: 256 |
| GX6192 HF Switch Card | Configuration: 16 x 16 matrix with (16) 1:12 multiplexers<br>Inputs: 16 external analog inputs or 16 internal analog bus<br>Outputs: 192 |
| Relay Specifications | Max AC pk or DC voltage 170 V<br>Max DC or AC pk carry current: 1A<br>Max DC or AC pk switching current: 0.5A<br>Max. contact rating:: 10 W |
| **Signal Bandwidth** | |
| Input to Output (input switch card to LF switch card output) | > 20 MHz |
| HF Switch Card (external input to output) | > 100 MHz |
| **I/O Connections** | |
| GX6032 Switch Card | 32 coaxial connections (Positronics BD8W8F85R70T0/AA) |
| GX6256 Switch Card | Receiver: Dual 200 pin connector<br>Digital I/O: 34 pin, dual row latched header |
| GX6192 Switch Card | Receiver: Dual 200 pin connector<br>Digital I/O: 34 pin, dual row latched header<br>External I/O: 16 coaxial connections (Positronics BD8W8F85R70T0/AA) |

The following table outlines the specifications of the GX7016 Chassis:

| **Chassis Specifications** | |
|---|---|
| Input AC Power | 90 to 264 VAC, 12 A max (PFC)<br>47 to 63 Hz |
| Total Available DC Power | 755 W |
| +5 V<br>+3.3 V<br>+12 V<br>-12 V<br>Note: Total output power cannot exceed 755 W. | 60 A (max)<br>60 A (max)<br>30 A (max)<br>5 A (max) |
| Power Supply Monitoring | Monitored voltages: 3.3, 5, +12, -12, VIO value<br>Accuracy: ± 2% of reading |

| | |
|---|---|
| Temperature Monitoring | Per slot monitoring, 1 reading/sec/slot<br>4 second moving average value<br>User selectable alarm criteria:<br>Maximum slot temperature<br>Average slot temperature<br>Accuracy: ± 2 °C<br>Default warning and shutdown limits: +50 °C & +70 °C<br>Warning and shutdown limits programmable via software driver<br>Status: Query via software driver and audible alarm for a warning limit condition |
| PXI Triggers | Slots: 2 – 20<br>Number: 8 per segment<br>Software controlled segment mapping supports:<br>Isolate a trigger line within a segment<br>Map a trigger line left to right<br>Map a trigger line right to left |
| PXI Clock | Integrated 10 MHz PXI clock with auto-detect function. Presence of an external 10 MHz PXI clock will disable the internal clock. PXI clock is distributed to all peripheral slots.<br>10 MHz PXI clock accuracy: ±100 ppm<br>External input: Rear panel (TTL compatible) or via timing slot<br>Output: Rear panel, (TTL compatible) |
| Slots | 20 PXI or cPCI Slots (19 instruments max) |
| Weight (core system) | 61 lbs |
| Overall Dimensions | 10U (17.5") H x 19" W x 24" D |
| Cooling | Four 79 CFM fans for system cooling. Integrated temperature monitoring via an on-board microcontroller with audible and software notification when preset temperature limits are exceeded. Fan speed control and monitoring is automatic and can be controlled / monitored via the GxChassis software. |

The following table outlines the environmental specifications of the switching subsystem:

| **Environmental Specifications** | |
|---|---|
| Operating | 0 °C to +50 °C |
| Storage | -20 °C to +60 °C |
| Relative Humidity (operating) | 5% to 80% RH,non-condensing |
| Relative Humidity (non-operating) | 5% to 95% RH,non-condensing |
| Altitude (operating) | Up to 2000 M |
| CE Compliance | EN61010-1 (pending)<br>EN61326 (pending) |

## Virtual Panel

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 **GX6256 Panel** from the **Marvin Test Solutions**, **GXSW** menu under the **Start** menu. The GX6256/GX6192/GX6032/GX6009 virtual panel opens as shown here:

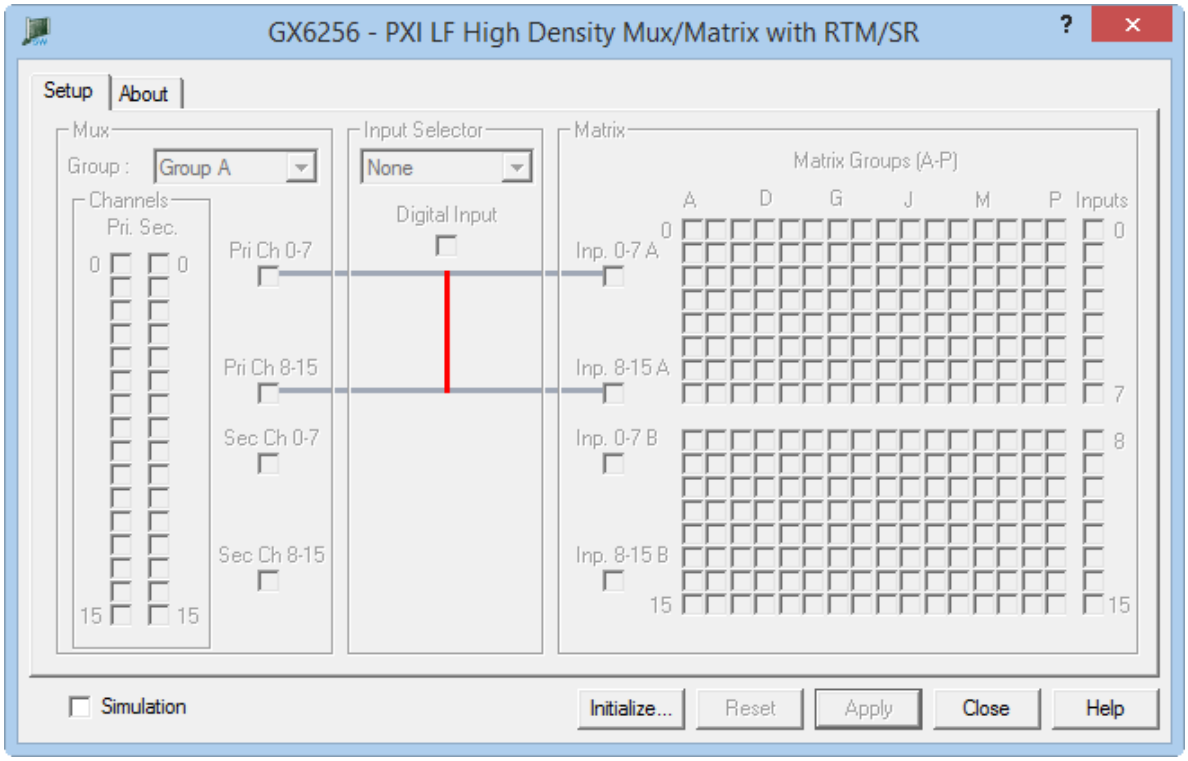


**Figure 2-14: Virtual Panel - Uninitialized**

**Simulation** – Selects simulation mode. When the check box is checked when the **Initialize** button is clicked, the panel will operate in simulation mode. If the check box is not checked when the **Initialize** button is clicked, the panel will control actual hardware.

**Initialize** – Scans the current system for any components of the GX7016 switching sub system (GX6256, GX6192, GX6009, and GX6032) and initializes each one. The current settings of the each device will not change after calling initialize. The panel will reflect the current settings of each device.

**Reset** – Resets the device settings of the device that corresponds to the currently active page, to their default state.

**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 displays the information regarding this control.

### Virtual Panel - LF Slot Page

After the board is initialized the panel is enabled and will display the current setting of the board. The following picture shows the **LF Slot Page, 2-LF (**LF PXI card that is installed on PXI Slot #2) settings:

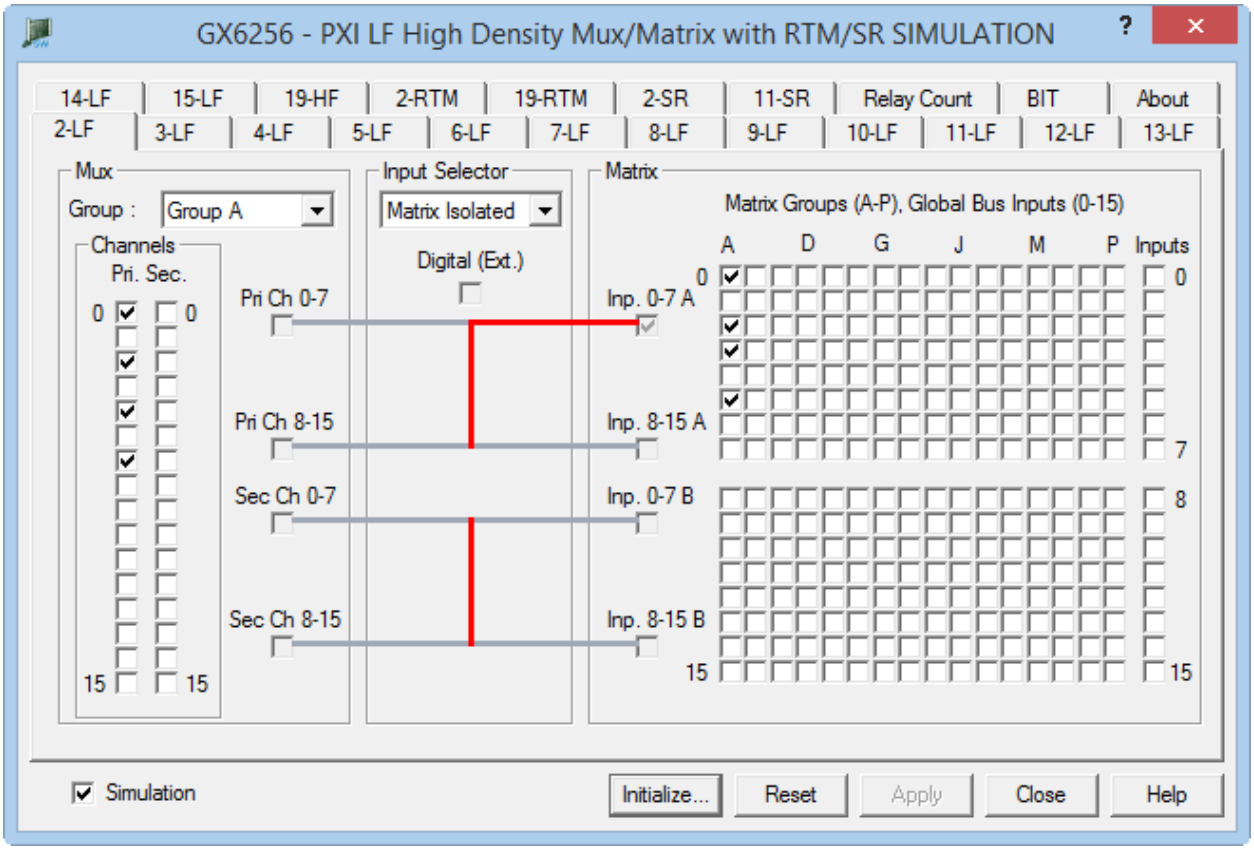


**Figure 2-15: Virtual Panel – LF (GX6256) Page**

The following controls are shown:

**Mux Group Selection (combo box):** Set/Displays the currently selected Mux Group

**Channels (check boxes):** Displays the current relay setting (closed or open). When a box in the matrix is checked, the relay in that row/column is closed. Similarly, unchecked boxes mean relays are open.

**Input Selector (combo box):** Sets/Displays the currently selected input selector mode

- **None:** The Mux, Matrix, and external lines are not connected to each other.
- **Matrix Only:** The Mux and Matrix are allowed to connect but the external lines are disconnected.
- **Ext. Only:** The Mux and the external lines are allowed to connect to each other.
- **Matrix and Ext.:** The Mux is allowed to connect to the Matrix and external lines.
- **Matrix Isolated:** The Mux is not allowed to connect to the selector, effectively making it isolated from the rest of the board.

**Digital (Ext.) (check box):** Displays the state of the connection between the digital external input and the input selector

**Pri Ch 0-7 (check box):** Displays the state of the connection between the primary mux low channels to the input selector

**Pri Ch 8-15 (check box):** Displays the state of the connection between the primary mux high channels to the input selector

**Sec Ch 0-7 (check box):** Displays the state of the connection between the secondary mux low channels to the input selector

**Sec Ch 8-15 (check box):** Displays the state of the connection between the secondary mux high channels to the input selector

**Inp. Ch 0-7 (check box):** Displays the state of the connection between the matrix low channels to the input selector

**Inp. Ch 8-15 (check box):** Displays the state of the connection between the matrix high channels to the input selector

**Matrix (check boxes):** Sets/Displays the states of each of the GX6256 Matrix relays. If the check box is checked, the associated relay is closed. If the check box is not checked, the associated relay is open.

**Inputs (check boxes):** Sets/Displays the states of each of the global bus input relays. If the check box is checked, the associated relay is closed. If the check box is not checked, the associated relay is open.

### Virtual Panel - HF Slot Page

The following picture shows the H**F Slot Page, 19-HF (**HF PXI card that is installed on PXI Slot #19) settings:

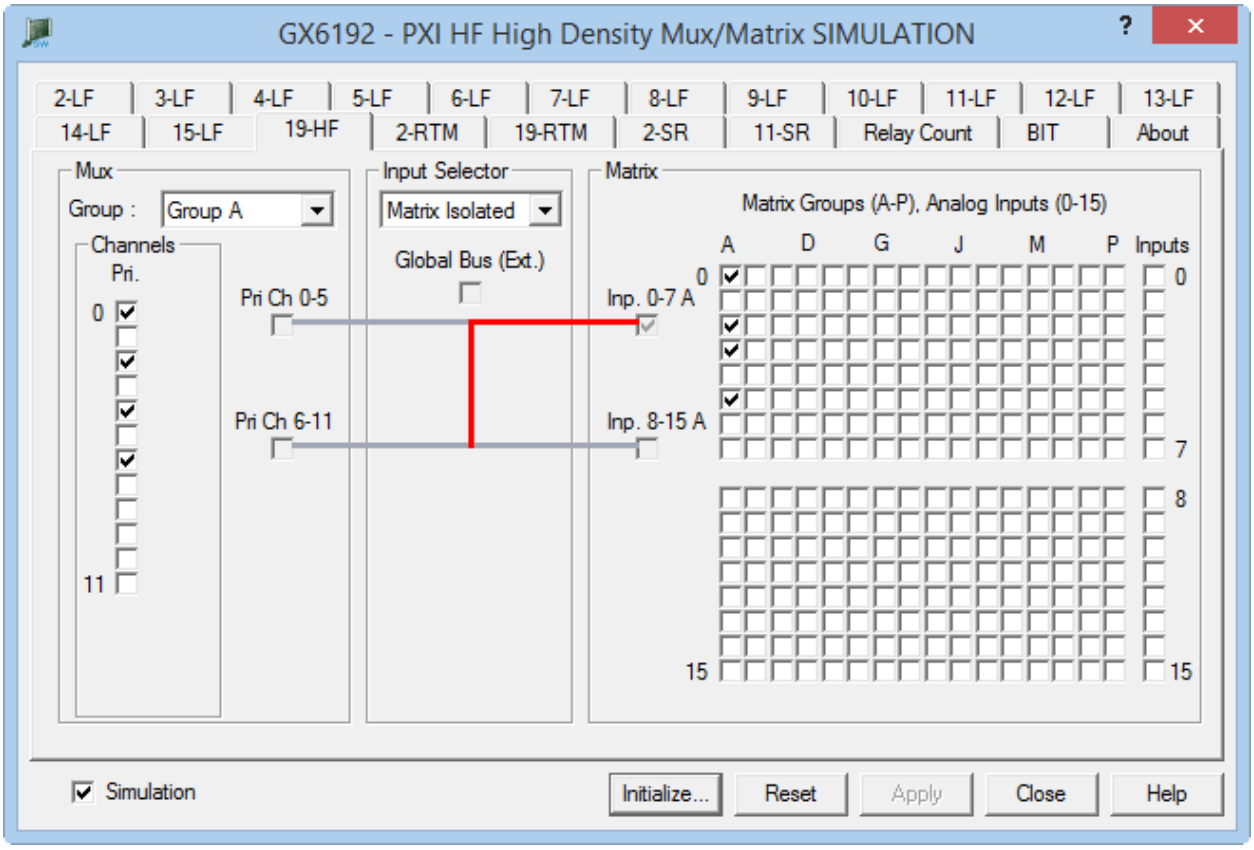


**Figure 2-16: Virtual Panel – HF (GX6192) Page**

The following controls are shown:

**Mux Group Selection (combo box):** Set/Displays the currently selected Mux Group

**Channels (check boxes):** Displays the current relay setting (closed or open). When a box in the matrix is checked, the relay in that row/column is closed. Similarly, unchecked boxes mean relays are open.

**Input Selector (combo box):** Sets/Displays the currently selected input selector mode

- **None:** The Mux, Matrix, and external lines are not connected to each other.
- **Matrix Only:** The Mux and Matrix are allowed to connect but the external lines are disconnected.
- **Ext. Only:** The Mux and the external lines are allowed to connect to each other.
- **Matrix and Ext.:** The Mux is allowed to connect to the Matrix and external lines.
- **Matrix Isolated:** The Mux is not allowed to connect to the selector, effectively making it isolated from the rest of the board.

**Global Bus (Ext.) (check box):** Displays the state of the connection between the global bus external input and the input selector

**Pri Ch 0-7 (check box):** Displays the state of the connection between the primary mux low channels to the input selector

**Pri Ch 8-15 (check box):** Displays the state of the connection between the primary mux high channels to the input selector

**Sec Ch 0-7 (check box):** Displays the state of the connection between the secondary mux low channels to the input selector

**Sec Ch 8-15 (check box):** Displays the state of the connection between the secondary mux high channels to the input selector

**Inp. Ch 0-7 (check box):** Displays the state of the connection between the matrix low channels to the input selector

**Inp. Ch 8-15 (check box):** Displays the state of the connection between the matrix high channels to the input selector

**Matrix (check boxes):** Sets/Displays the states of each of the GX6256 Matrix relays. If the check box is checked, the associated relay is closed. If the check box is not checked, the associated relay is open.

**Inputs (check boxes):** Sets/Displays the states of each of the instrument input relays. If the check box is checked, the associated relay is closed. If the check box is not checked, the associated relay is open.

