ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Astronics / EADS / Talon Instruments SR125A* at our website: *Click HERE*

SR192A Development System v4.0

User Manual

Document No: 021080010
July 19, 2002

Copyright © 1998-2002
Serendipity Systems, Inc.
All Rights Reserved

Talon Instruments
150 East Arrow Highway
San Dimas, CA 91773

909-599-0690
909-599-6529 Fax

www.taloninst.com
sales@taloninst.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PRODUCT INFORMATION:

SR192a Development System Software (Referred to as “the Software” or “this Software”)
Copyright © 1998-2002, Serendipity Systems, Inc., All rights reserved.
299 Van Deren
PO Box 10477
Sedona, AZ 86339

1. Limited Liability

IF INSTALLED AND OPERATED AS REQUIRED, THE SOFTWARE SHOULD PERFORM AS DESCRIBED IN THE DOCUMENTATION ENCLOSED HEREWITH. ALL OTHER ASPECTS THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY OTHER WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED.

YOU ARE NOT GRANTED ANY IMPLIED WARRANTIES OR MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS WITH YOU EXCEPT AS SET FORTH IN PARAGRAPH 5 BELOW. SHOULD THE SOFTWARE PROVE DEFECTIVE, YOU (AND NOT SSI OR ANY AUTHORIZED SSI DEALER) ASSUME THE ENTIRE RESPONSIBILITY AND COST OF ALL NECESSARY OR INCIDENTAL RESULTS PRODUCED, AS WELL AS ANY DAMAGES OF ANY KIND AND ANY SERVICE, REPAIR OR CORRECTION THAT MAY BE REQUIRED.

SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

Serendipity Systems, Inc. (SSI hereinafter) does not warrant that any of the functions contained in the Software will meet you requirements or that the operation of the Software will be uninterrupted or error free.

SSI warrants that the distribution media on which the Software is furnished to be free from defects in material or workmanship under its intended use, for a period of ninety (90) days from the date of delivery to you as evidenced by a copy of your receipt.

SSI shall not be responsible for any adverse effects caused by Acts of God or any other cause beyond SSI's reasonable control.

2. Limitation of Remedies

SSI's entire liability and your exclusive remedy shall be limited to the replacement within (30) days, for you (the original acquirer), of any distribution media not meeting SSI's Limited Warranty which is returned to SSI or any authorized dealer with a clearly legible copy of your receipt.

IN NO EVENT WILL SSI BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, LOST OPPORTUNITIES, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH SOFTWARE, EVEN IF SSI OR AN AUTHORIZED DEALER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MY NOT APPLY TO YOU.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents

1. INTRODUCTION 1

1.1 System Architecture 3
1.2 System Operation 4
1.3 SR192A Development Environment 5
1.4 SR192A Hardware 6
1.5 Documentation and Printing 6
1.6 Context-Sensitive Help 6

2. PROJECT BROWSER 7

2.1 Project Validation 8
2.2 Toolbar 8
2.3 Menus 9
2.3.1 File Menu 9
2.3.2 Edit Menu 11
2.3.3 Options Menu 12
2.3.4 Help Menu 13
2.4 Dialogs & Windows 14
2.4.1 Debug Log 14

3. SR MODULE CONFIGURATION 15

3.1 Modules 16
3.2 Timing Generators 17
3.3 Cell Test Parameters 18
3.4 Advanced Settings 19
3.5 MFC 20

4. SIGNAL LIST EDITOR 21

4.1 Pattern Trigger 23
4.2 Toolbar 24
4.3 Menus 25
4.3.1 File Menu 25
4.3.2 Edit Menu 26
4.4 Module Group Control 27
4.4.1 Module Group Properties 28
4.4.2 Signal Delays 29

5. TIMING SET EDITOR 31

5.1 Timing Cell Test 33
5.2 Toolbar 34
5.3 Menus 35
5.3.1 File Menu 35
5.3.2 Edit Menu 36
5.3.3 Set Menu 37
5.3.4 Options Menu 38
5.4 Dialogs & Windows 39
5.4.1 Cell Wizard 39
5.4.2 Assert & Return Wizard 40

6. TABLE EDITOR 41

6.1 Jump Enable & Vector Enable 43
6.2 Error Display 44

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.3 TOOLBAR 45
6.4 MENUS....46
6.4.1 File Menu 46
6.4.2 Import & Export Menu....47
6.4.3 Edit Menu....48
6.4.4 Set Menu 50
6.4.5 Options Menu 52
6.4.6 Fill Menu 53
6.4.7 Errors Menu....54
6.5 DIALOGS & WINDOWS 55
6.5.1 Import Value Change Dump 55
6.5.2 Export Value Change Dump....56
6.6 IMPORT & EXPORT FILE FORMATS....57
6.6.1 Value Change Dump File Format 57
6.6.2 Simple Hex File Format 59
6.6.3 FITS Expected State File Format 60