### Virtual Panel - Rear Transition Module (RTM) Page

The following picture shows the **RTM page, 2-RTM (**RTM module that is installed on PXI backplane on slot #2, controlled using 2-LF) settings:

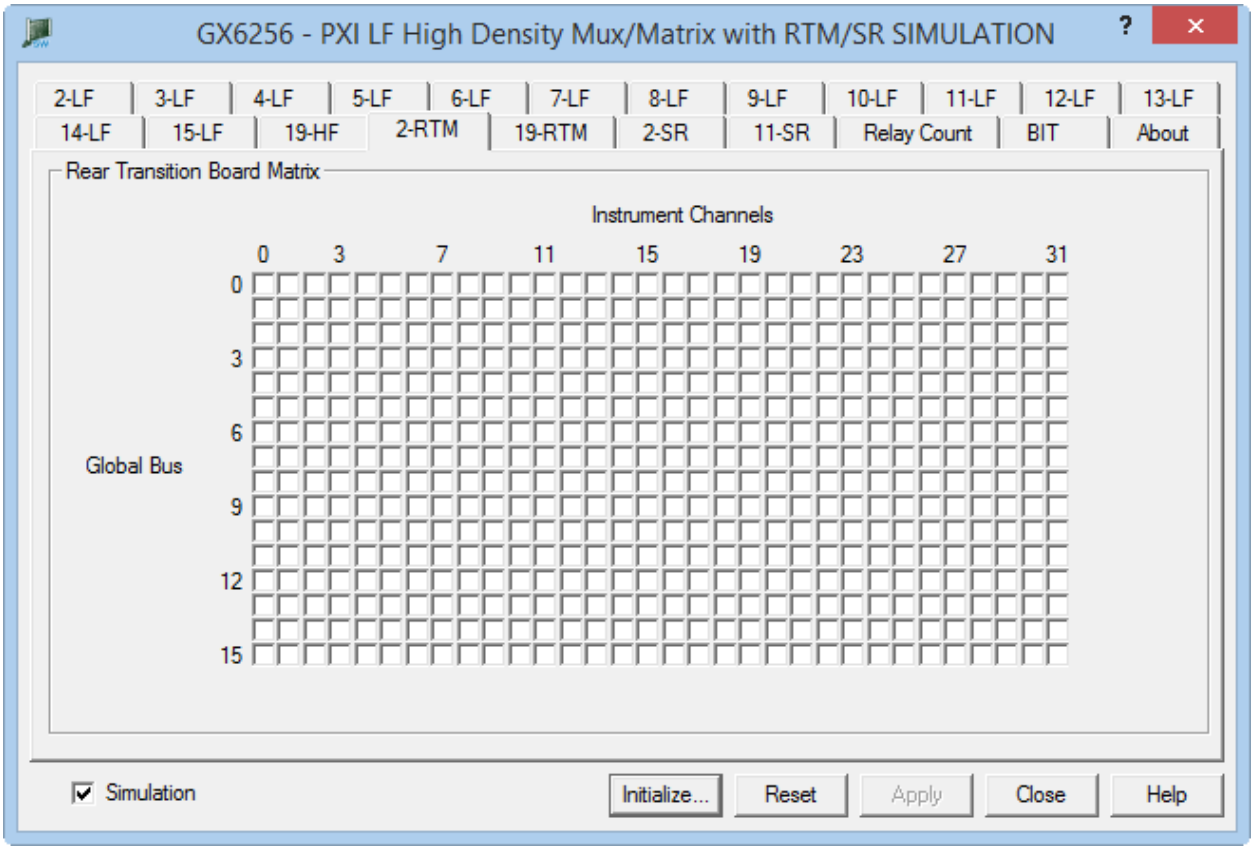


**Figure 2-17: Virtual Panel - Rear Transition Module (GX6032) Page**

The following controls are shown:

**Rear Transition Board Matrix (check boxes):** Sets/Displays the states of each of the rear transition module relays. If the check box is checked, the associated relay is closed. If the check box is not checked, the associated relay is open.

### Virtual Panel - Switch Router (SR) Page

The following picture shows the **SR page, 2-SR (**SR module that is installed on PXI backplane on slot #2, controlled using 2-LF) settings:

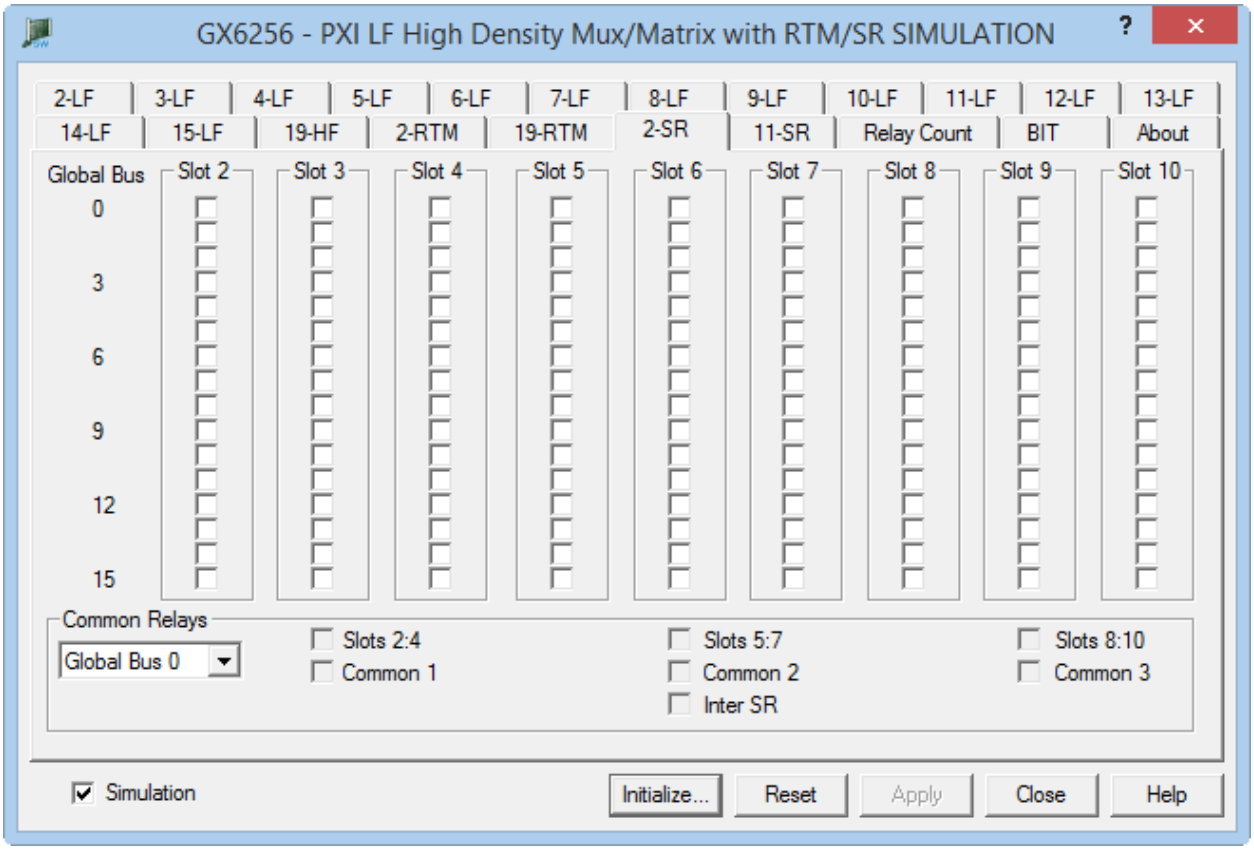


**Figure 2-18: Virtual Panel - Switching Router Module (GX6009) Page**

The following controls are shown:

**Slot Global Bus (check boxes):** Sets/Displays the states of each of the switch router relays. If the check box is checked, the associated relay is closed. If the check box is not checked, the associated relay is open. Each relay corresponds to a particular global bus line (0-15).

**Common Relays Global Bus Selection (combo box):** Set/Displays the currently selected global bus line

**Slots 2:4 and Common 1 (check boxes):** Displays the status of the relay that connects slots 2, 3, and 4 to the other slots

**Slots 5:7 and Common 2 (check boxes):** Displays the status of the relay that connects slots 5, 6, and 7 to the other slots

**Slots 8:10 and Common 3 (check boxes):** Displays the status of the relay that connects slots 8, 9, and 10 to the other slots

**Inter SR (check box):** Displays the status of the relay that connects this switch router to the other switch router

### Virtual Panel - Relay Count Page

Clicking on the **Relay Count** tab will show the **Relay Count Page** as shown in Figure 2-19:

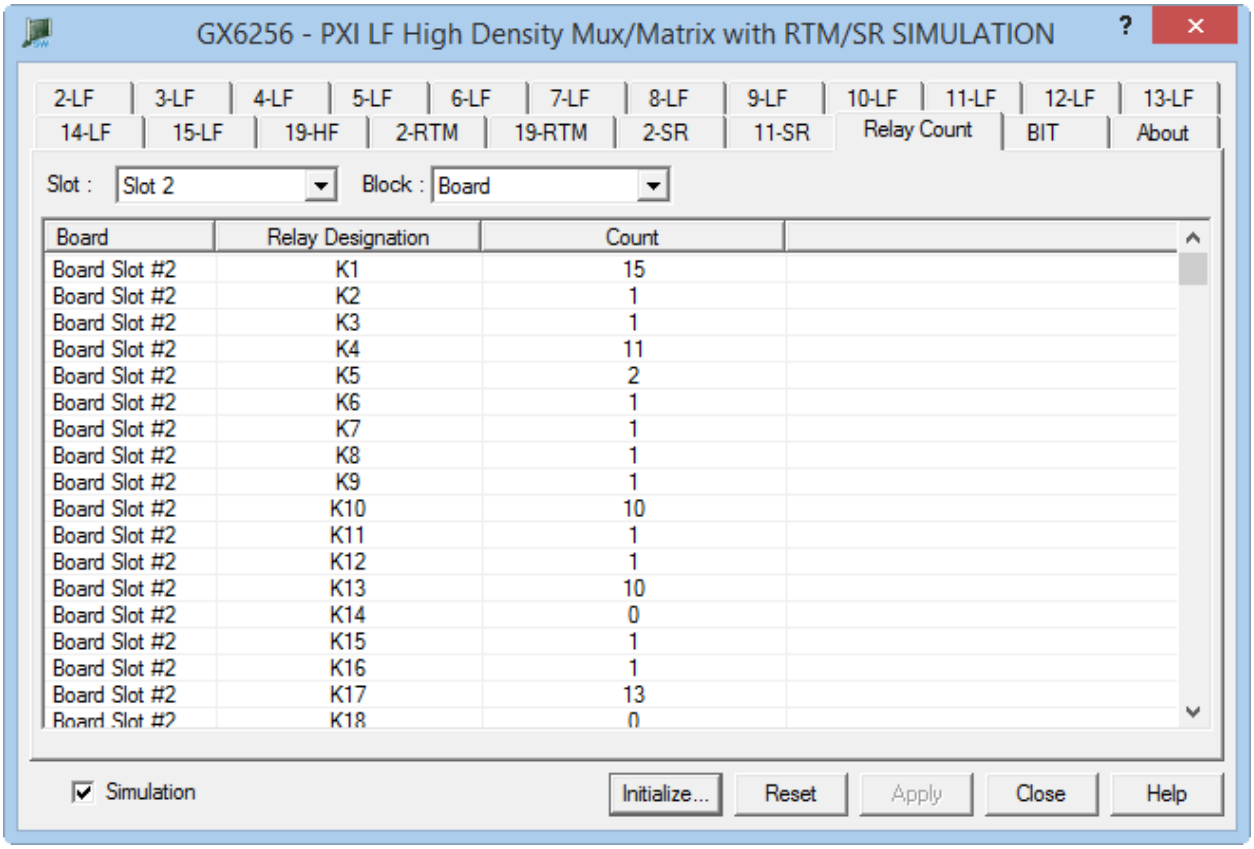


**Figure 2-19: Virtual Panel – Relay Count Page**

The following controls are shown in the Relay Count page:

**Slot Selection (combo box):** Set/Displays the currently selected board's slot

**Block Selection (combo box):** Sets/Displays the currently selected block (board, RTM, or SR) of the currently selected board

- **Board:** Show relays that are located only on the board (GX6256 or GX6192) that resides in the selected slot.
- **RTM:** Show relays that are located only on the Rear Transition Module (GX6032) that resides in the selected slot.
- **Switch Router:** Show relays that are located only on the Switch Router (GX6009) that resides in the selected slot.

**Relay Count List (list control):** Displays the number of openings and closures of each relay within the currently selected block of the currently selected board (slot).

### Virtual Panel - Built In Test (BIT) Page

Clicking on the **BIT** tab will show the **BIT Page** as shown here:

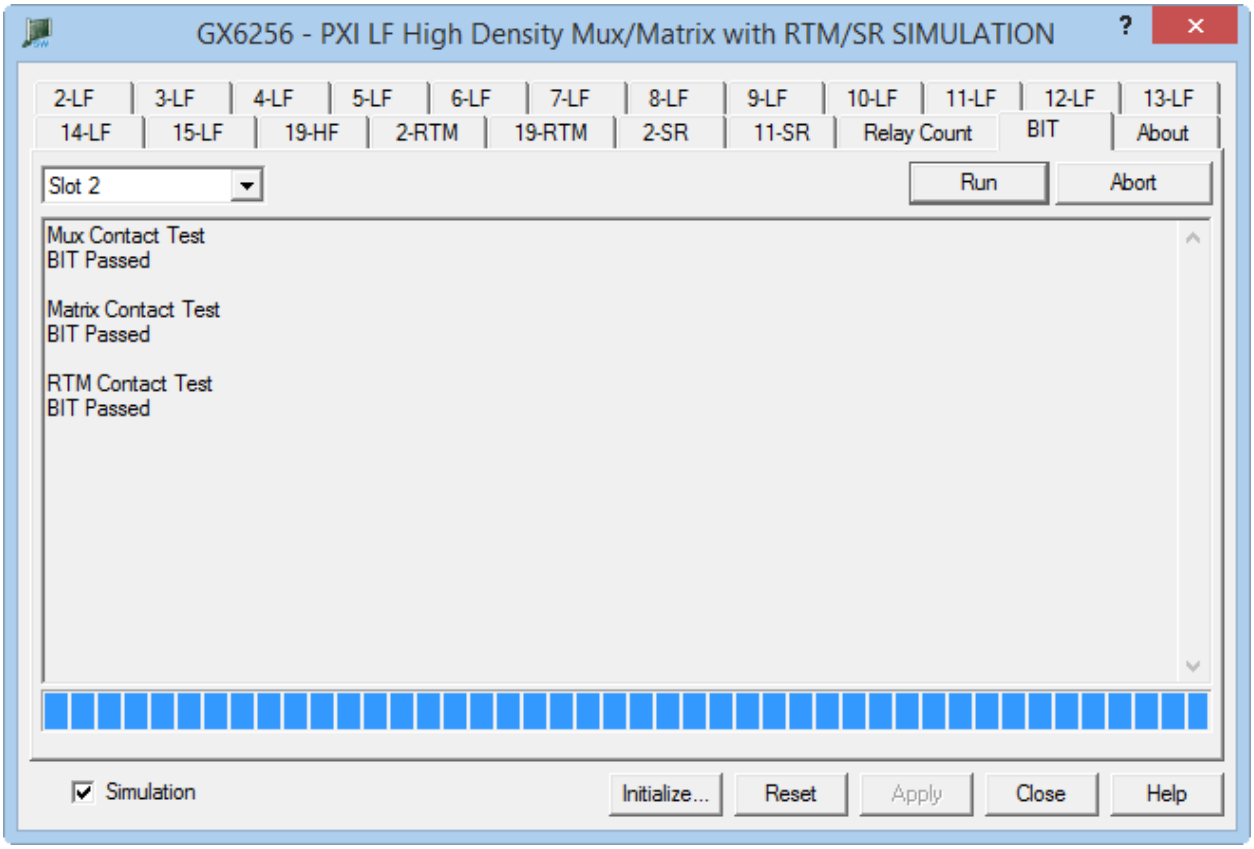


**Figure 2-20: GX6256/GX6192 Virtual Panel – Virtual Panel - Built In Test (BIT) Page**

The following controls are shown in the Built In Test (BIT) page:

**Slot Selection (combo box):** Set/Displays the currently selected board's slot

**Run Button (button):** Starts running the Built In Test (BIT)

**Abort Button (button):** Aborts a currently running Built In Test (BIT)

**Built In Test Results (text box):** Displays the results of the last completed Built In Test (BIT)

### Virtual Panel About Page

Clicking on the **About** tab will show the **About Page** as shown in Figure 2-21:

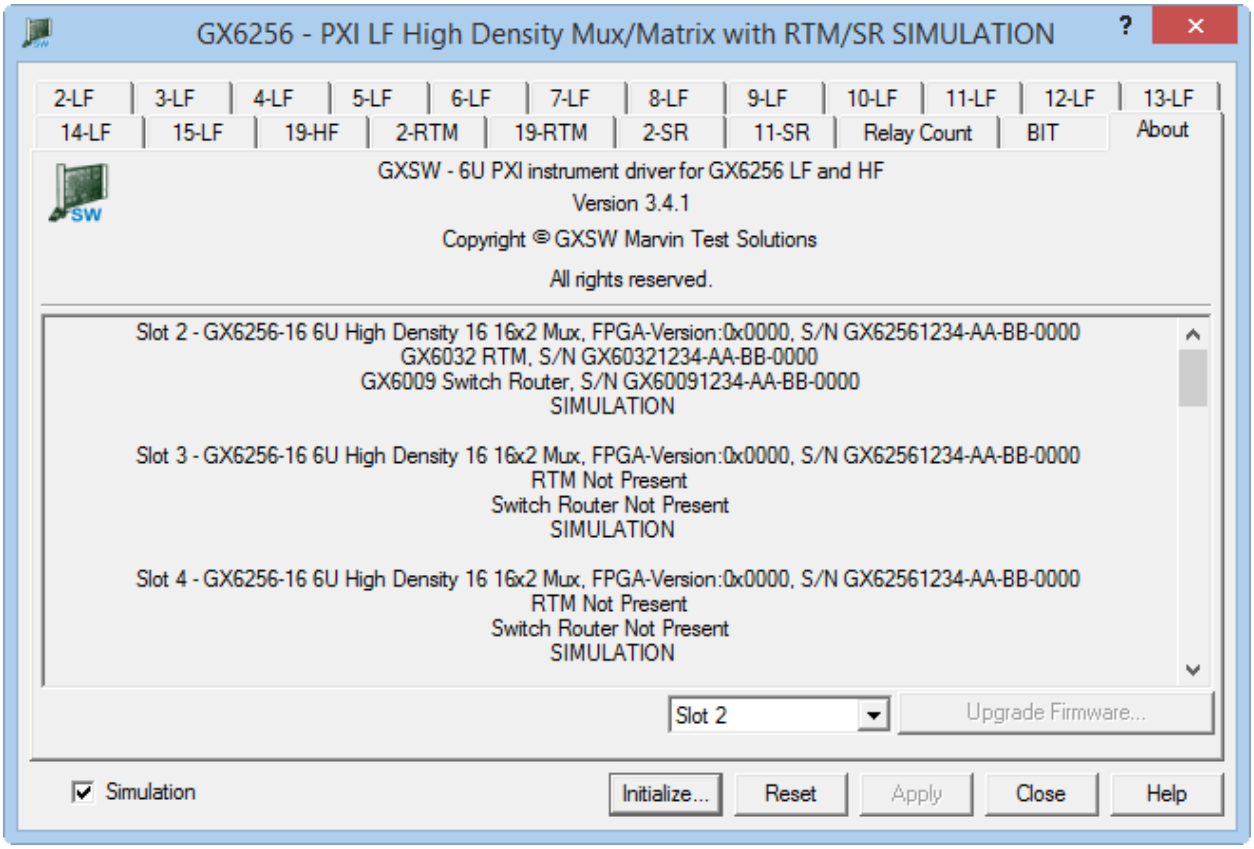


**Figure 2-21: Virtual Panel – About Page**

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.

**Slot Selection (combo box):** Set/Displays the currently selected board's slot for upgrading firmware.

# Chapter 3 - Installation and Connections

## Getting Started

This section includes general hardware installation procedures for the GX6256/GX6192 board and installation instructions for the GX6256/GX6192/GX6032/GX6009 (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 |
| Driver Installation | This Chapter |
| Function Reference | Chapter 5 |

### Packing List

All GX6256/GX6192 boards have the same basic packing list, which includes:

1. GX6256/GX6192 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 switching instrument boards are designed for use with a 6U cPCI or PXI compatible chassis. The software is compatible with any computer system running Windows

## 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 you 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 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 of `C:\Program Files\Marvin Test Solutions\`GXSW under 32-bit Windows, or `C:\Program Files (x86)\Marvin Test Solutions\`GXSW under 64-bit Windows.
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 GX7016 Switching Subsystem's various features via an interactive user interface.
- **GXSW driver** - A DLL based function library (GXSW.DLL for 32-bit applications or GXSW64.dll for 64-bit applications, 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. 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 Switching Subsystem, 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.vxipnp.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.

In addition to the GXSW software components, this manual also described the **SwitchEasy** software that is based on the GXSW driver and provides a rule-based, pin mapping and higher level access to the switching subsystem. SwitchEasy software is described later in this manual.

## Installation Folders

The GX6256/GX6192 driver files are installed in the default folder `C:\Program Files\Marvin Test Solutions\GXSW` under 32-bit Windows or `C:\Program Files (x86)\Marvin Test Solutions\GXSW` under 64-bit Windows . 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 64 and 32-bit system folders | Contains the GXSW DLL and GXSW64.DLL drivers, 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 **Gx6256Initialize** and **Gx6256InitializeVisa** 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).

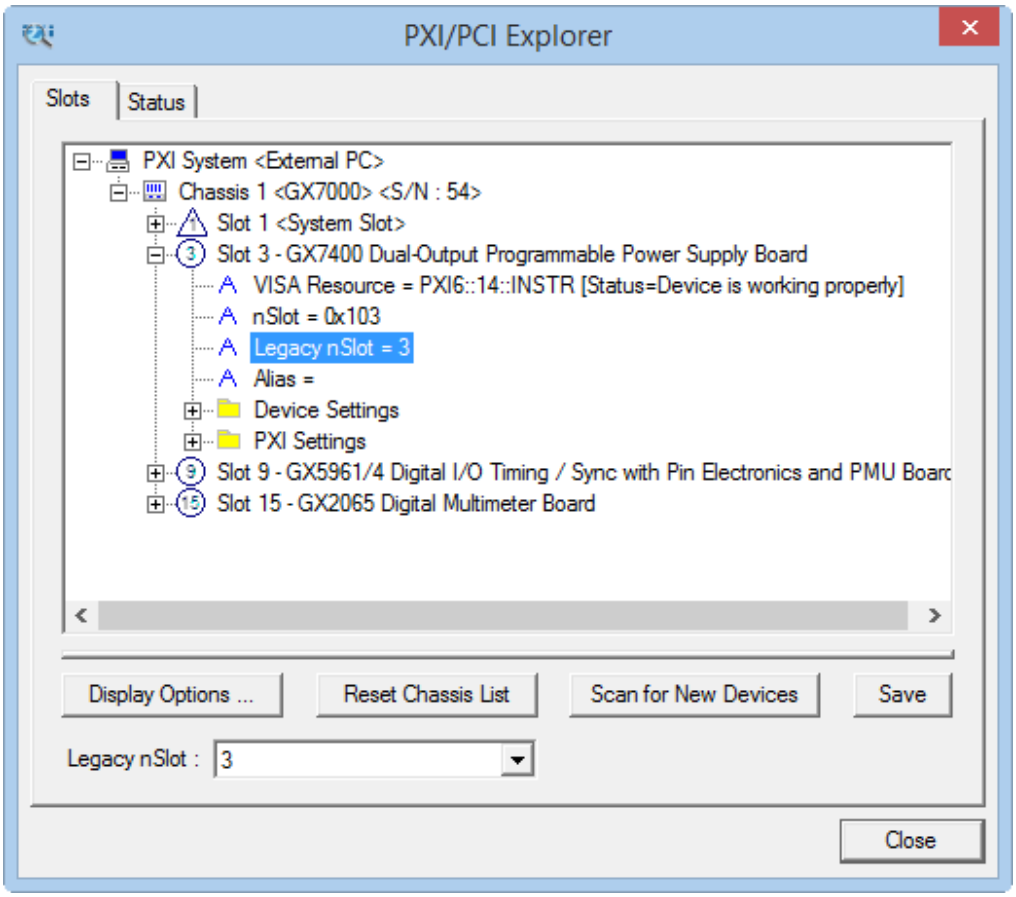


**Figure 3-1: PXI/PCI Explorer**

## 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 GX6256 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 PXI 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).

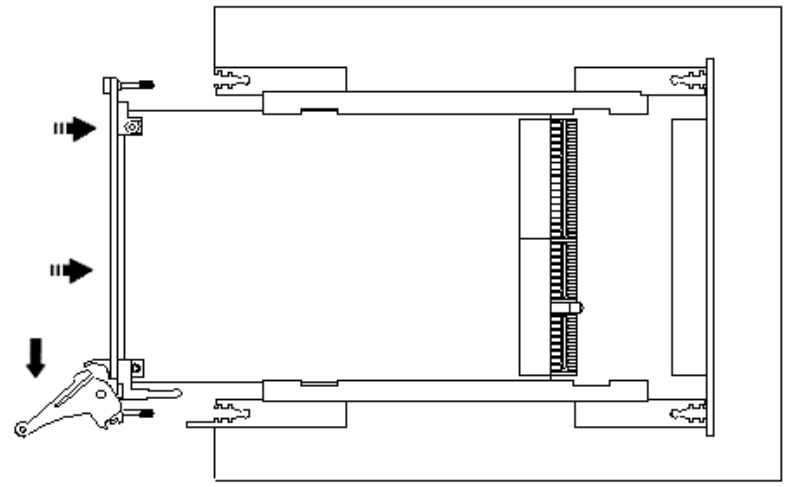

**Figure 3-2: Ejector handles position during module insertion**

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

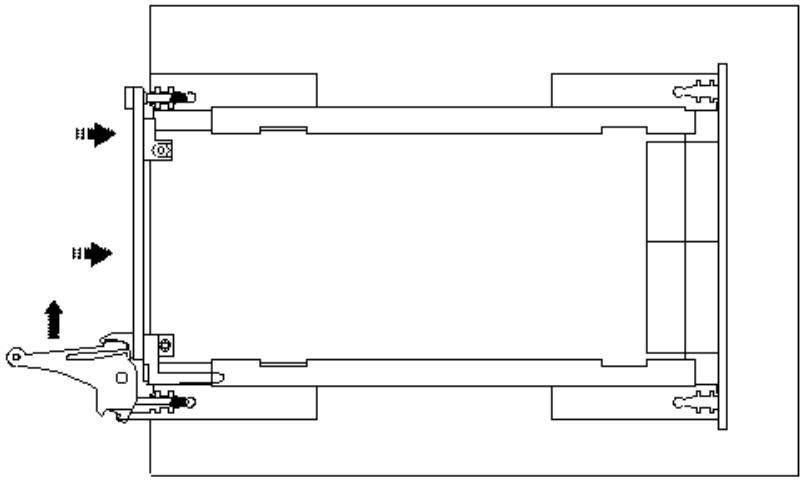

**Figure 3-3: Ejector handles position after module insertion**

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 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 Program Files\Marvin Test Solutions\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 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 Program File folder under \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

### GX6256

Figure 3-4 shows the GX6256 board connectors and jumpers followed by their description:


**Figure 3-4: GX6256 Connectors and Jumpers**

**J2 – Front Connector:** Mac-Panel DAK connector (200 pin).

**J3 – Front Connector:** Mac-Panel DAK connector (200 pin).

**J7 – Instrument Connector:** Shrouded right angle IDC header (34 pin).

**JP1 – Chassis Ground Jumper:** Connects chassis to power ground

### GX6192

Figure 3-5 shows the GX6192 board connectors and jumpers followed by their description:

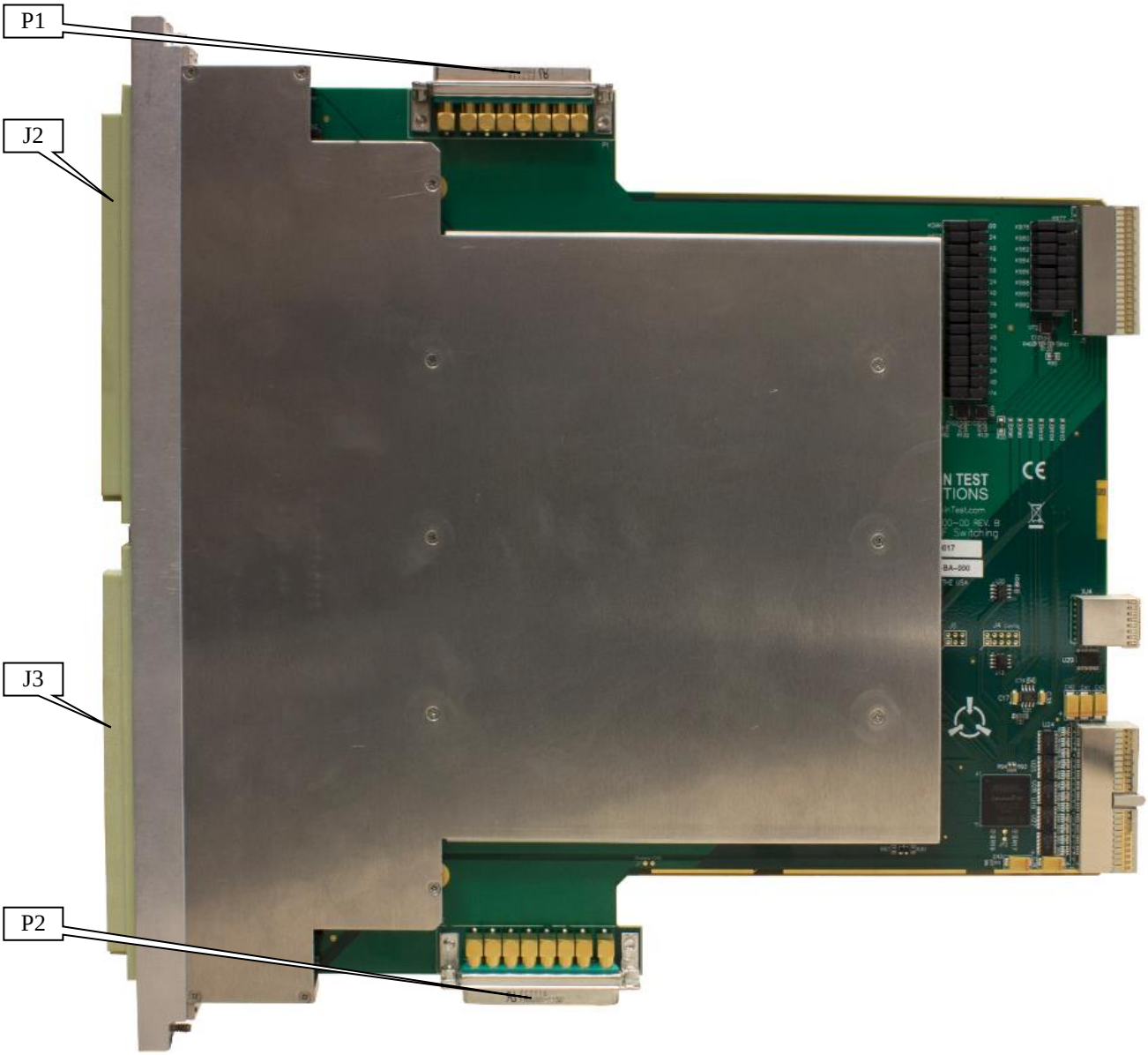


**Figure 3-5: GX6192 Connectors and Jumpers**

**J2 – Front Connector:** Mac-Panel DAK connector (200 pin).

**J3 – Front Connector:** Mac-Panel DAK connector (200 pin).

**P1 – Analog Input Connector:** BNC (8 pin).

**P2 – Analog Input Connector:** BNC (8 pin).

## Connectors and Accessories

### GX6256

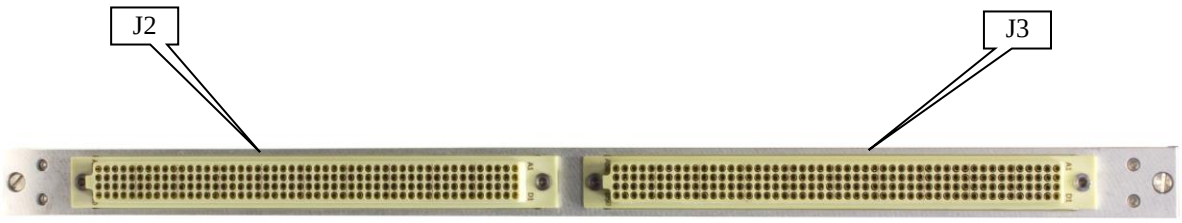


**Figure 3-6: GX6256 front Connectors**

### GX6192

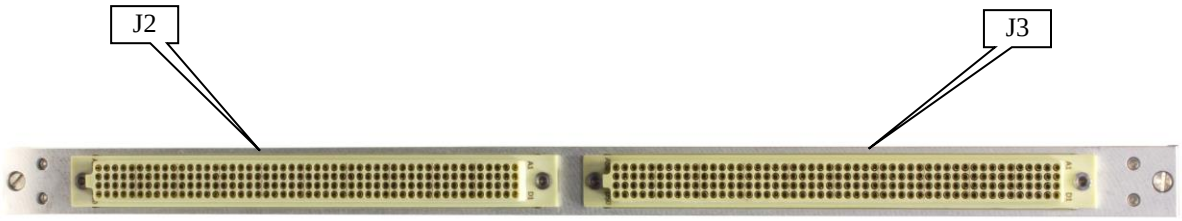


**Figure 3-7: GX6256 front Connectors**

## GX6256 J2 Connector

The following table describes the J2 (200 pin) connector's pin out:

| Pin | Column A | Column B | Column C | Column D |
|---|---|---|---|---|
| 1 | UUT GND | UUT GND | UUT GND | UUT GND |
| 2 | Group A Channel 1 | Group B Channel 1 | Group C Channel 1 | Group D Channel 1 |
| 3 | Group A Channel 2 | Group B Channel 2 | Group C Channel 2 | Group D Channel 2 |
| 4 | Group A Channel 3 | Group B Channel 3 | Group C Channel 3 | Group D Channel 3 |
| 5 | Group A Channel 4 | Group B Channel 4 | Group C Channel 4 | Group D Channel 4 |
| 6 | UUT GND | UUT GND | UUT GND | UUT GND |
| 7 | UUT GND | UUT GND | UUT GND | UUT GND |
| 8 | Group A Channel 5 | Group B Channel 5 | Group C Channel 5 | Group D Channel 5 |
| 9 | Group A Channel 6 | Group B Channel 6 | Group C Channel 6 | Group D Channel 6 |
| 10 | Group A Channel 7 | Group B Channel 7 | Group C Channel 7 | Group D Channel 7 |
| 11 | Group A Channel 8 | Group B Channel 8 | Group C Channel 8 | Group D Channel 8 |
| 12 | UUT GND | UUT GND | UUT GND | UUT GND |
| 13 | UUT GND | UUT GND | UUT GND | UUT GND |
| 14 | Group A Channel 9 | Group B Channel 9 | Group C Channel 9 | Group D Channel 9 |
| 15 | Group A Channel 10 | Group B Channel 10 | Group C Channel 10 | Group D Channel 10 |
| 16 | Group A Channel 11 | Group B Channel 11 | Group C Channel 11 | Group D Channel 11 |
| 17 | Group A Channel 12 | Group B Channel 12 | Group C Channel 12 | Group D Channel 12 |
| 18 | UUT GND | UUT GND | UUT GND | UUT GND |
| 19 | UUT GND | UUT GND | UUT GND | UUT GND |
| 20 | Group A Channel 13 | Group B Channel 13 | Group C Channel 13 | Group D Channel 13 |
| 21 | Group A Channel 14 | Group B Channel 14 | Group C Channel 14 | Group D Channel 14 |
| 22 | Group A Channel 15 | Group B Channel 15 | Group C Channel 15 | Group D Channel 15 |
| 23 | Group A Channel 16 | Group B Channel 16 | Group C Channel 16 | Group D Channel 16 |
| 24 | UUT GND | UUT GND | UUT GND | UUT GND |
| 25 | I/O0 | I/O0 | I/O0 | I/O0 |
| 26 | CHASSIS | CHASSIS | CHASSIS | CHASSIS |
| 27 | UUT GND | UUT GND | UUT GND | UUT GND |
| 28 | Group E Channel 1 | Group F Channel 1 | Group G Channel 1 | Group H Channel 1 |
| 29 | Group E Channel 2 | Group F Channel 2 | Group G Channel 2 | Group H Channel 2 |
| 30 | Group E Channel 3 | Group F Channel 3 | Group G Channel 3 | Group H Channel 3 |
| 31 | Group E Channel 4 | Group F Channel 4 | Group G Channel 4 | Group H Channel 4 |
| 32 | UUT GND | UUT GND | UUT GND | UUT GND |
| 33 | UUT GND | UUT GND | UUT GND | UUT GND |

| | | | | |
|---|---|---|---|---|
| 34 | Group E Channel 5 | Group F Channel 5 | Group G Channel 5 | Group H Channel 5 |
| 35 | Group E Channel 6 | Group F Channel 6 | Group G Channel 6 | Group H Channel 6 |
| 36 | Group E Channel 7 | Group F Channel 7 | Group G Channel 7 | Group H Channel 7 |
| 37 | Group E Channel 8 | Group F Channel 8 | Group G Channel 8 | Group H Channel 8 |
| 38 | UUT GND | UUT GND | UUT GND | UUT GND |
| 39 | UUT GND | UUT GND | UUT GND | UUT GND |
| 40 | Group E Channel 9 | Group F Channel 9 | Group G Channel 9 | Group H Channel 9 |
| 41 | Group E Channel 10 | Group F Channel 10 | Group G Channel 10 | Group H Channel 10 |
| 42 | Group E Channel 11 | Group F Channel 11 | Group G Channel 11 | Group H Channel 11 |
| 43 | Group E Channel 12 | Group F Channel 12 | Group G Channel 12 | Group H Channel 12 |
| 44 | UUT GND | UUT GND | UUT GND | UUT GND |
| 45 | UUT GND | UUT GND | UUT GND | UUT GND |
| 46 | Group E Channel 13 | Group F Channel 13 | Group G Channel 13 | Group H Channel 13 |
| 47 | Group E Channel 14 | Group F Channel 14 | Group G Channel 14 | Group H Channel 14 |
| 48 | Group E Channel 15 | Group F Channel 15 | Group G Channel 15 | Group H Channel 15 |
| 49 | Group E Channel 16 | Group F Channel 16 | Group G Channel 16 | Group H Channel 16 |
| 50 | UUT GND | UUT GND | UUT GND | UUT GND |