7. SEQUENCE EDITOR 63

7.1 SEQUENCE OPERATION 64
7.2 SEQUENCE LIST 65
7.3 PARAMETER PANE....66
7.4 TOOLBAR 67
7.5 MENUS....68
7.5.1 File Menu 68
7.5.2 Edit Menu....70
7.5.3 Options Menu 72
7.6 DIALOGS & WINDOWS 73
7.6.1 Vector Table 73
7.6.2 Remove Name....74
7.7 SEQUENCE FILE FORMAT....75

8. TEST MANAGER 77

8.1 OPERATION....78
8.2 SEQUENCE EXECUTION....79
8.3 AUTOMATIC SAVE/RELOAD 80
8.4 MENUS....81
8.4.1 File Menu 81
8.4.2 SR192A Menu 82
8.4.3 Timing Menu 83
8.4.4 Options Menu 84
8.5 EXECUTION RESULTS 85
8.6 EXECUTION RESULT FILE FORMAT 86

9. PROGRAMMING INTERFACE 89

9.1 EXECUTION RESULT REPORTING....91
9.2 DRIVER UPDATES....92
9.3 ERROR LOG FILE....93
9.4 FUNCTIONS 94
9.4.1 ssSrExecute 95
9.4.2 ssSrExecuteSequence 96
9.4.3 ssSrGetConfig 97
9.4.4 ssSrGetErrorMessage....98
9.4.5 ssSrGetExecResults....100
9.4.6 ssSrGetList....102
9.4.7 ssSrHalt....103
9.4.8 ssSrLoadProject 104

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.9 ssSrLoadSequence *105*
9.4.10 ssSrReset *106*
9.4.11 ssSrSelectSR192 ..... *107*
9.4.12 ssSrStatus *108*
9.5 APPLICATION DEVELOPMENT ENVIRONMENTS 109
9.5.1 Visual Basic *110*
9.5.2 Visual C++ *111*
9.5.3 HP-VEE *111*
9.5.4 National Instruments LabWindows/CVI *112*
9.5.5 National Instruments LabView *113*
9.6 EXAMPLES 114
9.6.1 Visual Basic Example *114*
9.6.2 Visual C++ Example *116*

10. APPENDIX A - VERSION CHANGES 119

11. SALES & SUPPORT 120

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1. Introduction

The Talon SR192A is a modern VXI-based high-performance digital instrument that is used as a digital word generator, bus emulator or as a key component in a digital test system. The SR192A is an extremely adaptable instrument that gets much of its versatility through software control. The SR192A Development System is a collection of software components that provide a comprehensive approach for developing, debugging and maintaining applications for the Talon SR192A. Tasks from design to deployment can be accomplished without writing a single line of code.

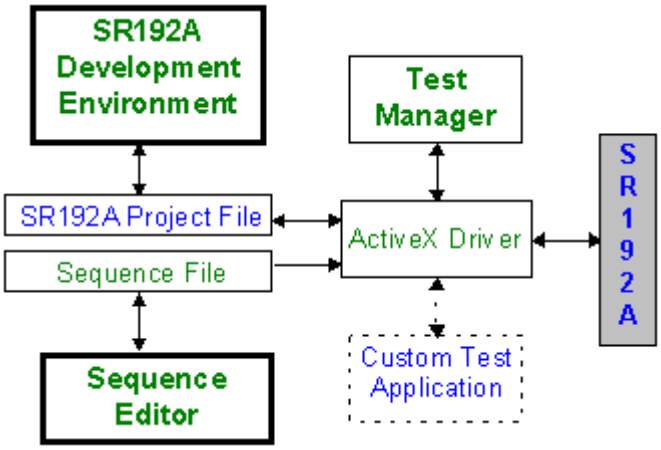
Coupled with the Talon SR192A hardware, the SR192A Development System provides an application developer with a modern, graphical, open-architecture tool that is applicable to a broad range of operating situations and environments.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.1 System Architecture

The SR192A Development System includes the SR192A Development Environment, Sequence Editor, ActiveX Driver DLL, Test Manager and SR192A project file. Digital test data is read and written to a project file by the SR192A Development Environment. From there it is loaded and executed on the SR192A hardware by the Test Manager. Alternately, other applications may use the ActiveX Driver DLL to directly control loading and test execution on the SR192A.



The SR192A Development Environment is used to visually define and view timing and table information for the Talon SR192A. Its advanced user interface design streamlines the process of creating vector patterns and tests. This 32-bit Windows application defines digital test data and timing without cumbersome and arcane programming. In addition, it provides graphical access to the SR192A hardware controls. The SR192A Development Environment writes and reads a project file which stores SR192A application data.