**Table 3-1: GX6256 Connector J2**

The following table describes the signals mentioned in Table 3-1:

| Signal name | Comments |
|---|---|
| Group X Channel Y | Mux Group channels for group A through H |
| UUT GND | Unit Under Test ground. This ground is isolated from power ground and chassis. |
| I/O X | PIO (Static general purpose I/O) pins 0 to 3 |

## GX6256 J3 Connector

The following table describes the J3 (200 pin) connector's pin out:

| Pin | Column A | Column B | Column C | Column D |
|---|---|---|---|---|
| 1 | UUT GND | UUT GND | UUT GND | UUT GND |
| 2 | Group I Channel 1 | Group J Channel 1 | Group K Channel 1 | Group L Channel 1 |
| 3 | Group I Channel 2 | Group J Channel 2 | Group K Channel 2 | Group L Channel 2 |
| 4 | Group I Channel 3 | Group J Channel 3 | Group K Channel 3 | Group L Channel 3 |
| 5 | Group I Channel 4 | Group J Channel 4 | Group K Channel 4 | Group L Channel 4 |
| 6 | UUT GND | UUT GND | UUT GND | UUT GND |
| 7 | UUT GND | UUT GND | UUT GND | UUT GND |
| 8 | Group I Channel 5 | Group J Channel 5 | Group K Channel 5 | Group L Channel 5 |
| 9 | Group I Channel 6 | Group J Channel 6 | Group K Channel 6 | Group L Channel 6 |
| 10 | Group I Channel 7 | Group J Channel 7 | Group K Channel 7 | Group L Channel 7 |
| 11 | Group I Channel 8 | Group J Channel 8 | Group K Channel 8 | Group L Channel 8 |
| 12 | UUT GND | UUT GND | UUT GND | UUT GND |
| 13 | UUT GND | UUT GND | UUT GND | UUT GND |
| 14 | Group I Channel 9 | Group J Channel 9 | Group K Channel 9 | Group L Channel 9 |
| 15 | Group I Channel 10 | Group J Channel 10 | Group K Channel 10 | Group L Channel 10 |
| 16 | Group I Channel 11 | Group J Channel 11 | Group K Channel 11 | Group L Channel 11 |
| 17 | Group I Channel 12 | Group J Channel 12 | Group K Channel 12 | Group L Channel 12 |
| 18 | UUT GND | UUT GND | UUT GND | UUT GND |
| 19 | UUT GND | UUT GND | UUT GND | UUT GND |
| 20 | Group I Channel 13 | Group J Channel 13 | Group K Channel 13 | Group L Channel 13 |
| 21 | Group I Channel 14 | Group J Channel 14 | Group K Channel 14 | Group L Channel 14 |
| 22 | Group I Channel 15 | Group J Channel 15 | Group K Channel 15 | Group L Channel 15 |
| 23 | Group I Channel 16 | Group J Channel 16 | Group K Channel 16 | Group L Channel 16 |
| 24 | UUT GND | UUT GND | UUT GND | UUT GND |
| 25 | I/O0 | I/O0 | I/O0 | I/O0 |
| 26 | CHASSIS | CHASSIS | CHASSIS | CHASSIS |
| 27 | UUT GND | UUT GND | UUT GND | UUT GND |
| 28 | Group M Channel 1 | Group N Channel 1 | Group O Channel 1 | Group P Channel 1 |
| 29 | Group M Channel 2 | Group N Channel 2 | Group O Channel 2 | Group P Channel 2 |
| 30 | Group M Channel 3 | Group N Channel 3 | Group O Channel 3 | Group P Channel 3 |
| 31 | Group M Channel 4 | Group N Channel 4 | Group O Channel 4 | Group P Channel 4 |
| 32 | UUT GND | UUT GND | UUT GND | UUT GND |
| 33 | UUT GND | UUT GND | UUT GND | UUT GND |

| | | | | |
|---|---|---|---|---|
| 34 | Group M Channel 5 | Group N Channel 5 | Group O Channel 5 | Group P Channel 5 |
| 35 | Group M Channel 6 | Group N Channel 6 | Group O Channel 6 | Group P Channel 6 |
| 36 | Group M Channel 7 | Group N Channel 7 | Group O Channel 7 | Group P Channel 7 |
| 37 | Group M Channel 8 | Group N Channel 8 | Group O Channel 8 | Group P Channel 8 |
| 38 | UUT GND | UUT GND | UUT GND | UUT GND |
| 39 | UUT GND | UUT GND | UUT GND | UUT GND |
| 40 | Group M Channel 9 | Group N Channel 9 | Group O Channel 9 | Group P Channel 9 |
| 41 | Group M Channel 10 | Group N Channel 10 | Group O Channel 10 | Group P Channel 10 |
| 42 | Group M Channel 11 | Group N Channel 11 | Group O Channel 11 | Group P Channel 11 |
| 43 | Group M Channel 12 | Group N Channel 12 | Group O Channel 12 | Group P Channel 12 |
| 44 | UUT GND | UUT GND | UUT GND | UUT GND |
| 45 | UUT GND | UUT GND | UUT GND | UUT GND |
| 46 | Group M Channel 13 | Group N Channel 13 | Group O Channel 13 | Group P Channel 13 |
| 47 | Group M Channel 14 | Group N Channel 14 | Group O Channel 14 | Group P Channel 14 |
| 48 | Group M Channel 15 | Group N Channel 15 | Group O Channel 15 | Group P Channel 15 |
| 49 | Group M Channel 16 | Group N Channel 16 | Group O Channel 16 | Group P Channel 16 |
| 50 | UUT GND | UUT GND | UUT GND | UUT GND |

**Table 3-2: GX6256 Connector J3**

The following table describes the signals mentioned in Table 3-2:

| Signal name | Comments |
|---|---|
| Group X Channel Y | Mux Group channels for group I through P |
| UUT GND | Unit Under Test ground. This ground is isolated from power ground and chassis. |
| I/O X | PIO (Static general purpose I/O) pins 4 to 7 |

## GX6256 J7 Connector

| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | Din1 | 2 | UUT GND |
| 3 | Din2 | 4 | UUT GND |
| 5 | Din3 | 6 | UUT GND |
| 7 | Din4 | 8 | UUT GND |
| 9 | Din5 | 10 | UUT GND |
| 11 | Din6 | 12 | UUT GND |
| 13 | Din7 | 14 | UUT GND |
| 15 | Din8 | 16 | UUT GND |
| 17 | Din9 | 18 | UUT GND |
| 19 | Din10 | 20 | UUT GND |
| 21 | Din11 | 22 | UUT GND |
| 23 | Din12 | 24 | UUT GND |
| 25 | Din13 | 26 | UUT GND |
| 27 | Din14 | 28 | UUT GND |
| 29 | Din15 | 30 | UUT GND |
| 31 | Din16 | 32 | UUT GND |
| 33 | NC | 34 | DUTGND |

**Table 3-3: GX6256 Connector J7**

## GX6192 J2 Connector

The following table describes the J2 (200 pin) connector's pin out:

| Pin | Column A | Column B | Column C | Column D |
|---|---|---|---|---|
| 1 | Group A Channel 1 | UUT GND | Group C Channel 1 | UUT GND |
| 2 | UUT GND | Group B Channel 1 | UUT GND | Group D Channel 1 |
| 3 | Group A Channel 2 | UUT GND | Group C Channel 2 | UUT GND |
| 4 | UUT GND | Group B Channel 2 | UUT GND | Group D Channel 2 |
| 5 | Group A Channel 3 | UUT GND | Group C Channel 3 | UUT GND |
| 6 | UUT GND | Group B Channel 3 | UUT GND | Group D Channel 3 |
| 7 | Group A Channel 4 | UUT GND | Group C Channel 4 | UUT GND |
| 8 | UUT GND | Group B Channel 4 | UUT GND | Group D Channel 4 |
| 9 | Group A Channel 5 | UUT GND | Group C Channel 5 | UUT GND |
| 10 | UUT GND | Group B Channel 5 | UUT GND | Group D Channel 5 |
| 11 | Group A Channel 6 | UUT GND | Group C Channel 6 | UUT GND |
| 12 | UUT GND | Group B Channel 6 | UUT GND | Group D Channel 6 |
| 13 | Group A Channel 7 | UUT GND | Group C Channel 7 | UUT GND |
| 14 | UUT GND | Group B Channel 7 | UUT GND | Group D Channel 7 |
| 15 | Group A Channel 8 | UUT GND | Group C Channel 8 | UUT GND |
| 16 | UUT GND | Group B Channel 8 | UUT GND | Group D Channel 8 |
| 17 | Group A Channel 9 | UUT GND | Group C Channel 9 | UUT GND |
| 18 | UUT GND | Group B Channel 9 | UUT GND | Group D Channel 9 |
| 19 | Group A Channel 10 | UUT GND | Group C Channel 10 | UUT GND |
| 20 | UUT GND | Group B Channel 10 | UUT GND | Group D Channel 10 |
| 21 | Group A Channel 11 | UUT GND | Group C Channel 11 | UUT GND |
| 22 | UUT GND | Group B Channel 11 | UUT GND | Group D Channel 11 |
| 23 | Group A Channel 12 | UUT GND | Group C Channel 12 | UUT GND |
| 24 | UUT GND | Group B Channel 12 | UUT GND | Group D Channel 12 |
| 25 | I/O0 | I/O1 | I/O2 | I/O3 |
| 26 | CHASSIS | CHASSIS | CHASSIS | CHASSIS |
| 27 | Group E Channel 1 | UUT GND | Group G Channel 1 | UUT GND |
| 28 | UUT GND | Group F Channel 1 | UUT GND | Group H Channel 1 |
| 29 | Group E Channel 2 | UUT GND | Group G Channel 2 | UUT GND |
| 30 | UUT GND | Group F Channel 2 | UUT GND | Group H Channel 2 |
| 31 | Group E Channel 3 | UUT GND | Group G Channel 3 | UUT GND |
| 32 | UUT GND | Group F Channel 3 | UUT GND | Group H Channel 3 |
| 33 | Group E Channel 4 | UUT GND | Group G Channel 4 | UUT GND |

| | | | | |
|---|---|---|---|---|
| 34 | UUT GND | Group F Channel 4 | UUT GND | Group H Channel 4 |
| 35 | Group E Channel 5 | UUT GND | Group G Channel 5 | UUT GND |
| 36 | UUT GND | Group F Channel 5 | UUT GND | Group H Channel 5 |
| 37 | Group E Channel 6 | UUT GND | Group G Channel 6 | UUT GND |
| 38 | UUT GND | Group F Channel 6 | UUT GND | Group H Channel 6 |
| 39 | Group E Channel 7 | UUT GND | Group G Channel 7 | UUT GND |
| 40 | UUT GND | Group F Channel 7 | UUT GND | Group H Channel 7 |
| 41 | Group E Channel 8 | UUT GND | Group G Channel 8 | UUT GND |
| 42 | UUT GND | Group F Channel 8 | UUT GND | Group H Channel 8 |
| 43 | Group E Channel 9 | UUT GND | Group G Channel 9 | UUT GND |
| 44 | UUT GND | Group F Channel 9 | UUT GND | Group H Channel 9 |
| 45 | Group E Channel 10 | UUT GND | Group G Channel 10 | UUT GND |
| 46 | UUT GND | Group F Channel 10 | UUT GND | Group H Channel 10 |
| 47 | Group E Channel 11 | UUT GND | Group G Channel 11 | UUT GND |
| 48 | UUT GND | Group F Channel 11 | UUT GND | Group H Channel 11 |
| 49 | Group E Channel 12 | UUT GND | Group G Channel 12 | UUT GND |
| 50 | UUT GND | Group F Channel 12 | UUT GND | Group H Channel 12 |

**Table 3-4: GX6192 Connector J2**

The following table describes the signals mentioned in Table 3-4:

| **Signal name** | **Comments** |
|---|---|
| Group X Channel Y | Mux Group channels for group A through H |
| UUT GND | Unit Under Test ground. This ground is isolated from power ground and chassis. |
| I/O X | PIO (Static general purpose I/O) pins 0 to 3 |

## GX6192 J3 Connector

The following table describes the J3 (200 pin) connector's pin out:

| Pin | Column A | Column B | Column C | Column D |
|---|---|---|---|---|
| 1 | Group I Channel 1 | UUT GND | Group K Channel 1 | UUT GND |
| 2 | UUT GND | Group J Channel 1 | UUT GND | Group L Channel 1 |
| 3 | Group I Channel 2 | UUT GND | Group K Channel 2 | UUT GND |
| 4 | UUT GND | Group J Channel 2 | UUT GND | Group L Channel 2 |
| 5 | Group I Channel 3 | UUT GND | Group K Channel 3 | UUT GND |
| 6 | UUT GND | Group J Channel 3 | UUT GND | Group L Channel 3 |
| 7 | Group I Channel 4 | UUT GND | Group K Channel 4 | UUT GND |
| 8 | UUT GND | Group J Channel 4 | UUT GND | Group L Channel 4 |
| 9 | Group I Channel 5 | UUT GND | Group K Channel 5 | UUT GND |
| 10 | UUT GND | Group J Channel 5 | UUT GND | Group L Channel 5 |
| 11 | Group I Channel 6 | UUT GND | Group K Channel 6 | UUT GND |
| 12 | UUT GND | Group J Channel 6 | UUT GND | Group L Channel 6 |
| 13 | Group I Channel 7 | UUT GND | Group K Channel 7 | UUT GND |
| 14 | UUT GND | Group J Channel 7 | UUT GND | Group L Channel 7 |
| 15 | Group I Channel 8 | UUT GND | Group K Channel 8 | UUT GND |
| 16 | UUT GND | Group J Channel 8 | UUT GND | Group L Channel 8 |
| 17 | Group I Channel 9 | UUT GND | Group K Channel 9 | UUT GND |
| 18 | UUT GND | Group J Channel 9 | UUT GND | Group L Channel 9 |
| 19 | Group I Channel 10 | UUT GND | Group K Channel 10 | UUT GND |
| 20 | UUT GND | Group J Channel 10 | UUT GND | Group L Channel 10 |
| 21 | Group I Channel 11 | UUT GND | Group K Channel 11 | UUT GND |
| 22 | UUT GND | Group J Channel 11 | UUT GND | Group L Channel 11 |
| 23 | Group I Channel 12 | UUT GND | Group K Channel 12 | UUT GND |
| 24 | UUT GND | Group J Channel 12 | UUT GND | Group L Channel 12 |
| 25 | I/O0 | I/O0 | I/O0 | I/O0 |
| 26 | CHASSIS | CHASSIS | CHASSIS | CHASSIS |
| 27 | Group M Channel 1 | UUT GND | Group O Channel 1 | UUT GND |
| 28 | UUT GND | Group N Channel 1 | UUT GND | Group P Channel 1 |
| 29 | Group M Channel 2 | UUT GND | Group O Channel 2 | UUT GND |
| 30 | UUT GND | Group N Channel 2 | UUT GND | Group P Channel 2 |
| 31 | Group M Channel 3 | UUT GND | Group O Channel 3 | UUT GND |
| 32 | UUT GND | Group N Channel 3 | UUT GND | Group P Channel 3 |

| | | | | |
|---|---|---|---|---|
| 33 | Group M Channel 4 | UUT GND | Group O Channel 4 | UUT GND |
| 34 | UUT GND | Group N Channel 4 | UUT GND | Group P Channel 4 |
| 35 | Group M Channel 5 | UUT GND | Group O Channel 5 | UUT GND |
| 36 | UUT GND | Group N Channel 5 | UUT GND | Group P Channel 5 |
| 37 | Group M Channel 6 | UUT GND | Group O Channel 6 | UUT GND |
| 38 | UUT GND | Group N Channel 6 | UUT GND | Group P Channel 6 |
| 39 | Group M Channel 7 | UUT GND | Group O Channel 7 | UUT GND |
| 40 | UUT GND | Group N Channel 7 | UUT GND | Group P Channel 7 |
| 41 | Group M Channel 8 | UUT GND | Group O Channel 8 | UUT GND |
| 42 | UUT GND | Group N Channel 8 | UUT GND | Group P Channel 8 |
| 43 | Group M Channel 9 | UUT GND | Group O Channel 9 | UUT GND |
| 44 | UUT GND | Group N Channel 9 | UUT GND | Group P Channel 9 |
| 45 | Group M Channel 10 | UUT GND | Group O Channel 10 | UUT GND |
| 46 | UUT GND | Group N Channel 10 | UUT GND | Group P Channel 10 |
| 47 | Group M Channel 11 | UUT GND | Group O Channel 11 | UUT GND |
| 48 | UUT GND | Group N Channel 11 | UUT GND | Group P Channel 11 |
| 49 | Group M Channel 12 | UUT GND | Group O Channel 12 | UUT GND |
| 50 | UUT GND | Group N Channel 12 | UUT GND | Group P Channel 12 |

**Table 3-5: GX6192 Connector J3**

The following table describes the signals mentioned in Table 3-5:

| Signal name | Comments |
|---|---|
| Group X Channel Y | Mux Group channels for group I through P |
| UUT GND | Unit Under Test ground. This ground is isolated from power ground and chassis. |
| I/O X | PIO (Static general purpose I/O) pins 4 to 7 |

## GX6192 P1 Connector

The following table describes the P1 Analog Input (8 pin) connector's pin out:

| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | Analog Input 15 | 2 | Analog Input 13 |
| 3 | Analog Input 11 | 4 | Analog Input 9 |
| 5 | Analog Input 7 | 6 | Analog Input 5 |
| 7 | Analog Input 3 | 8 | Analog Input 1 |

**Table 3-6: GX6192 Connector P1**

## GX6192 P2 Connector

The following table describes the P2 Analog Input (8 pin) connector's pin out:

| Pin | Function | Pin | Function |
|---|---|---|---|
| 1 | Analog Input 2 | 2 | Analog Input 4 |
| 3 | Analog Input 6 | 4 | Analog Input 8 |
| 5 | Analog Input 10 | 6 | Analog Input 12 |
| 7 | Analog Input 14 | 8 | Analog Input 16 |

**Table 3-7: GX6192 Connector P2**

# Chapter 4 - SwitchEasy Software

SwitchEasy is a software platform based on ATEasy that allows users to control and configure the GX7016 Switching Subsystem through an ATEasy driver commands providing pin-mapping, rule based switching and higher level operation of the GX7016 switching subsystem.

## SwitchEasy Overview

SwitchEasy is an ATEasy software driver provided with the TS-323 test system (GENASYS) that controls the connections and disconnections of the UUT signals to and from the Test Station equipment. Signal path connections, can be automatically chosen by the program or routed per user preference, must comply with a predefined set of rules, and provides repeatable paths configurations. The program supports UUT pin (name or user supplied ID) to MacPanel pin mapping and includes panels to provide the user with additional switching path information.

SwitchEasy is unique to TS-323 and only supports TS-323 switching architecture and topologies. Support for other switching architectures is not planned at this time (although it may be added in the future).

The SwitchEasy software is not part of the GXSW software and is installed as part of the TS-323/GENASYS test system software.

## SwitchEasy Panel

The SwitchEasy Panel is designed to work as a standalone application that displays the switching subsystem's current status. The panel lists the instrument pins, UUT pins, triggers, and busses that are currently in use by the subsystem.

### Status Page

The Status page displays the current state of connections between instrument resources and pins (UUT/Mac panel). This page also shows the global bus resources that are being used to facilitate connections.

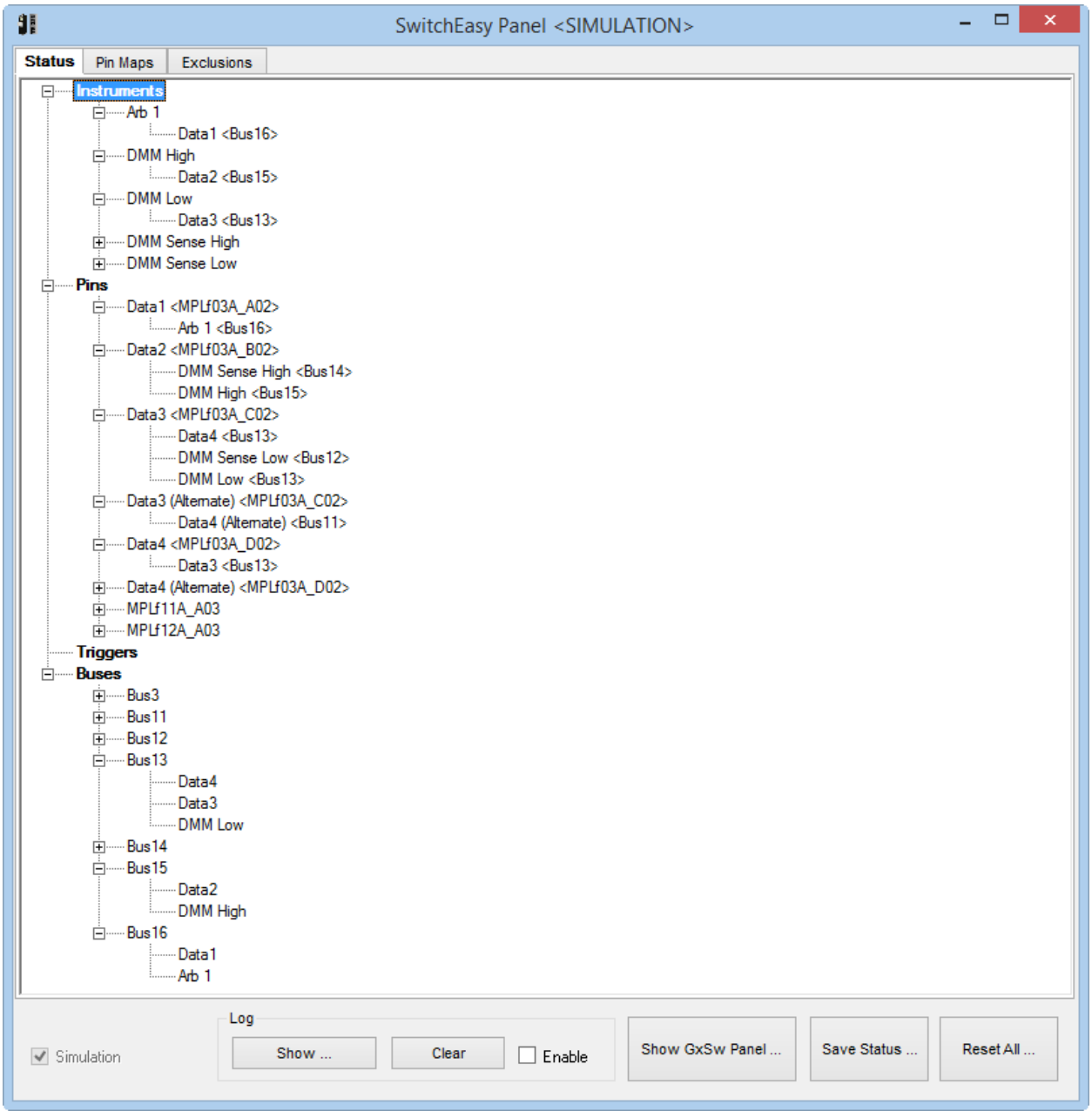


**Figure 4-1: Switch Easy Panel Status Page**

**Status (Tree)**: The tree view shows currently made connections from the perspective of Instruments, Pins, Triggers, and Buses. The Instruments node expands to show what UUT/MacPanel pin are connected to each instrument resource. The Pins node expands to show what each UUT/MacPanel pin is connected to. For example, in Figure 4-1, under the Pins node, the Data1 pin is shown being connected to the Arb 1 instrument resource. The same information is mirrored under the Instruments node, where the Arb 1 instrument resource is shown connected to the Data 1 pin.

**Log Group Box**

**Show... (Button):** This button will launch the Windows event log which will show a log of Switch Easy events that have occurred.

**Clear (Button):** This button will clear Switch Easy events from the Windows event log.

**Enable (Checkbox):** This checkbox will enable or disable logging of Switch Easy events to the Windows event log.

**Show GxSw Panel... (Button):** This button will launch the GxSw low level panel.

**Save Status... (Button):** This button will save the connection state of Switch Easy to a file

**Reset All... (Button):** This button will reset all connections (disconnect everything).

### Pin Maps Page

The Pins Maps page displays all pin mappings. The physical location of the pin (slot, tier, column, and row) are shown along with the mapped name.

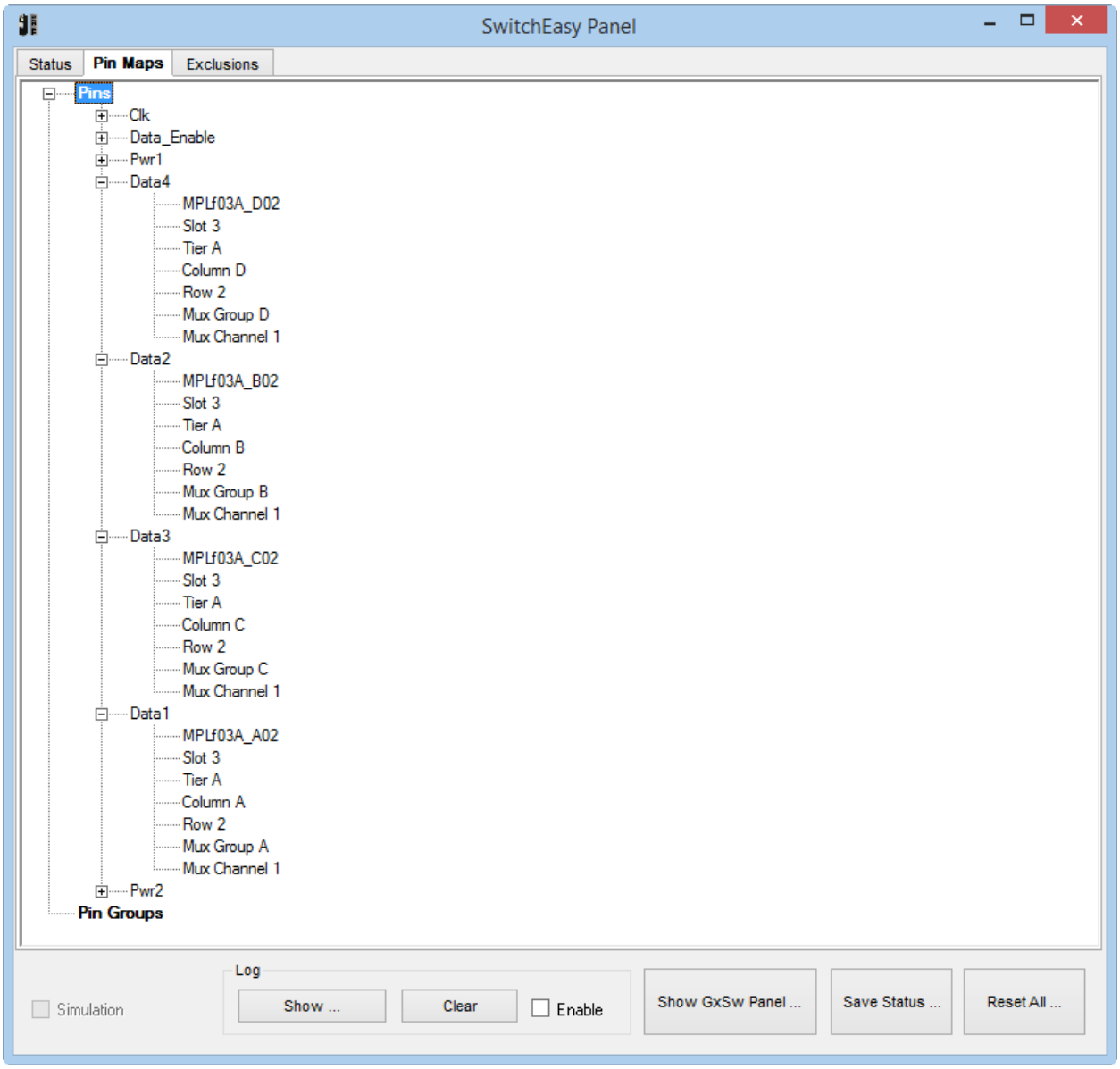


**Figure 4-2: Switch Easy Panel Pin Maps Page**

**Pin Maps (Tree)**: The tree view shows currently mapped pins. Each mapped name node expands to show the physical location and description of the pin. For example, in Figure 4-2, the **Data 4** node is expanded to show its location at Slot 3, Tier A, Column D, and Row 2. Mux group (D) and Mux Channel (1) are also displayed to provide the user more information about the physical pin.

### Exclusions Page

The Exclusions page allows the user to select which instrument resources, pins, or global bus resources will be excluded from making connections. Any attempt to connect with an excluded instrument, pin or global bus will result in a run-time error.

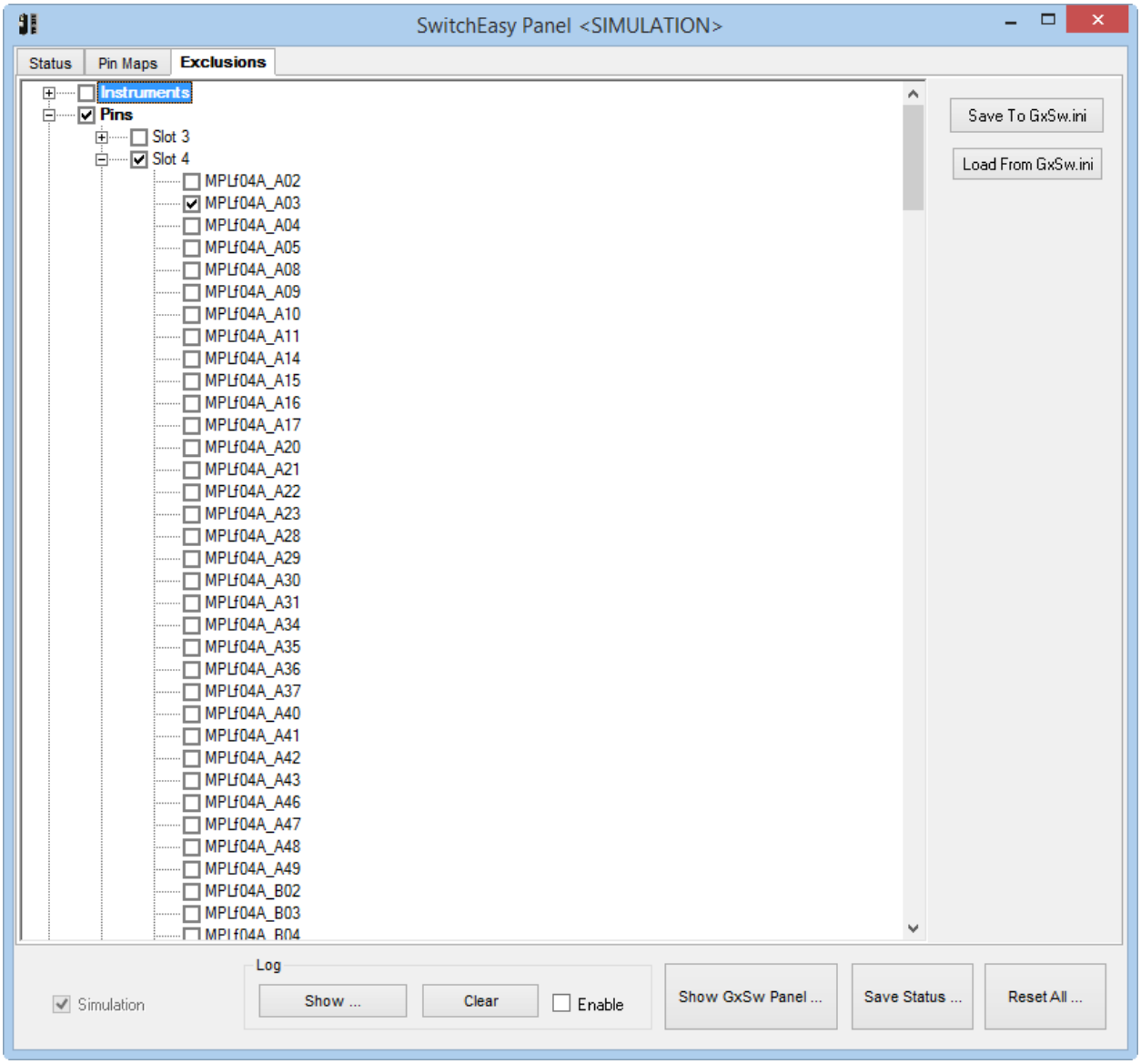


**Figure 4-3: Switch Easy Panel Exclusions Page**

**Exclusions (Tree)**: The tree view shows all instruments, pins, triggers, and buses that are part of the switching subsystem. A marked checkbox indicates that the corresponding resource cannot be connected. For example, in Figure 4-3, the **MPLf04A_A02** pin has been checked. This indicates that this particular pin will not be allowed to make a connection to any instrument resource, pin, trigger, or bus.

**Save To GxSw.ini (Button):** This button allows the user to save the exclusion rules to the GxSw.ini file.

**Load From GxSw.ini (Button):** This button loads the previously saved exclusions rules from the GxSw.ini file to Switch Easy.

## SwitchEasy Base Commands

The following is a list of the SwitchEasy base commands:

**Connect** - These commands establishes a connection between an instrument and a MacPanel pin

**Disconnect** - These commands breaks a connection between an instrument and a MacPanel pin

**Reset** - Force all relays to an open state

**Setup** - Configures SwitchEasy (Instrument/Pin mapping, windows logging, trigger)

**Panel** - Starts the SwitchEasy panel

**Utility** - Configure and use the built in test, interrupts, and the PIO

## SwitchEasy Command Reference

| Command | Description |
|---|---|
| **Connect/Disconnect** | |
| **Connect Instrument Differential** | Connects Differential Instrument to two Pins using optional bus and rules |
| **Connect Instrument Instrument** | Connects an Instrument to an Instrument |
| **Connect Instrument MpPin** | Connect Instrument Pin to MacPanel Pin |
| **Connect Instrument Pin** | Connect Instrument Pin to UUT Pin |
| **Connect Pin Pin** | Connect UUT Pin to UUT Pin |
| **Connect Pin MpPin** | Connect UUT Pin to MacPanel Pin |
| **Connect MpPin MpPin** | Connect MacPanel Pin to MacPanel Pin |
| **Connect Trigger Instrument** | Connect Trigger to Instrument |
| **Connect Trigger MpPin** | Connect Trigger to MacPanel pin |
| **Connect Trigger Pin** | Connect Trigger to UUT Pin |
| **Connect Trigger Trigger** | Connect Trigger to Trigger |
| **Connect Digital Pin** | Connect UUT Pin to digital line |
| **Connect Digital MpPin** | Connect MacPanel Pin to digital line |
| **Disconnect Bus** | Disconnect Bus from All, Instrument or Pin |
| **Disconnect Digital Pin** | Disconnect UUT Pin from digital line |
| **Disconnect Digital MpPin** | Disconnect MacPanel Pin from digital line |
| **Disconnect Instrument Instrument** | Disconnect Instrument from another instrument |
| **Disconnect Instrument MpPin** | Disconnect Instrument from MacPanel pin |
| **Disconnect Instrument Pin** | Disconnect Instrument from UUT pin |
| **Disconnect MpPin MpPin** | Disconnect MacPanel pin from another MacPanel pin |
| **Disconnect Pin Pin** | Disconnect UUT pin from another UUT pin |
| **Disconnect Pin MpPin** | Disconnect UUT Pin from MacPanel pin |
| **Disconnect Trigger Instrument** | Disconnect trigger from instrument |
| **Disconnect Trigger MpPin** | Disconnect Trigger from MacPanel pin |

| Command | Description |
|---|---|
| **Disconnect Trigger Pin** | Disconnect Trigger pin from UUT pin |
| **Disconnect Trigger Trigger** | Disconnect Trigger from Trigger |
| **Reset All** | Reset all boards (LF and HF) along with any RTMs and SRs. |
| **Setup** | |
| **Setup Instrument Name** | Set instrument name |
| **Setup Instrument PreferredBus** | Sets a preferred bus to be used in connecting to a particular instrument |
| **Setup Instrument Map** | Maps instrument to a specific input channel (located on the RTM for an LF board, or on the HF board itself)<br>The instrument name is also set in this command. |
| **Setup Log Clear** | Clears the SwitchEasy Windows event log |
| **Setup Log On** | Turns the SwitchEasy Windows event logging on |
| **Setup Log Off** | Turns the SwitchEasy Windows event logging off |
| **Setup Pin Group** | Maps UUT pins to a UUT pin group |
| **Setup Pin Map** | Maps UUT pin to a specific MacPanel pin |
| **Setup Pin Remove** | Remove UUT or MacPanel pin from mapping |
| **Setup State MpPin Analog** | Sets the pin state of a MacPanel pin to analog |
| **Setup State MpPin Digital** | Sets the pin state of a MacPanel pin to digital |
| **Setup State MpPin Hybrid** | Sets the pin state of a MacPanel pin to hybrid (analog and digital) |
| **Setup State MpPin Null** | Sets the pin state of a MacPanel pin to none |
| **Setup State Pin Analog** | Sets the pin state of a UUT pin or pin group to analog |
| **Setup State Pin Digital** | Sets the pin state of a UUT pin or pin group to digital |
| **Setup State Pin Hybrid** | Sets the pin state of a UUT pin or pin group to hybrid |
| **Setup State Pin Null** | Sets the pin state of a UUT pin or pin group to none |
| **Setup Trigger Name** | Set trigger name |
| **Setup Trigger Map** | Maps trigger to a specific input channel (located on the RTM)<br>The trigger name is also set in this command. |
| **Setup Dio Connection** | Maps a group of 16 DIO channels to a specific LF board |
| **Get** | |
| **Get Instrument Name** | Get instrument's name |
| **Get Instrument PreferredBus** | Get instrument's preferred bus |
| **Get MpPin Count Lf** | Return number of LF channels |
| **Get MpPin Count Hf** | Return number of HF channels |
| **Get MpPin Column** | Returns the MacPanel column (A, B, C, or D) that is associated with a specified MacPanel pin |
| **Get MpPin Tier** | Returns the MacPanel Tier (A or B) that is associated with a specified MacPanel pin |
| **Get MpPin DioChannel** | Returns the DIO channel number (0-based) that is mapped to a MacPanel |