The SR192A Development Environment may be used on a workstation independent of the SR192A hardware. Alternatively, integrated operation with an SR192A is supported when used with the ActiveX Driver DLL and the Test Manager. The SR192A Development Environment is designed to accommodate add-ins for importing test programs from legacy systems, CAD data and simulation data.

The Sequence Editor allows execution sequences to be edited, validated, stored to file and exported as a SCPI file. Sequences are collections of timing set and table pairs (i.e. subsequences) that are executed in a defined order. Their execution order is controlled through conditional jumps and subroutine calls. Each subsequence has a separate count that allows it to be looped up to 32768 times. Building sequence definitions by hand can be tedious and error prone. The Sequence Editor is designed to quickly capture sequence requirements, even while providing a great degree of visibility and flexibility.

The Test Manager is a sample Windows application that uses the ActiveX Driver DLL to handle loading and executing a project file on an SR192A. Test results are displayed and several execution options are supported. The Test Manager is written in Visual Basic and its source code is included with the system. The Test Manager may be used as is, customized or simply serve as a program interface example.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The SR192A ActiveX Driver DLL provides high-level software access to the SR192A hardware. It coordinates the loading and execution of a project file on the SR192A. It can be used to integrate SR192A control with other instruments or applications. Its programming interface supports development in a wide variety of languages and environments. These include HP-VEE, National Instruments LabWindows CVI, National Instruments LabView, C++, Java and Visual Basic.

The SR192A project file is a repository for SR192A software control settings, timing set and table data. It holds all the information required for a specific operation. This greatly simplifies project development, tracking and maintenance. SR192A project files are identifiable by their file extension (*.SRP).

1.2 System Operation

During development, the SR192A Development Environment is used to create a new project, define table data, define timing set data and specify other SR192A properties. Then the signals used in the table definitions are mapped to the physical SR192A channels. A byproduct of this mapping is the realization of a physical interface design. Once the tables, timing sets and control properties have been defined, the information is stored in an SR192A project file.

The debug phase begins when the project file is downloaded to the SR192A and executed with the Test Manager. The Test Manager facilitates initial testing and debug by providing capabilities such as running, looping and reporting test results. The integral connection between the SR192A Development Environment and the Test Manager provides a systematic way to edit, download and run a revised project file.

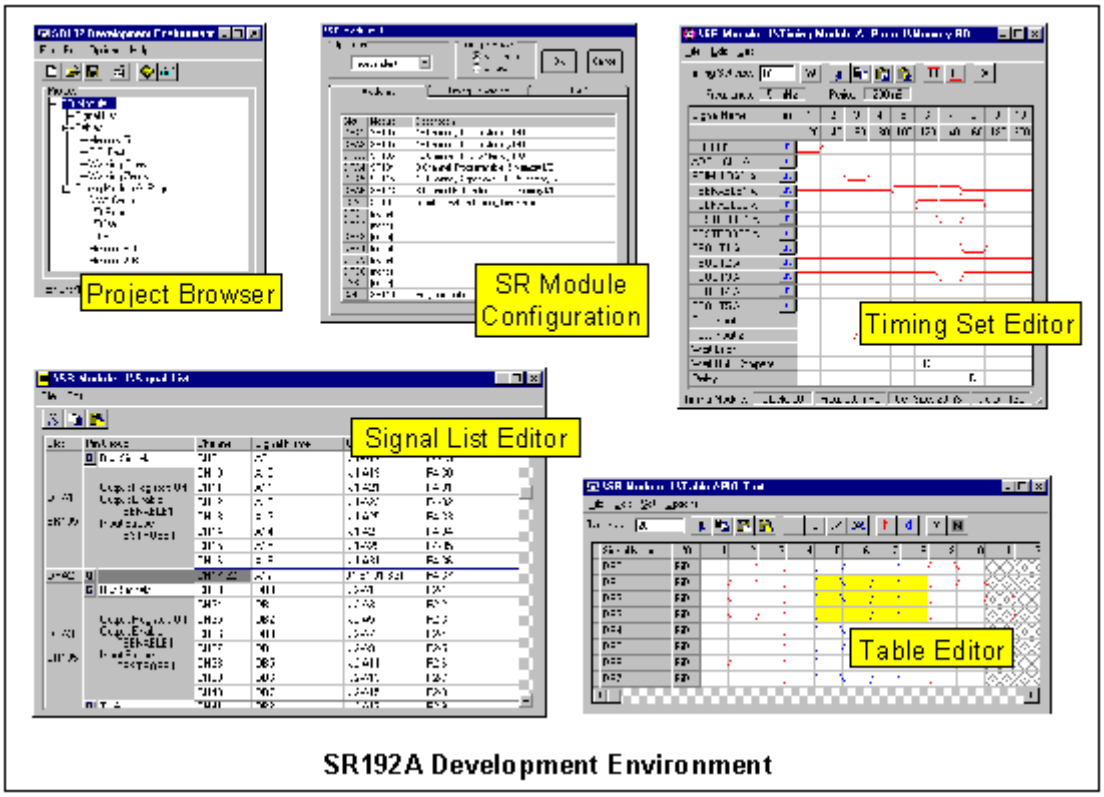
Once debug is complete, the Test Manager can be used by itself for production test. In many circumstances it has sufficient functionality for deployment. If not, the Test Manager source code is provided so that it can be modified to meet specific requirements. Alternatively, the project file can be invoked from a high-level test application, or test executive, through the ActiveX Driver DLL.