| Command | Description |
|---|---|
| | pin |
| **Get MpPin From Name** | Returns MacPanel pin from a given pin name |
| **Get MpPin From Position** | Returns MacPanel pin from physical location specified by MacPanel Slot, Tier, Column, and Row |
| **Get MpPin From MuxPosition** | Returns MacPanel pin from mux location specified by slot, mux group (A-P), and mux channel (1-16) |
| **Get MpPin Name** | Returns the MacPanel Pin Name |
| **Get MpPin Row** | Returns the MacPanel Row location for a given MacPanel pin |
| **Get MpPin Type** | Returns a MacPanel's slot type (LF or HF) |
| **Get MpPin MuxGroup** | Return Mux Group for specified MacPanel pin |
| **Get MpPin MuxChannel** | Return Mux Channel for specified MacPanel pin |
| **Get Pin Count** | Returns number of UUT pins. |
| **Get Pin DioChannel** | Returns the DIO channel mapped to a specified UUT pin |
| **Get Pin IsGroup** | Returns True if the specified pin is a UUT pin group. |
| **Get Pin MpPin** | Returns the MacPanel pin mapped to the specified UUT pin |
| **Get Pin MpSlot** | Returns the MacPanel slot mapped to the specified UUT pin |
| **Get Pin Name** | Returns name associated with UUT pin or group |
| **Get Pin Type** | Returns a UUT pin slot type (LF or HF) |
| **Get Pin Group Count** | Returns number of UUT pin groups. |
| **Get Pin Group Pins** | Returns the array of children UUT pins. |
| **Get Pin Group PinsCount** | Returns number of children UUT pins. |
| **Get Pin Group NextPin** | Return next UUT pin is available, otherwise returns empty string. |
| **Get Pin Group Pin** | Returns name or id of the specified child UUT pin given the pin index |
| **Get Relay Count** | Returns the total number of relay openings/closures that have occurred for a specified block (board, RTM or SR)<br>The count for each relay is accumulated throughout the life of a board |
| **Get Relay States** | Returns the state of each relay of a specified block (board, RTM or SR)<br>The index of an array element specifies its relay designation minus 1.<br>For example, index 0 corresponds with relay K1, index 256 corresponds with relay K257. |
| **Get Trigger Name** | Returns trigger pin name |
| **Get Connections State** | Returns the current connection status of the switching system<br>All types or a type (instrument, trigger, or pin) of connection can be specified |
| **Get Slot Information** | Return True if SW card is installed at the specified slot, also return slot type LF/HF, and whether RTM and SR are present. |
| **Panel** | Shows the Switch Easy Panel |
| **Utility** | |

| Command | Description |
|---|---|
| **Utility BIT Abort** | Aborts the Built In Test |
| **Utility BIT GetStatus** | Returns Built in Test Status |
| **Utility BIT Run** | Starts a Built In Test |
| **Utility Interrupt Disable** | Disable PCI interrupt. |
| **Utility Interrupt Resume** | Resume PCI interrupt. |
| **Utility Interrupt Reset** | Reset interrupt source. |
| **Utility Interrupt Setup** | Configures the behavior of the interrupts. |
| **Utility PIO Set Output Data** | Set the PIO Output Data |
| **Utility PIO Set Output Enable** | Set the PIO Output Enable |
| **Utility PIO Get Input Data** | Reads the PIO Input |
| **Utility PIO Get Output Data** | Returns the PIO Output Data |
| **Utility PIO Get Output Enable** | Returns the PIO Output Enable |

**Table 4-1: Switch Easy driver commands**

## SwitchEasy Connect/Disconnect Commands

From the Connect/Disconnect commands the following connections types can be made:

| | |
|---|---|
| Instrument | Test Station Instruments, includes Digital Multimeter (DMM), Source Measurement Unit (SMU), Digital Storage Oscilloscope (Scope), Arbitrary Waveform Generator (Arb), … . Supported Test Station Instruments connections are enumerated under enumSeInstruments. |
| Pin | Pin is the equivalent of a named MpPin/Slot, users must define Pin characteristics before using (see Setup/Pin/Map) |
| MpPin | MacPanel Pin of the Gx6256 (LF) and Gx6192 (HF). Switchable MpPin are enumerated under enumSeMpPin. |
| Trigger | Some of the Test Station Instruments have connections for indicating an event or starting a process. Supported Trigger interfaces are enumerated under enumSeTrig. |
| Digital | Digital connections are from the Gx5964 (DIO). |

**Table 4-2: Connect / Disconnect Commands, Connections Types**

Once the "to/from" connections types are defined, specific connections details can be added.

Next, the choice of internal bus(es) can be defined (Bus 0 to Bus 15 or default). Note "default" allows Switch Easy to choose the Bus.

Finally, the choice of routing rules can be defined (note switching rules can be ORed). The choices are:

**aSeRuleDefaultDefault**: No multiple connections, no hybrid connections, no auto-disconnect, bleed on disconnect, no bleed on disconnect, use main channel

**aSeRuleAllowMuxMultiConnect**: Allow multiple channels within the same mux group

**aSeRuleAllowMultiConnect**: Allow multiple items (Pin, Instrument, Trigger) connections

**aSeRuleHybrid**: Allow digital and analog connections together

**aSeRuleAlternate**: Connection to Alternate channel (LF pins only)

**aSeRuleAlternate2**: For commands with two pins, Applied to second pin, connection to Alternate channel (LF pins only)

**aSeRuleConnectBleedInhibit**: Disable bleeding during connection

**aSeRuleDisconnectBleedEnable**: Enable bleeding on disconnection

**aSeRuleAutoDisconnectMpPin**: Auto disconnect connected MacPanel pin before connect to a new MacPanel pin

**aSeRuleAllowConnectMatrixInput**: Allow connect matrix input when connecting HF pins from different tiers within the same board.

#### SwitchEasy – Reset

The command **SwitchExec Reset All()** will force all relays to an open state, thereby disconnecting all signal paths.

## SwitchEasy Commands Conventions

Use Enum for MacPanel (Slot, Pin), buses, instruments, etc

Use UUT Pin name (string) or ID (user supplied enum constant) mapped to MacPanel Pin for commands

Use of optional Parameters with default values to avoid entering all parameters

## Programming Examples

### Mapping pins and pin groups

Switch Easy allows users to assign names to physical pins. These names can later be used in place of the physical pin designation when calling commands that make or break connections. Switch Easy also allows users to define pin groups, which are collections of assigned pin names.

The following is an example of assigning the name **Data1**, **Data2**, **Data3**, and **Data4** to the physical Mac Panel pins located at Slot 3, Tier A, Row2, Column A, B, C, and D respectively:

```
SwitchExec Setup Pin Map("Data1", 0, aSeMpSlot3, aSeLfA_A02) ! Slot3 Tier A, Column A, Row 2
SwitchExec Setup Pin Map("Data2", 1, aSeMpSlot3, aSeLfA_B02) ! Slot3 Tier A, Column B, Row 2
SwitchExec Setup Pin Map("Data3", 2, aSeMpSlot3, aSeLfA_C02) ! Slot3 Tier A, Column C, Row 2
SwitchExec Setup Pin Map("Data4", 3, aSeMpSlot3, aSeLfA_D02) ! Slot3 Tier A, Column D, Row 2
```

Note that the second (optional) parameter is the unique numerical ID that is associated with the particular mapping.

The following is an example of creating a pin group called **Data Pins** that contains **Data1**, **Data2**, **Data3**, and **Data4**.

```
SwitchExec Setup Pin Group("Data Pins", {"Data1", "Data2", "Data3", "Data4"})
```

The name **Data Pins** can now be used in other commands to reference the group of four pins.

### Making a connection

Switch Easy provides the user with the ability make connections between pins and instruments. The connection commands use a mapped name or a physical location (slot, tier, column and row) to specify a pin. The pin is referred to as a UUT pin when using a mapped name, and a Mac Panel pin when using a physical location.

Switch Easy can make the following types of connections:

#### Instrument to UUT pin

A predefined instrument resource is connected to a pin which has been previously mapped to a name.

The following is an example of making a connection between the instrument named **Arb1** and a mapped pin named **Data1**:

```
SwitchExec Connect Instrument Pin(aSeInstrumentArb1, "Data1")
```

#### Instrument to Mac Panel pin

A predefined instrument resource is connected to a pin using its phyiscal location (slot, tier, column, and row).

The following is an example of making a connection between the instrument named **Arb1** and the pin phyiscally located in slot 3, on Tier A, Column A, Row 2:

```
SwitchExec Connect Instrument MpPin(aSeInstrumentArb1, aSeMpSlot3, aSeLfA_A02, ,
aSeRuleAllowMultiConnect)
```

Note that this particular example specifies a rule (allow multiple connections) when making the connection.

#### Multiple Instruments to UUT pin

Multiple instrument resources can be connected to the same pin concurrently. This can be accomplished when using the Allow Multiple Connections rule.

The following is an example of creating connections required for a 4-wire measurement resistance measurement:

```
SwitchExec Connect Instrument Pin(aSeInstrumentDmmHi, "Data2")
SwitchExec Connect Instrument Pin(aSeInstrumentDmmSenseHi, "Data2",, aSeRuleAllowMultiConnect)
SwitchExec Connect Instrument Pin(aSeInstrumentDmmLo, "Data3")
SwitchExec Connect Instrument Pin(aSeInstrumentDmmSenseLo, "Data3",, aSeRuleAllowMultiConnect)
```

The instrument resources **DmmHi**, and **DmmSensesHi** are both connected to the **Data2** pin. The instrument resources **DmmLo** and **DmmSenseLo** are both connected to the **Data2** pin.

#### Differential Instrument to Mac Panel pin

Instruments that use differential channels (such as a DMM) can be connected to two pins using only one command.

The following is an example of connecting the DMM's differential inputs (DmmHi and DmmLo) to the pin located at Slot 10, Tier A, ColumnA, Row 2, and Slot 10, Tier A, Column B, Row 2:

```
SwitchExec Connect Instrument Differential(aSeInstrumentDmm, aSeMPSlot10, aSeLfA_A02,
aSeMPSlot10, aSeLfA_B02)
```

#### UUT pin to UUT pin

One pin can be connected to another pin, without having to connect to an instrument resource.

The following is an example of connecting the **Data4** pin to the **Data3** pin:

```
SwitchExec Connect Pin Pin("Data4", "Data3", , aSeRuleAllowMultiConnect)
```

Note that the Allow Multiple Connections rule must be specified when connecting one pin to another.

#### Mac Panel pin to Mac Panel pin

One pin can be connected to another pin, without having to connect to an instrument resource.

The following is an example of connecting the pin located at Slot 11, Tier A, Column A, Row 3 to the pin located at Slot 12, Tier A, Column A, Row 3:

```
SwitchExec Connect MpPin MpPin(aSeMPSlot11, aSeLfA_A03, aSeMpSlot12, aSeLfA_A03, aSeBus3)
```

### Disconnecting

Switch Easy allows the user to disconnect pins from instruments, and instruments from all connections.

The following is an example of disconnecting the **Data1** pin from the instrument resrouce **Arb1:**

```
SwitchExec Disconnect Instrument Pin(aSeInstrumentArb1, "Data1")
```

The following is an example of disconnecting the **Arb1** instrument resource from all pins:

```
SwitchExec Disconnect Instrument(aSeInstrumentArb1)
```

All connections can be removed while still maintaining the current pin mapping by using the following command:

```
SwitchExec Reset All()
```

# Chapter 5 - Function Reference

## Introduction

The GX7016 switching subsystem function reference includes low level GX6256/GX6192/RTM/SR functions used to program the digital subsystem components. User should use the SwitchEasy ATEasy command reference to use the digital subsystem. The functions are prefix with the card type they are addressing Gx6256 or Gx6192 followed by the main components that they are addressing (SR for switch router, etc.). The function reference describes the Gx256XXX function similar set is available to the Gx6192XXX and was not included in this chapter since it is identical. The 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 required for operating the board and is returned by the **Gx6256Initialize** or the Gx6256InitializeVisa 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 GX6256.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 5-1: Parameter Prefixes**

## GX6256/GX6192/GX6009/GX6032 Functions

The following list is a summary of functions available for the GX6256/GX6192/GX6009/GX6032:

| Driver Functions | Description |
|---|---|
| **Gx6256BITAbort** | Abort currently running BIT test |
| **Gx6256BITGetStatus** | Returns BIT status |
| **Gx6256BITRun** | Starts BIT test |
| **Gx6256BleedRelayClose** | Closes a mux group's bleed relay |
| **Gx6256BleedRelayGetState** | Returns the state of a mux group's bleed relay |
| **Gx6256BleedRelayOpen** | Opens a mux group's bleed relay |
| **Gx6256GetBoardSummary** | Returns the board summary. |
| **Gx6256GetBoardType** | Returns the board type. |
| **Gx6256GetRelayCount** | Returns the number of relays on a device (board, RTM, or SR) |
| **Gx6256GetRelayStates** | Returns the close/open states of all relays on a device (board, RTM, or SR) |
| **Gx6256Initialize** | Initializes the driver for the board at the specified slot number. The function returns a handle that can be used with other GX6256 functions to program the board |
| **Gx6256InitializeVisa** | Initializes the driver for the specified slot using VISA. The function returns a handle that can be used with other GX6256 functions to program the board. |
| **Gx6256MatrixClose** | Closes a matrix relay |
| **Gx6256MatrixGetChannel** | Returns a matrix relay's state |
| **Gx6256MatrixOpen** | Opens a matrix relay |
| **Gx6256MatrixReset** | Resets (opens) all relays in the matrix |
| **Gx6256MatrixInputSelectorClose** | Closes a matrix selector relay |
| **Gx6256MatrixInputSelectorGetChannel** | Returns a matrix selector relay's state |
| **Gx6256MatrixInputSelectorOpen** | Opens a matrix selector relay |
| **Gx6256MuxClose** | Close a mux relay |
| **Gx6256MuxGetChannel** | Returns a mux relay's state |
| **Gx6256MuxOpen** | Open a mux relay |
| **Gx6256MuxReset** | Resets (opens) all relays in a mux groups |
| **Gx6256MuxGetSelectorMode** | Returns the mux selector mode for a mux group |
| **Gx6256MuxSetSelectorMode** | Sets the mux selector mode for a mux group |
| **Gx6256Panel** | Opens a virtual panel used to interactively control the GX6256 board. |
| **Gx6256PioGetOutputData** | Returns PIO output data |
| **Gx6256PioGetOutputEnable** | Returns PIO output enable (tri-state) data |
| **Gx6256PioReadInput** | Returns PIO input data |

| Driver Functions | Description |
|---|---|
| **Gx6256PioSetOutputData** | Set PIO output data |
| **Gx6256PioSetOutputEnable** | Set PIO output enable (tri-state) data |
| **Gx6256Reset** | Opens all the board relays. |
| **Gx6256RtmMatrixClose** | Closes a RTM matrix relay |
| **Gx6256RtmMatrixGetChannel** | Returns a RTM matrix relay's state |
| **Gx6256RtmMatrixOpen** | Opens a RTM matrix relay |
| **Gx6256SrMatrixClose** | Closes a SR bus relay |
| **Gx6256SrMatrixGetChannel** | Returns a SR bus relay's state |
| **Gx6256SrMatrixOpen** | Opens a SR bus relay |
| **GxSWGetErrorString** | Returns the error string associated with the specified error number. |

## Gx6256BITAbort

### Purpose

Aborts a running built in test.

### Syntax

**Gx6256BITAbort** (*nHandle, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The built in test checks each relay in a specified device for continuity. This function aborts a test that is currently running.

### Example

The following example runs the built in test, waits one second, and then aborts the test:

```
Gx6256Initilize (10, FALSE, &nHandle, &nStatus);
Gx6256BITRun (nHandle, GX6256_BIT_MODE_ALL, &nStatus);
sleep(1000);
Gx6256BITAbort (nHandle, &nStatus);
```

### See Also

**Gx6256BITGetStatus, Gx6256BITRun**, **GxSWGetErrorString**

## Gx6256BITGetStatus

### Purpose

Aborts a running built in test.

### Syntax

**Gx6256BITGetStatus** (*nHandle, pbPass, pszBITResult, nResultMaxLen, pdPercentCompleted, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| pbPass | PBOOL | Returns with the pass/fail status of the built in test. TRUE is pass and FALSE is fail. |
| pszBITResult | PSTR | Returned string containing the built in test result. |
| nResultMaxLen | SHORT | Size of the buffer to contain the BIT result. |
| pdPercentCompleted | PDOUBLE | Return the percentage completed |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The built in test checks each relay in a specified device for continuity. This function returns the percent completed while the test is running. When the test has completed, this function returns the result of the test in the form of a string (**pszBITResult**) and a flag (**pbPass**) that indicates pass or fail.

### Example

The following example runs the built in test and then polls the built in test status until it has completed:

```
BOOL  bPass;
CHAR  szBITResult[1024];
DOUBLE dPercentCompleted;

Gx6256BITRun (nHandle, GX6256_BIT_MODE_ALL, &nStatus);
Gx6256BITGetStatus (nHandle, &bPass, szBITResult, 1024, &dPercentCompleted, &nStatus);
while (dPercentCompleted<100)
{
    printf("BIT Percent Completed = %f %", dPercentCompleted);
    Gx6256BITGetStatus (nHandle, &bPass, szBITResult, 1024, &dPercentCompleted, &nStatus);
}
```

### See Also

**Gx6256BITGetStatus, Gx6256BITRun**, **GxSWGetErrorString**

## Gx6256BITRun

### Purpose

Aborts a running built in test.