Another use for the SR192A Development System is the preservation and maintenance of legacy test data. The SR192A Development System can import several types of digital test data including a value change dump (VCD) file as defined in Section 15 of IEEE Standard 1364-1995. Alternately, LASAR simulation data can be translated to an SR192A project file via an optional utility. This imported project file can then be opened by the SR192A Development Environment for examination, documentation, archiving or modification.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.3 SR192A Development Environment

The SR192A Development Environment contains a set of software components that collectively define the operation of an SR192A instrument. Each aspect of an SR192A's behavior is supported by one of these components.



SR192A Development Environment

The Project Browser defines the structure of an SR192A project. A hierarchical list shows the relationship between various project elements and allows them to be added, deleted, copied and edited. The project list manages timing sets, data tables, signal lists and multiple SR192A modules.

The SR Module Configuration dialog defines the installed SR192A hardware and timing generator settings. This information is typically specified when a project file is first created.

The Signal List Editor matches digital signals to physical channels and defines their operating characteristics. It also handles module group management and cross referencing between SR192A connectors and UUT connector pins. Signals cannot be assigned until pin modules have been specified with the SR Module Configuration dialog.

The Timing Set Editor defines the timing characteristics required for generating digital data. A timing set provides the "pacing" for digital generation. A timing set cycles for each vector of digital data. Within a timing set, stimulus output and response sampling are determined. Additional timing and control signals are also available, as well as a selection of test conditions.

The Table Editor defines the digital data that is output or sampled by an SR192A. A data table is matched with a timing set during execution on an SR192A. Each digital signal has multiple properties including direction, level and monitoring.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.4 SR192A Hardware

The hardware component of the system is a Talon Instruments SR192A Bus Emulator/Word Generator. The SR192A is a VXI-based digital stimulus/response instrument housed in a dual slot VXI card. Each SR192A module supports up to 192 digital I/O channels. Each I/O channel is RAM-backed to a depth of 256K and operates at up to 100 MHz. The SR192A has a unique timing set architecture that is conducive for bus emulation.

This hardware versatility, combined with the SR192A Development System tools, provides several operational possibilities. The system can be used as a classic programmable digital word generator, configurable bus emulator or as a high-speed digital test component in a VXI-based test system.

1.5 Documentation and Printing

Sometimes it is necessary to have digital test information included with paper or on-screen documentation. This is easily achieved by using Windows to capture images from the SR192A Development Environment components. This is particularly useful for signal lists, hardware configurations and timing sets. The following describes how to capture images in Windows.

To copy the window or screen contents

> To copy an image of the window that is currently active, press ALT+PRINT SCREEN.
>
> To copy an image of the entire screen, press PRINT SCREEN.

Tip

> To paste the image into a document, click the Edit menu in the document window, and then click Paste.

To directly print a window or dialog contents

Many of the SR192A Development Environment windows and dialogs support printing from a File Menu and/or via shortcut key (**Ctrl-P**). This print option reduces large window images to fit on a single page.

1.6 Context-Sensitive Help

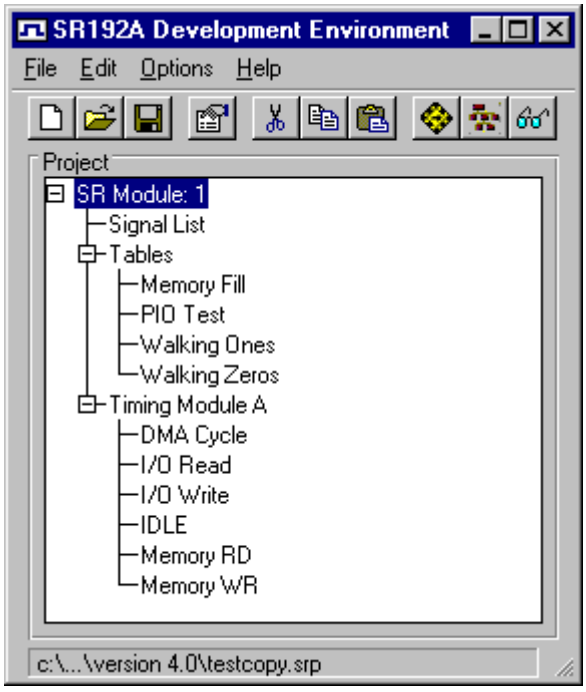
SR192A Development System applications have context-sensitive help for most of their windows and dialog boxes. To access the appropriate help topic, simply press function key **F1**.