### Syntax

**Gx6256BITRun** (*nHandle, nMode, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMode | SHORT | Selects which BIT test<br>0. GX6256_BIT_MODE_MUX: Test Mux relays only<br>1. GX6256_BIT_MODE_MATRIX: Test Matrix relays only<br>2. GX6256_BIT_MODE_RTM: Test Rear Transition Module relays only (if one is connected)<br>3. GX6256_BIT_MODE_ALL: Test all relays |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The built in test checks each relay in a specified device for continuity. When this function is called, the BIT runs asynchronously in the background. The status and progress of the BIT can be checked by calling **Gx6256BITGetStatus**.

### Example

The following example runs the built in test and checks the status until the test has completed:

```
BOOL   bPass;
CHAR   szBITResult[1024];
DOUBLE dPercentCompleted;

Gx6256BITRun (nHandle, GX6256_BIT_MODE_ALL, &nStatus);
Gx6256BITGetStatus (nHandle, &bPass, szBITResult, 1024, &dPercentCompleted, &nStatus);
while (dPercentCompleted<100)
{
    printf("BIT Percent Completed = %f %", dPercentCompleted);
    Gx6256BITGetStatus (nHandle, &bPass, szBITResult, 1024, &dPercentCompleted, &nStatus);
}
```

### See Also

**Gx6256BITGetStatus, Gx6256BITRun**, **GxSWGetErrorString**

## Gx6256BleedRelayClose

**Purpose**

Closes the bleed relay for a specified mux group

**Syntax**

**Gx6256BleedRelayClose** (*nHandle, nMuxGroup, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

Each mux group has a bleed relay that can be used to bleed the path when making a new connection.

**Example**

The following example closes the mux group A bleed relay:

```
Gx6256BleedRelayClose (nHandle, GX6256_GROUP_A, &nStatus);
```

**See Also**

**Gx6256BleedRelayGetState, Gx6256BleedRelayOpen, GxSWGetErrorString**

## Gx6256BleedRelayGetState

**Purpose**

Returns a bleed relay's state

**Syntax**

**Gx6256BleedRelayGetState** (*nHandle, nMuxGroup, pnState, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| pnState | SHORT | Returns the state of a bleed relay |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

The Gx6256/Gx6192 switching boards contain a matrix.

The Gx6256 matrix connects the mux groups to the global bus lines.

The Gx6192 matrix connects the mux groups to the analog inputs.

### Example

The following returns the state of Mux Group A bleed relay state:

```
Gx6256BleedRelayGetState(nHandle, GX6256_GROUP_A, &nState, &nStatus);
```

### See Also

**Gx6256MatrixClose, Gx6256MatrixOpen, GxSWGetErrorString**

## Gx6256BleedRelayOpen

**Purpose**

Opens the bleed relay for a specified mux group

**Syntax**

**Gx6256BleedRelayOpen** (*nHandle, nMuxGroup, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

Each mux group has a bleed relay that can be used to bleed the path when making a new connection.

**Example**

The following example opens the mux group A bleed relay:

```
Gx6256BleedRelayOpen (nHandle, GX6256_GROUP_A, &nStatus);
```

**See Also**

**Gx6256BleedRelayGetState, Gx6256BleedRelayClose, GxSWGetErrorString**

## Gx6256GetBoardSummary

### Purpose

Returns the board summary.

### Syntax

**Gx6256GetBoardSummary** (*nHandle, szSummary, nSumMaxLen, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 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 GX6256 summary string provides the following data from in the order shown:

Instrument Name (e.g., GX6256)

FPGA version (e.g. 0xA002)

Serial Number (e.g. 62560210)

### Example

For example, the returned string looks like the following:

```
"GX6256 6U High Density 16 16x2 Mux, FPGA-Version:0xA003, S/N 62561234"
```

### See Also

**GxSWGetDriverSummary, GxSWGetErrorString**

## Gx6256GetBoardType

### Purpose

Closes a Mux path

### Syntax

**Gx6256GetBoardType** (*nHandle, pwBoardType, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| pwBoardType | PWORD | Returns the board's type (0x6256 or 0x6192) |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The GX6256 has two paths, primary and secondary, from mux channels to mux selector.

### Example

The following example closes returns the board type:

```
Gx6256GetBoardType(nHandle, &wBoardType , &nStatus);
```

### See Also

**Gx6256GetBoardSummary, GxSWGetErrorString**

## Gx6256GetRelayCount

### Purpose

Returns the relay count associated with all the relays on the specified device

### Syntax

**Gx6256GetRelayCount** (*nHandle, nBlock, padwRelayCount, pnMaxLen, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nBlock | SHORT | The device to read the relay count from<br>0. GX6256_BLOCK_BOARD: Read relay count for all relays on the switching board<br>1. GX6256_BLOCK_RTM: Read relay count for all relays on the connected Rear Transition Module if it exists<br>2. GX6256_BLOCK_SR: Read relay count for all relays on the connected Switching Router if it exists |
| padwRelayCount | PDWORD | Returns an array containing each relay's cumulative open/close count |
| pnMaxLen | PSHORT | Size of the array passed into the padwRelayCount parameter<br>The function returns the actual number of elements written to the array in this variable as well. |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

Each board, Rear Transition Module, and Switching Router contain many relays. Whenever each relay is opened or closed, the count for the specific relays is incremented and then stored on an onboard EEPROM.

Each element in the relay count array, **padwRelayCount**, represents the total number of open/close events for the lifetime of a relay. The element index corresponds with the relay designation number minus one. For example, element 0 in the array contains the relay count for the relay designated "K1", element 1 in the array contains the relay count for the relay designated "K2", and so on.

### Example

The following example returns the relay count for relay K200 on a rear transition board that is connected to slot 5:

```
SHORT nHandle, nStatus;
DWORD adwRelayCount[256];

Gx6256Initialize(5, &nHandle, &nStatus);
Gx6256GetRelayCount(nHandle, GX6256_BLOCK_RTM, adwRelayCount, 256, &nStatus);
printf("Relay K200 has been opened/closed %d times.", adwRelayCount[199]);
```

### See Also

**Gx6256GetRelayStates, GxSWGetErrorString**

## Gx6256GetRelayStates

### Purpose

Returns the state (open or closed) of each relay on the device specified

### Syntax

**Gx6256GetRelayCount** (*nHandle, nBlock, padwRelayStates, pnMaxLen, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nBlock | SHORT | The device to read the relay count from<br>0. GX6256_BLOCK_BOARD: Read relay count for all relays on the switching board<br>1. GX6256_BLOCK_RTM: Read relay count for all relays on the connected Rear Transition Module if it exists<br>2. GX6256_BLOCK_SR: Read relay count for all relays on the connected Switching Router if it exists |
| padwRelayStates | PDWORD | Returns an array containing each relay's state (open or closed) |
| pnMaxLen | PSHORT | Size of the array passed into the padwRelayStates parameter<br>The function returns the actual number of elements written to the array in this variable as well. |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

Each board, Rear Transition Module, and Switching Router contain many relays..

Each element in the relay count array, **padwRelayCount**, represents the current state a relay. The element index corresponds with the relay designation number minus one. For example, element 0 in the array contains the relay state for the relay designated "K1", element 1 in the array contains the relay state for the relay designated "K2", and so on.

### Example

The following example returns the relay state for relay K200 on a rear transition board that is connected to slot 5:

```
SHORT nHandle, nStatus;
DWORD adwRelayStates[256];

Gx6256Initialize(5, &nHandle, &nStatus);
Gx6256GetRelayStates(nHandle, GX6256_BLOCK_RTM, adwRelayStates, 256, &nStatus);
if (adwRelayStates[199]==0)
    printf("Relay K200 is open");
else
    printf("Relay K200 is closed");
```

### See Also

**Gx6256GetRelayCount, GxSWGetErrorString**

## Gx6256Initialize

### Purpose

Initializes the driver for the board at the specified slot number. The function returns a handle that can be used with other GX6256 functions to program the board.

### Syntax

**Gx6256Initialize** (*nSlot, bSimulation, pnHandle, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nSlot | SHORT | GX6256/GX6192 board slot number on the PXI bus. |
| bSimulation | BOOL | Simulation mode flag. Set to TRUE to enable simulation mode. |
| 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 **Gx6256Initialize** function verifies whether or not the GX6256 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 GX6256 functions.

### Example

The following example initializes two GX6256 boards at slot 10 and 12 with simulation mode disabled.

```
SHORT nHandle1, nHandle2, nStatus;
Gx6256Initilize (10, FALSE, &nHandle1, &nStatus);
Gx6256Initilize (12, FALSE, &nHandle2, &nStatus);
if (nHandle1==0 || nHandle2==0)
    {printf("Unable to Initialize the board")
    return;
}
```

### See Also

**Gx6256Reset**, **GxSWGetErrorString**

## Gx6256InitializeVisa

### Purpose

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

### Syntax

**Gx6256InitializeVisa** (*szVisaResource, bSimulation, pnHandle, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| szVisaResource | LPCTSTR | String identifying the location of the specified board in order to establish a session. |
| bSimulation | BOOL | Simulation mode flag. Set to TRUE to enable simulation mode. |
| 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 **Gx6256InitializeVisa** 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 GX6256 boards at PXI bus 5 and device 11 with simulation disabled.

```
SHORT nHandle, nStatus;
Gx6256InitializeVisa (“PXI5::11::INSTR”, FALSE, &nHandle, &nStatus);
if (nHandle==0)
{
    printf("Unable to Initialize the board")
    return;
}
```

### See Also

**Gx6256Initialize, Gx6256Reset**, **GxSWGetErrorString**

## Gx6256MatrixClose

**Purpose**

Closes a Matrix relay

**Syntax**

**Gx6256MatrixClose** (*nHandle, nMuxGroup, nInput, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| nInput | SHORT | Inputs numbers are 0-15 |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

The Gx6256/Gx6192 switching boards contain a matrix.

The Gx6256 matrix connects the mux groups to the global bus lines.

The Gx6192 matrix connects the mux groups to the analog inputs.

### Example

The following example closes input 5 in Mux Group B:

```
Gx6256MatrixClose(nHandle, GX6256_GROUP_B, 5, &nStatus);
```

### See Also

**Gx6256MatrixGetChannel, Gx6256MatrixOpen, GxSWGetErrorString**

## Gx6256MatrixGetChannel

**Purpose**

Returns a Matrix relay's state

**Syntax**

**Gx6256MatrixGetChannel** (*nHandle, nMuxGroup, nInput, pnState, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| nInput | SHORT | Inputs numbers are 0-15 |
| pnState | SHORT | Returns the state of a matrix relay |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

The Gx6256/6192 switching boards contain a matrix.

The Gx6256 matrix connects the mux groups to the global bus lines.

The Gx6192 matrix connects the mux groups to the analog inputs.

**Example**

The following returns the state of Mux Group A, input 5 connection state:

```
Gx6256MatrixGetChannel(nHandle, GX6256_GROUP_A, 5, &nState , &nStatus);
```

**See Also**

**Gx6256MatrixClose, Gx6256MatrixOpen, GxSWGetErrorString**

## Gx6256MatrixOpen

**Purpose**

Opens a Matrix relay

**Syntax**

**Gx6256MatrixOpen** (*nHandle, nMuxGroup, nInput, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| nInput | SHORT | Inputs numbers are 0-15 |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

The Gx6256/6192 switching boards contain a matrix.

The Gx6256 matrix connects the mux groups to the global bus lines.

The Gx6192 matrix connects the mux groups to the analog inputs.

### Example

The following example opens input 5 in Mux Group B:

```
Gx6256MatrixOpen(nHandle, GX6256_GROUP_B, 5, &nStatus);
```

### See Also

**Gx6256MatrixGetChannel, Gx6256MatrixClose, GxSWGetErrorString**

## Gx6256MatrixInputSelectorClose

### Purpose

Connects inputs to the board matrix

### Syntax

**Gx6256MatrixInputSelectorClose** (*nHandle, nInput, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nInput | SHORT | Inputs numbers are 0-15 |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The Gx6256/Gx6192 switching boards contain a matrix.

The Gx6256 matrix connects the mux groups to the global bus lines.

The Gx6192 matrix connects the mux groups to the analog inputs.

This function makes the connection between the board matrix and the "inputs" which can be either global bus lines or analog inputs depending on the type of switching card as described above.

### Example

The following example connects input 5 to the board matrix:

```
Gx6256MatrixInputSelectorClose(nHandle, 5, &nStatus);
```

### See Also

**Gx6256MatrixInputSelectorGetChannel, Gx6256MatrixInputSelectorOpen, GxSWGetErrorString**

## Gx6256MatrixInputSelectorGetChannel

### Purpose

Returns an matrix input connection state

### Syntax

**Gx6256MatrixInputSelectorGetChannel** (*nHandle, nMuxGroup, pnState, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nInput | SHORT | Inputs numbers are 0-15 |
| pnState | SHORT | Returns the state of a matrix input connection |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The Gx6256/Gx6192 switching boards contain a matrix.

The Gx6256 matrix connects the mux groups to the global bus lines.

The Gx6192 matrix connects the mux groups to the analog inputs.

### Example

The following returns the board's matrix input 5 connection state:

```
Gx6256MatrixGetChannel(nHandle, 5, &nState , &nStatus);
```

### See Also

**Gx6256MatrixInputSelectorClose, Gx6256MatrixInputSelectorOpen, GxSWGetErrorString**

## Gx6256MatrixInputSelectorOpen

### Purpose

Disconnects inputs to the board matrix

### Syntax

**Gx6256MatrixInputSelectorOpen** (*nHandle, nInput, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nInput | SHORT | Inputs numbers are 0-15 |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The Gx6256/Gx6192 switching boards contain a matrix.

The Gx6256 matrix connects the mux groups to the global bus lines.

The Gx6192 matrix connects the mux groups to the analog inputs.

This function breaks the connection between the board matrix and the "inputs" which can be either global bus lines or analog inputs depending on the type of switching card as described above.

### Example

The following example disconnects input 5 to the board matrix:

```
Gx6256MatrixInputSelectorOpen(nHandle, 5, &nStatus);
```

### See Also

**Gx6256MatrixInputSelectorGetChannel, Gx6256MatrixInputSelectorClose, GxSWGetErrorString**

## Gx6256MatrixReset

### Purpose

Opens all the Mux relays that are connected to a specified mux group.

### Syntax

**Gx6256MatrixReset** (*nHandle, nMuxGroup, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

This function opens all Matrix relays within a column that represents a specified mux group.

### Example

The following resets Mux group B column in the matrix.

```
Gx6256MatrixReset (nHandle, GX6256_GROUP_B, &nStatus);
```

### See Also

**Gx6256Initialize, Gx6256MuxReset, GxSWGetErrorString**

## Gx6256MuxClose

**Purpose**

Closes a Mux path

**Syntax**

**Gx6256MuxClose** (*nHandle, nMuxGroup, nChannel, nPath, nMuxMode, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| nChannel | SHORT | Channel numbers are 0-15 for GX6256 and 0-11 for GX6192 |
| nPath | SHORT | Mux Paths are (Only applicable to GX6256):<br>0. GX6256_PATH_PRIMARY: The primary path is used to make the connection<br>1. GX6256_PATH_SECONDARY: The secondary path is used to make the connection |
| nMuxMode | SHORT | Mux connection modes are:<br>0. GX6256_MUX_MODE_SINGLE_CONNECTION: Only one mux channel can be closed at a time<br>1. GX6256_MUX_MODE_MULTI_CONNECTION: Multiple mux channels can be closed at the same time |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The GX6256 has two paths, primary and secondary, from mux channels to mux selector.

### Example

The following example closes the channel 5 in Mux Group B using the primary path and multiple connection mode:

```
Gx6256MuxClose(nHandle, GX6256_GROUP_A, 5, GX6256_PATH_PRIMARY, GX6256_MUX_MODE_MULTI_CONNECTION
, &nStatus);
```

### See Also

**Gx6256MuxGetChannel, Gx6256MuxOpen, GxSWGetErrorString**

## Gx6256MuxGetChannel

**Purpose**

Closes a Mux path

**Syntax**

**Gx6256MuxGetChannel** (*nHandle, nMuxGroup, nChannel, nPath, pnState, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| nChannel | SHORT | Channel numbers are 0-15 for GX6256 and 0-11 for GX6192 |
| nPath | SHORT | Mux Paths are (Only applicable to GX6256):<br>0. GX6256_PATH_PRIMARY: The primary path is used to make the connection<br>1. GX6256_PATH_SECONDARY: The secondary path is used to make the connection |
| pnState | SHORT | Returns the state of a Mux connection |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

The GX6256 has two paths, primary and secondary, from mux channels to mux selector.

### Example

The following returns the state of Mux Group A, channel 5 connection state:

```
Gx6256MuxGetChannel(nHandle, GX6256_GROUP_A, 5, &nState , &nStatus);
```

### See Also

**Gx6256MuxClose, Gx6256MuxOpen, GxSWGetErrorString**

## Gx6256MuxOpen

**Purpose**

Opens a Mux path

**Syntax**

**Gx6256MuxOpen** (*nHandle, nMuxGroup, nChannel, nPath, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| nChannel | SHORT | Channel numbers are 0-15 for GX6256 and 0-11 for GX6192 |
| nPath | SHORT | Mux Paths are (Only applicable to GX6256):<br>0. GX6256_PATH_PRIMARY: The primary path is used to make the connection<br>1. GX6256_PATH_SECONDARY: The secondary path is used to make the connection |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

The GX6256 has two paths, primary and secondary, from mux channels to mux selector.

**Example**

The following example opens the primary path for channel 5 in Mux Group B:

```
Gx6256MuxOpen(nHandle, GX6256_GROUP_A, 5, GX6256_PATH_PRIMARY, &nStatus);
```

**See Also**

**Gx6256MuxGetChannel, Gx6256MuxClose, GxSWGetErrorString**

## Gx6256MuxReset

### Purpose

Opens all the Mux relays.

### Syntax

**Gx6256MuxReset** (*nHandle, nMuxGroup, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

This function opens all Mux and Matrix relays. If an RTM or SR is connected to the board, then the relays on those devices are also opened.

### Example

The following resets Mux group B.

```
Gx6256MuxReset (nHandle, GX6256_GROUP_B, &nStatus);
```

### See Also

**Gx6256Initialize, Gx6256MuxReset, GxSWGetErrorString**

## Gx6256MuxGetSelectorMode

### Purpose

Returns a mux group's selector mode.

### Syntax

**Gx6256MuxGetSelectorMode** (*nHandle, nMuxGroup, pnSelectorMode, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| pnSelectorMode | PSHORT | Mux selector modes are:<br>0. GX6256_SELECTOR_MODE_NONE: The Mux, Matrix, and external lines are not connected to each other.<br>1. GX6256_SELECTOR_MODE_MATRIX_ONLY: The Mux and Matrix are allowed to connect but the external lines are disconnected.<br>2. GX6256_SELECTOR_MODE_EXTERNAL_ONLY: The Mux and the external lines are allowed to connect to each other.<br>3. GX6256_SELECTOR_MODE_MATRIX_AND_EXTERNAL: The Mux is allowed to connect to the Matrix and external lines.<br>4. GX6256_SELECTOR_MODE_MATRIX_ISOLATED: The Mux is not allowed to connect to the selector, effectively making it isolated from the rest of the board. |

| | | |
|---|---|---|
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

**Example**

The following returns Mux group B selector mode:

```
SHORT nSelectorMode;
Gx6256MuxGetSelectorMode(nHandle, GX6256_GROUP_B, &nSelectorMode, &nStatus);
```

**See Also**

**Gx6256MuxSetSelectorMode, GxSWGetErrorString**

## Gx6256MuxSetSelectorMode

### Purpose

Sets a mux group's selector mode.