> **Note: There are extensive indexing and search capabilities included with the help file (SR192ADEV.HLP). These greatly aid locating necessary information.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. Project Browser

The Project Browser is the primary interface for the SR192A Development Environment. It is used to load or create a project file, design a project list and select individual items for editing or viewing. The Project Browser is also used to dispatch companion tools and coordinate import and conversion tasks.



The project list is an outline or tree view of the elements contained in the project. A single left mouse click on a branch expands [+] or collapses it [-]. Scroll bars are automatically provided if the list exceeds the size of the pane. Individual elements in the list are selected for editing, copying, deleting or renaming. Elements are added with the Edit Menu or a similar pop-up menu (right mouse click on project list). Placement of new elements is relative to the currently selected item. Note that elements with user defined names are listed in alphabetical order.

Tables, timing sets and complete SR Modules can be copied, cut and pasted via the Edit Menu or toolbar. Elements can be copied and pasted within a single project file or between two project files. Copying to a different project file is most easily accomplished by having two SR192A Development Environments active at the same time.

A project contains SR192A modules, signal lists, data tables and timing sets. Each SR192A module has one signal list and one or more tables and timing sets. Timing sets are further subdivided by timing module (A or B). A selected element is viewed or edited by double clicking the left mouse button, pressing the appropriate toolbar button, or through the Edit Menu.

A status bar at the bottom of the Project Browser displays the name of the current project file. Minimizing the Project Browser causes all associated windows to be hidden.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.1 Project Validation

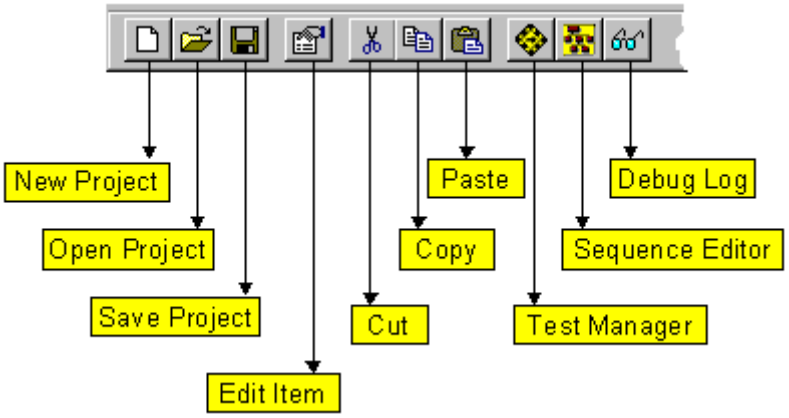
Whenever an SR192A project file is saved, an automatic validation is performed on it. The purpose of the validation is to identify any missing or mismatched elements in the project. This includes missing pin modules, signal names or signal mismatches. Early detection of mismatched signal names can save many hours of hardware debugging.

To avoid warnings about unused signals in a table, preface those signal entries with two forward slashes (e.g. "// Reference Signal" or just "//"). It is often helpful to create unused signal rows for visual spacing or reference.

Validation warnings are displayed on the Debug Log window. Double-clicking a signal warning highlights that signal (if the table is currently displayed).

2.2 Toolbar

The toolbar on the Project Browser provides quick access to commonly used commands. All of the buttons on the toolbar have corresponding menu entries. Many of them also have shortcut keys that perform the same command. The shortcut keys are shown to the right of their menu entries. Tooltips are displayed when the mouse cursor is held over a toolbar button for approximately two seconds. A tooltip box appears temporarily to identify the button's function.

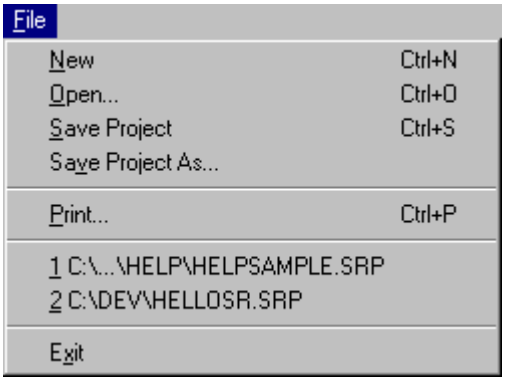


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.3 Menus

2.3.1 File Menu

The Project Browser File Menu is used to manage the loading and saving of SR192A Project files (*.SRP). With this menu, SR192A project files are created, loaded, saved and renamed. A file history list permits quick reloading of recently accessed project files. The SR192A Development Environment can also be closed from this menu.



Menu Option	Description
New	Opens a file browser for naming and placing a new project file. A default configuration is displayed in the project list.
Open…	Opens a file browser for choosing a project file. The chosen file is loaded and displayed in the project list.
Save Project	Updates the project file with the latest editing changes. Project validation is automatically performed when a project is saved.
Save Project As…	Creates a new project file with the latest editing changes. It then becomes the current project file.
Print…	Send an image of the current Project Browser to the printer. Also accomplished by pressing **Ctrl-P**.
File History	This provides a quick selection from the last four project files loaded. The list is updated each time a new file is read. The file names are stored in a state file (SR192ADEV.INI).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Exit	Close the SR192A Development Environment. If the current project has not been saved, the user is prompted to do so before closing.

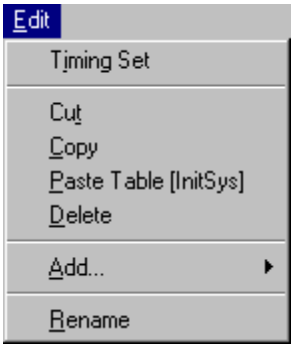
When a project file is loaded, or created, a working copy (*.tmp) is made to support interactive editing. When a project is saved, the original file is replaced with the working copy.

> **Note: If work is lost due to a system crash or power failure, try to recover the working copy of your project file. It is very important to locate and rename it before restarting the SR192A Development Environment. Otherwise, it might be automatically deleted or overwritten.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.3.2 Edit Menu

The Project Browser Edit Menu is used to manage project contents. It controls editing, adding, deleting and renaming of elements in the project list. It can also be accessed as a pop-up menu (right mouse click on project list). Some menu options are dependent upon which project element is selected. Timing sets, tables and complete SR Modules can be copied and pasted within a single project or between two projects.



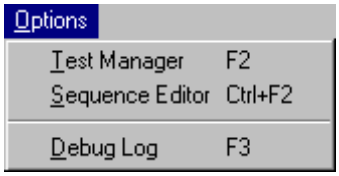
Menu Option	**Description**
<*Selected Item***>**	Activates appropriate editing window for the currently selected item. Also accomplished by double clicking the left mouse button on the item to edit.
Cut	Copy and then delete the selected timing set, table or SR Module.
Copy	Copy the selected timing set, table or SR Module to the paste buffer file (SR192aCopy.buf).
Paste <*Item*>	Paste the named item relative to the current selection.
Delete	Removes the selected item from the project list. Some elements (e.g. signal lists) cannot be removed.
Add…	Presents a list of elements that can be added to the project list. Placement of new elements is relative to the currently selected item. Tables and timing sets are listed in alphabetical order.
Rename	Allows the currently selected item to be renamed. Some elements (e.g. SR Modules) cannot be renamed. Also accomplished by pressing **Ctrl+R**.

Note: Table and timing set names are truncated to ten characters when they are downloaded to the SR192A hardware. This may affect the behavior of sequences or a programmatic interface.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.3.3 Options Menu

The Project Browser Options Menu is used to access external utilities and logging information. The Test Manager is an external utility for interactive control of an SR192A. The Sequence Editor is an external utility for editing execution sequences. They both can be initiated from the Options Menu or executed directly.

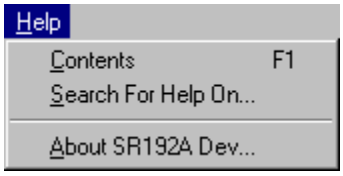


Menu Option	Description
Test Manager	Starts the Test Manager utility for interactive control of an SR192A. The Test Manager automatically reads the current project file and writes it to the SR192A. Function key **F2** also performs this task.
Sequence Editor	Starts the Sequence Editor utility for creating, viewing and modifying execution sequences. Function key **Ctrl+F2** also performs this task.
Debug Log	Allows the user to toggle between the Debug Log window and the Project Browser. Function key **F3** also performs this task.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.3.4 Help Menu

The Project Browser Help Menu provides access to the SR192A Development Environment help file and version information. The help file is most easily accessed by pressing the **F1** function key.