### Syntax

**Gx6256MuxSetSelectorMode** (*nHandle, nMuxGroup, nSelectorMode, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nMuxGroup | SHORT | Mux Group numbers are:<br>0. GX6256_GROUP_A<br>1. GX6256_GROUP_B<br>2. GX6256_GROUP_C<br>3. GX6256_GROUP_D<br>4. GX6256_GROUP_E<br>5. GX6256_GROUP_F<br>6. GX6256_GROUP_G<br>7. GX6256_GROUP_H<br>8. GX6256_GROUP_I<br>9. GX6256_GROUP_J<br>10. GX6256_GROUP_K<br>11. GX6256_GROUP_L<br>12. GX6256_GROUP_M<br>13. GX6256_GROUP_N<br>14. GX6256_GROUP_O<br>15. GX6256_GROUP_P |
| nSelectorMode | SHORT | Mux selector modes are:<br>0. GX6256_SELECTOR_MODE_NONE: The Mux, Matrix, and external lines are not connected to each other.<br>1. GX6256_SELECTOR_MODE_MATRIX_ONLY: The Mux and Matrix are allowed to connect but the external lines are disconnected.<br>2. GX6256_SELECTOR_MODE_EXTERNAL_ONLY: The Mux and the external lines are allowed to connect to each other.<br>3. GX6256_SELECTOR_MODE_MATRIX_AND_EXTERNAL: The Mux is allowed to connect to the Matrix and external lines.<br>4. GX6256_SELECTOR_MODE_MATRIX_ISOLATED: The Mux is not allowed to connect to the selector, effectively making it isolated from the rest of the board. |

| | | |
|---|---|---|
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

**Example**

The following sets Mux group B selector mode to matrix only:

```
Gx6256MuxSetSelectorMode(nHandle, GX6256_GROUP_B, GX6256_SELECTOR_MODE_MATRIX_ONLY, &nStatus);
```

**See Also**

**Gx6256MuxGetSelectorMode, GxSWGetErrorString**

## Gx6256Panel

### Purpose

Opens a virtual panel used to interactively control the GX6256 board.

### Syntax

**Gx6256Panel** (*pnHandle, hwndParent, nMode, phwndPanel, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| panhandle | PSHORT | Handle to a GX6256/GX6192 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 GX6256 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;
...
Gx6256Panel (&nHandle, 0, 1, &dwPanel, &nStatus);
```

### See Also

**Gx6256Initialize**, **GxSWGetErrorString**

## Gx6256PioGetOutputData

### Purpose

Returns the PIO output value from the output register

### Syntax

**Gx6256PioGetOutputData** (*nHandle*, *pucOutputData*, *pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| pucOutputData | PBYTE | Bit field that represents the 8 output lines of the PIO |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

This function controls the output (driving) data of the 8 line PIO device.

### Example

The following example returns the output data register value:

```
BYTE ucOuputData;
Gx6256PioGetOutputData (nHandle, &ucOutputData, &nStatus);
```

### See Also

**Gx6256PioGetOutputData, Gx6256PioSetOutputEnable, Gx6256PioGetOutputEnable, GxSWGetErrorString**

## Gx6256PioGetOutputEnable

### Purpose

Returns the PIO output enable (tri-state) value of the output register

### Syntax

**Gx6256PioGetOutputEnable** (*nHandle, pucOutputEnable, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| pucOutputEnable | PBYTE | Bit field that represents the 8 enable (tri-state) control lines of the PIO |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

This function controls the tri-state data of the 8 line PIO device.

### Example

The following example returns the output enable (tri-state) control data:

```
BYTE ucOutputEnable
Gx6256PioGetOutputEnable (nHandle, &ucOutputEnable, &nStatus);
```

### See Also

**Gx6256PioSetOutputEnable, Gx6256PioSetOutputData, Gx6256PioGetOutputData, GxSWGetErrorString**

## Gx6256PioReadInput

### Purpose

Reads PIO sampled value from input register

### Syntax

**Gx6256PioReadInput**(*nHandle, pucInputData, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| pucInputData | PBYTE | Bit field that represents the 8 input lines of the PIO |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

This function reads the last sampled value from the input register.

### Example

The following example reads all 8 input lines from the PIO:

```
BYTE ucInputData;
Gx6256PioReadINput (nHandle, &ucInputData &nStatus);
```

### See Also

**Gx6256PioSetOutputData , Gx6256PioGetOutputData, Gx6256PioSetOutputEnable, Gx6256PioGetOutputEnable, GxSWGetErrorString**

## Gx6256PioSetOutputData

### Purpose

Writes PIO output value to output register

### Syntax

**Gx6256PioGetOutputData** (*nHandle*, *ucOutputData*, *pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| ucOutputData | BYTE | Bit field that represents the 8 output lines of the PIO |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

This function controls the output (driving) data of the 8 line PIO device.

### Example

The following example sets all 8 IO lines to generate a logic HI:

```
Gx6256PioSetOutputEnable (nHandle, 0xFF &nStatus);
Gx6256PioSetOutputData (nHandle, 0xFF &nStatus);
```

### See Also

**Gx6256PioGetOutputData, Gx6256PioSetOutputEnable, Gx6256PioGetOutputEnable, GxSWGetErrorString**

## Gx6256PioSetOutputEnable

### Purpose

Writes PIO output enable (tri-state) value to the output register

### Syntax

**Gx6256PioSetOutputEnable** (*nHandle, ucOutputEnable, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| ucOutputEnable | BYTE | Bit field that represents the 8 enable (tri-state) control lines of the PIO |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

This function controls the tri-state data of the 8 line PIO device.

### Example

The following example sets all 8 lines to output enabled:

```
Gx6256PioSetOutputEnable (nHandle, 0xFF, &nStatus);
```

### See Also

**Gx6256PioGetOutputEnable, Gx6256PioSetOutputData, Gx6256PioGetOutputData, GxSWGetErrorString**

## Gx6256Reset

### Purpose

Opens all the board relays.

### Syntax

**Gx6256Reset** (*nHandle*, *pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

This function opens all Mux and Matrix relays. If an RTM or SR is connected to the board, then the relays on those devices are also opened.

### Example

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

```
Gx6256Initilize (10, FALSE, &nHandle, &nStatus);
if (nStatus < 0)
    return nStatus;// return error
Gx6256Reset (nHandle, &nStatus);
```

### See Also

**Gx6256Initialize**, **GxSWGetErrorString**

## Gx6256RtmMatrixClose

**Purpose**

Closes a Matrix relay on the rear transition module

**Syntax**

**Gx6256RtmMatrixClose** (*nHandle, nChannel, nGlobalBusLine, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nChannel | SHORT | Instrument channels (0-31) |
| nGlobalBusLine | SHORT | Global Bus line (0-15) |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

The rear transition module contains a matrix that connects instrument inputs to global bus lines (on the backplane).

**Example**

The following example closes the channel 5 and global bus line 15:

```
Gx6256RtmMatrixClose(nHandle, 5, 15, &nStatus);
```

**See Also**

**Gx6256RtmMatrixGetChannel, Gx6256RtmMatrixOpen, GxSWGetErrorString**

## Gx6256RtmMatrixGetChannel

### Purpose

Returns a rear transition board Matrix relay's state

### Syntax

**Gx6256RtmMatrixGetChannel** (*nHandle, nChannel, nGlobalBusLine, pnState, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nChannel | SHORT | Instrument channels (0-31) |
| nGlobalBusLine | SHORT | Global Bus line (0-15) |
| pnState | SHORT | Returns the state of a rear transition module matrix relay |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The rear transition module contains a matrix that connects instrument inputs to global bus lines (on the backplane).

### Example

The following returns the state of channel 5, global bus line 15 connection state:

```
Gx6256RtmMatrixGetChannel(nHandle, 5, 15, &nState , &nStatus);
```

### See Also

**Gx6256RtmMatrixClose, Gx6256RtmMatrixOpen, GxSWGetErrorString**

## Gx6256RtmMatrixOpen

### Purpose

Opens a Matrix relay on the rear transition module

### Syntax

**Gx6256RtmMatrixOpen** (*nHandle, nChannel, nGlobalBusLine, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nChannel | SHORT | Instrument channels (0-31) |
| nGloblBusLine | SHORT | Global Bus line (0-15) |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The rear transition module contains a matrix that connects instrument inputs to global bus lines (on the backplane).

### Example

The following example opens the channel 5 and global bus line 15:

```
Gx6256RtmMatrixOpen (nHandle, 5, 15, &nStatus);
```

### See Also

**Gx6256RtmMatrixGetChannel, Gx6256RtmMatrixClose, GxSWGetErrorString**

## Gx6256SrBusClose

### Purpose

Makes a switch router connection between switching boards

### Syntax

**Gx6256SrBusClose** (*nHandle, nGlobalBusLine, nSlot, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nGlobalBusLine | SHORT | Global Bus line (0-15) |
| nSlot | SHORT | Slot number (1-9) |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The Switching router routes the 16 lines of the global bus between the slots.

### Example

The following example connects the switch boards from slot 2 and slot 3 via global bus line 5:

```
Gx6256Initialize(2, FALSE, &nHandle1, &nStatus);
Gx6256Initialize(3, FALSE, &nHandle2, &nStatus);
Gx6256SrBusClose(nHandle1, 5, 1, &nStatus);
Gx6256SrBusClose(nHandle2, 5, 2, &nStatus);
```

### See Also

**Gx6256SrBusGetChannel, Gx6256SrBusOpen, GxSWGetErrorString**

## Gx6256SrBusGetChannel

**Purpose**

Returns a switch router connection state between switching boards

**Syntax**

**Gx6256SrBusGetChannel** (*nHandle, nGlobalBusLine, nSlot, pnState, pnStatus*)

**Parameters**

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nGlobalBusLine | SHORT | Global Bus line (0-15) |
| nSlot | SHORT | Slot number (1-9) |
| pnState | PSHORT | Returns the state of a rear transition module matrix relay |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

**Comments**

The Switching router routes the 16 lines of the global bus between the slots.

**Example**

The following returns the state of channel 5, global bus line 15 connection state:

```
Gx6256RtmMatrixGetChannel(nHandle, 5, 15, &nState , &nStatus);
```

**See Also**

**Gx6256SrBusClose, Gx6256SrBusOpen, GxSWGetErrorString**

## Gx6256SrBusOpen

### Purpose

Opens a Matrix relay on the rear transition module

### Syntax

**Gx6256SrBusOpen** (*nHandle, nGlobalBusLine, nSlot, pnStatus*)

### Parameters

| Name | Type | Comments |
|---|---|---|
| nHandle | SHORT | Handle to a GX6256/GX6192 board. |
| nGloblBusLine | SHORT | Global Bus line (0-15) |
| nSlot | SHORT | Slot number (1-9) |
| pnStatus | PSHORT | Returned status: 0 on success, negative number on failure. |

### Comments

The rear transition module contains a matrix that connects instrument inputs to global bus lines (on the backplane).

### Example

The following example disconnects the switch boards from slot 2 and slot 3 via global bus line 5:

```
Gx6256Initialize(2, FALSE, &nHandle1, &nStatus);
Gx6256Initialize(3, FALSE, &nHandle2, &nStatus);
Gx6256SrBusOpen(nHandle1, 5, 1, &nStatus);
Gx6256SrBusOpen(nHandle2, 5, 2, &nStatus);
```

### See Also

**Gx6256SrBusGetChannel, Gx6256SrBusClose, 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 | Cant open hw |
| -2 | Board not exist |
| -3 | Wrong board |
| -4 | Slot not configuration |
| -5 | Unable register device |
| -6 | Unable allocate device resource |
| -7 | Unable allocate memory |
| -8 | Unable create panel |
| -9 | Unable to get timer |
| -10 | Board invalid Eeprom |
| -11 | Not in calibration mode |
| -12 | Not calibrated |
| -13 | Err function not supported |
| -14 | Not MTS board |
| -15 | Unable to open file |
| -16 | Error file not exist |

-17 Error mode not supported by slot

-18 Not PXI board

-19 Not PCI board

**Parameter Errors**

-20 Invalid error

-21 Invalid parameter

-22 Invalid slot

-23 Invalid handle

-24 Invalid string length

-25 Invalid mode

-26 Parameter out of range

-27 File extension not supported

-28 Invalid chassis number

-29 Invalid calibration time stamp

**VISA error**

-30 Load DLL error

-31 Open default resource manager error

-32 Open error

-33 Get attribute error

-34 VI-In error

-35 Memory map error

**Misc error**

-37 Sync create

-38 Sync timeout

-39 Lvrt unsupported

-40 License error

-41 Event enable failed

-42 Event disable failed

-43 Event wait timeout

-44 Event wait error

-45 DMA mem alloc failed

-46 Error DMA mem un map

-47 Error DMA mem free

-48 Error checksum

**Bult In Test error**

-50 BIT adaptor not connected

-51 BIT comparator

-52 BIT close open

-53 BIT close column

-54 BIT switch other

-55 BIT partial

**Board Errors/Warnings**

-60 Error closed loop

-61 Invalid relay number

-62 Invalid number of relays

-63 Invalid relays cycles limit

-64 Invalid relay cycles array size

-65 Error over limit relay replacement

-66 Eeprom busy timeout

-67 Error closed relays over limit

-68 Error set relays timeout

**Board specific parameter error**

-80 Invalid configuration

-81 Invalid channel

-82 Invalid bus

-83 Invalid group

-84 Invalid row

-85 Invalid column

-86 Invalid relay

-87 Invalid daisy chain mode

-88 Invalid relay type

-89 Invalid group mode

-90 Invalid section

-91 Invalid path

-92 Invalid signal type

-93 Invalid input

-94 Simulation INI file read error

-95 Invalid block

-96 Error bit still running

-97 Eeprom checksum invalid

**Example**

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

```
CHARsz[256];
SHORT nStatus, nHandle;

GX6256Initialize(3, FALSE, &Handle, &Status);
if (nStatus<0)
{   GxSWGetErrorString(nStatus, sz, sizeof sz, &nStatus);
    printf(sz);// print the error string return;
}
```

# Index


**A**

About Page ....26
Accessories ....36
Architecture ....1, 10
ATEasy ....29

**B**

BIT Page ....25
Board Description ....1

**C**

Connectors ....33, 34, 36
Connectors and Accessories ....36
Connectors and Jumpers ....34
Copyright ....1
Corrupt files ....28

**D**

Directories ....29
Disclaimer ....1
Driver
  Directory ....29
  Files ....29

**G**

GENASYS ....i, 3, 4
GX6009 ....1, 4, 5, 9, 15, 16, 17, 18, 23, 24, 27, 60
GX6032 ....1, 4, 5, 8, 14, 16, 18, 22, 24, 27, 60
GX6192 1, 4, 5, 7, 8, 9, 10, 12, 13, 16, 18, 20, 24, 25, 27, 29, 35, 36, 42, 43, 44, 45, 46, 59, 60, 62, 63, 64, 65, 66, 68, 69, 70, 71, 72, 73, 76, 78, 80, 82, 83, 84, 85, 86, 88, 90, 92, 93, 95, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109
GX6256 ..1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 16, 18, 19, 20, 21, 24, 25, 27, 29, 31, 34, 36, 37, 38, 39, 40, 41, 59, 60, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109
GX6256 Functions ....60
Gx6256BITAbort ....62
Gx6256BITGetStatus ....63
Gx6256BITRun ....64
Gx6256BleedRelayClose ....65
Gx6256BleedRelayGetState ....66
Gx6256BleedRelayOpen ....68
Gx6256GetBoardSummary ....69
Gx6256GetBoardType ....70
Gx6256GetRelayCount ....71
Gx6256GetRelayStates ....72
Gx6256Initialize ....59, 73
Gx6256InitializeVisa ....30, 75
Gx6256MatrixClose ....76
Gx6256MatrixGetChannel ....78
Gx6256MatrixInputSelectorClose ....82
Gx6256MatrixInputSelectorGetChannel ....83
Gx6256MatrixInputSelectorOpen ....84
Gx6256MatrixOpen ....80
Gx6256MatrixReset ....85
Gx6256MuxClose ....86
Gx6256MuxGetChannel ....88
Gx6256MuxGetSelectorMode ....93
Gx6256MuxOpen ....90
Gx6256MuxReset ....92
Gx6256MuxSetSelectorMode ....95
Gx6256Panel ....97
Gx6256PioGetOutputData ....98
Gx6256PioGetOutputEnable ....99
Gx6256PioReadInput ....100
Gx6256PioSetOutputData ....101
Gx6256PioSetOutputEnable ....102
Gx6256Reset ....103
Gx6256RtmMatrixClose ....104
Gx6256RtmMatrixGetChannel ....105
Gx6256RtmMatrixOpen ....106

Gx6256SrBusClose .....107
Gx6256SrBusGetChannel.....108
Gx6256SrBusOpen.....109
GX7016 .....i, 1, 3, 4, 5, 8, 9, 10, 18, 29, 47, 59
GXSW .....28
GXSW.EXE.....28
GxSWGetErrorString .....59, 110

**H**

Handle.....31, 32
HF Slot Page.....20
HW .....29, 33

**I**

If You Need Help .....1
Installation Directories.....29
Installation: .....31, 33

**J**

J2 .....37, 42
  Connector .....37, 42
J3 .....39, 44
  Connector .....39, 44
J7 .....41
  Connector .....41
Jumpers.....34

**L**

LF Slot Page .....19

**P**

Panel .....18, 19, 20, 22, 23, 26, 28
PCI.....29
Plug & Play.....33
PXI.....30, 31, 32, 33
PXI/PCI Explorer .....30, 73, 75

**R**

README.TXT .....29
Relay Count Page .....24
Reset .....18

**S**

Safety and Handling.....1
Setup .....28, 29
Setup Maintenance.....28
Setup-and-Installation.....27
Side view of the GX6256.....6
Side view of the GX7016.....5
Simulation.....18, 73, 75, 112
Slot.....31, 33
Specifications.....1, 16
SwitchEasy .....1, 47
SwitchEasy Panel.....48
System
  Directory.....29
System-Requirements .....27

**T**

Trademarks .....2
TS-323 .....1, 47

**V**

Virtual Panel .....1, 18, 19, 20, 22, 23, 26, 28
  About Page .....26
  BIT Page.....25
  Relay Count Page .....24
  Setup Page .....19, 22, 23
  Setup Page .....20
VISA.....29, 30, 60, 73, 75

**W**

Warranty .....1