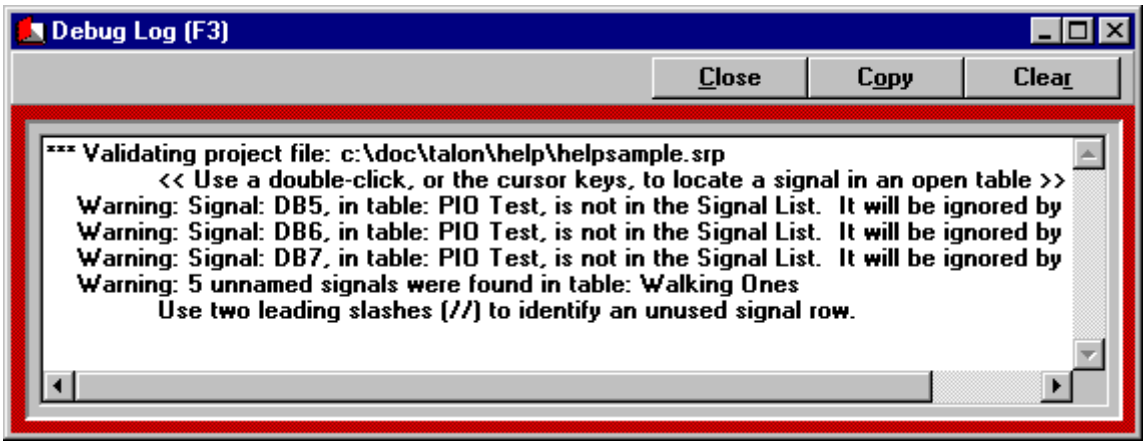
Menu Option	Description
Contents	Displays the Contents page of the help file. Function key **F1** also performs this task.
Search for Help On	Displays the Search index of the help file. Use this to locate specific information in the help file.
About SR192A Dev…	Displays a dialog containing version information, a serial number and copyright notice.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.4 Dialogs & Windows

2.4.1 Debug Log

The Debug Log window displays warnings and errors that occur during SR192A Development Environment operations. This includes problems encountered when loading or saving SR192A Project files. The Debug Log window is activated via the Options Menu. Function key **F3** toggles between the Debug Log and the Project Browser window. Scroll bars and key commands (PgUp, PgDn, Home, etc.) are used to move around the window. Push buttons are provided to copy the selected contents to the Windows clipboard or to clear the display.



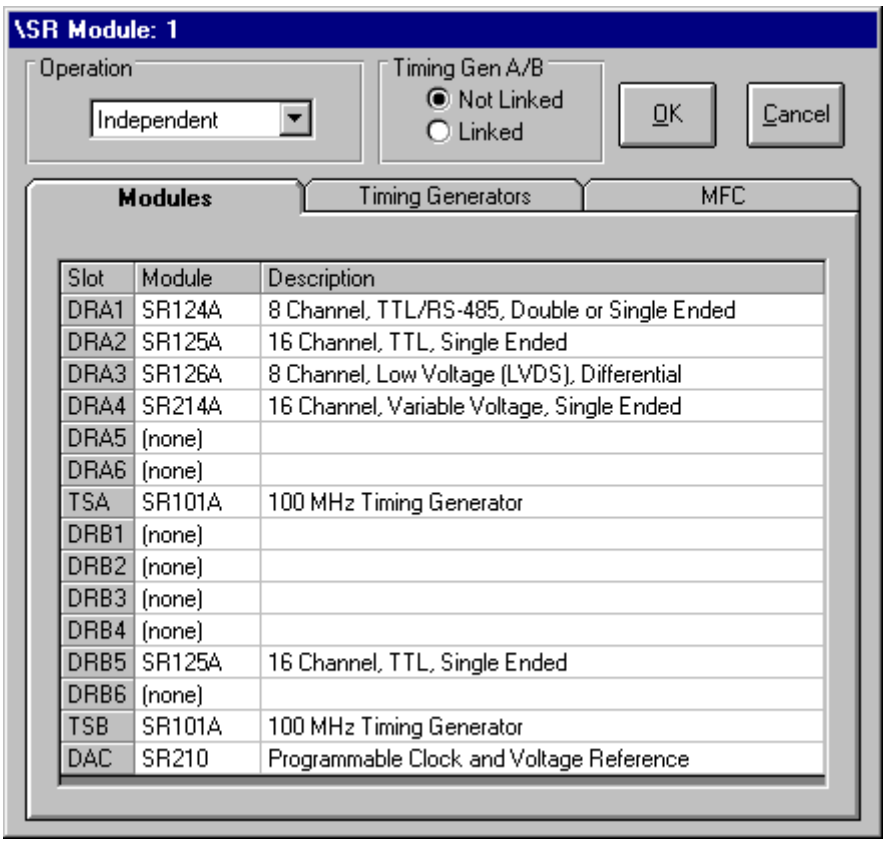
All of the information displayed in this window is also written to a log file (SR192ADev.log). Once this file reaches approximately 200K, it is copied to another file (SR192ADev.bak) as a buffer. The buffer file is deleted when the next one is copied over it.

> **Note: Some error and warning messages respond to a double-click of the left mouse button, or cursor key scrolling. This causes the source of the error, or warning, to be displayed. This is mostly used to highlight signal problems on tables displayed with the Table Editor. These messages are often generated by project validation and VCD importing.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. SR Module Configuration

The SR Module Configuration dialog is used to define SR192A hardware organization, timing generator settings and voltage levels. General operating parameters are also set with this window. Pressing the OK button saves changes and causes them to be distributed to the other elements of the SR192A Development Environment. This window has to be closed before accessing other parts of the system. Use a shortcut key (**Ctrl-P**) to print this dialog.



Slot	Module	Description
DRA1	SR124A	8 Channel, TTL/RS-485, Double or Single Ended
DRA2	SR125A	16 Channel, TTL, Single Ended
DRA3	SR126A	8 Channel, Low Voltage (LVDS), Differential
DRA4	SR214A	16 Channel, Variable Voltage, Single Ended
DRA5	(none)	
DRA6	(none)	
TSA	SR101A	100 MHz Timing Generator
DRB1	(none)	
DRB2	(none)	
DRB3	(none)	
DRB4	(none)	
DRB5	SR125A	16 Channel, TTL, Single Ended
DRB6	(none)	
TSB	SR101A	100 MHz Timing Generator
DAC	SR210	Programmable Clock and Voltage Reference

The top of the dialog has a drop-down list for selecting the operating mode of the SR192A. This choice configures the SR192A to operate independently of other SR192s, as the master to one or more SR192As, or as a slave to another SR192A.

Two radio buttons define whether timing generators A and B will operate linked or unlinked. When unlinked, the two timing generators operate independently of each other. When linked, their operation is synchronized and they execute a shared timing set and data table. The Linked radio button is disabled if only one timing generator is present.

> **Note: It is best to make the linked/unlinked choice early in development because it affects how tables and timing sets are defined.**

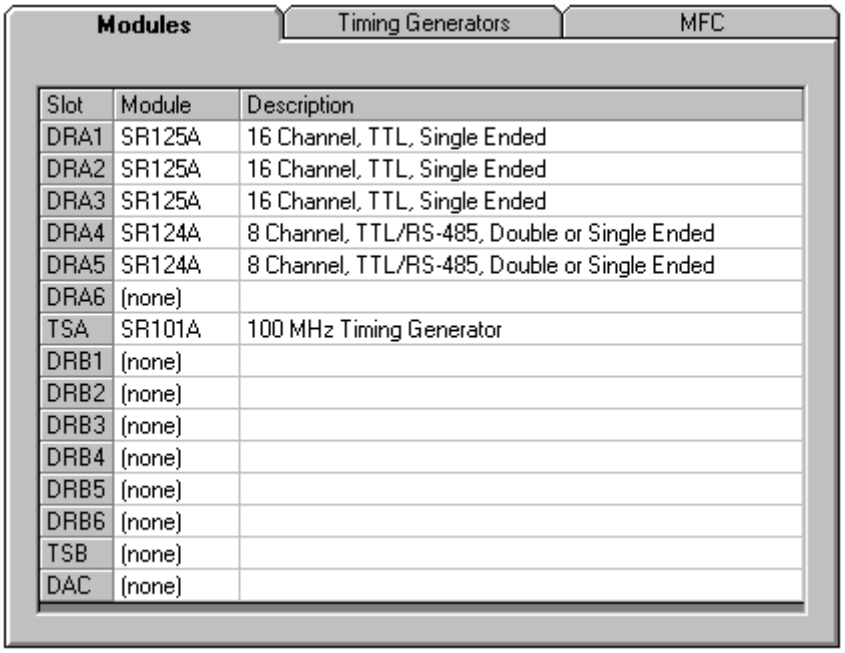
Three tabs allow further selection between Modules, Timing Generators and MFC settings. These define installed hardware modules, set general timing behavior and control voltage settings. Installed pin modules need to be defined before signal mapping and control can be specified with the Signal List Editor. A timing generator and clock source needs to be selected in order to enable certain timing attributes of the Timing Set Editor.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Modules

The Modules tab, of the SR Module Configuration window, is used to define the hardware installed in an SR192A. The specified hardware configuration controls several aspects of the SR192A Development Environment. These include signal lists, timing and voltage levels.

An SR192A is populated with up to 12 pin modules (DRA1-6, DRB1-6), two timing generators (TSA, TSB) and an MFC module. Each of these modules contribute to the characteristics of a particular SR192A. The wide assortment of pin modules includes 8 and 16 channels; TTL, differential and variable voltage levels.



Slot	Module	Description
DRA1	SR125A	16 Channel, TTL, Single Ended
DRA2	SR125A	16 Channel, TTL, Single Ended
DRA3	SR125A	16 Channel, TTL, Single Ended
DRA4	SR124A	8 Channel, TTL/RS-485, Double or Single Ended
DRA5	SR124A	8 Channel, TTL/RS-485, Double or Single Ended
DRA6	(none)	
TSA	SR101A	100 MHz Timing Generator
DRB1	(none)	
DRB2	(none)	
DRB3	(none)	
DRB4	(none)	
DRB5	(none)	
DRB6	(none)	
TSB	(none)	
DAC	(none)	

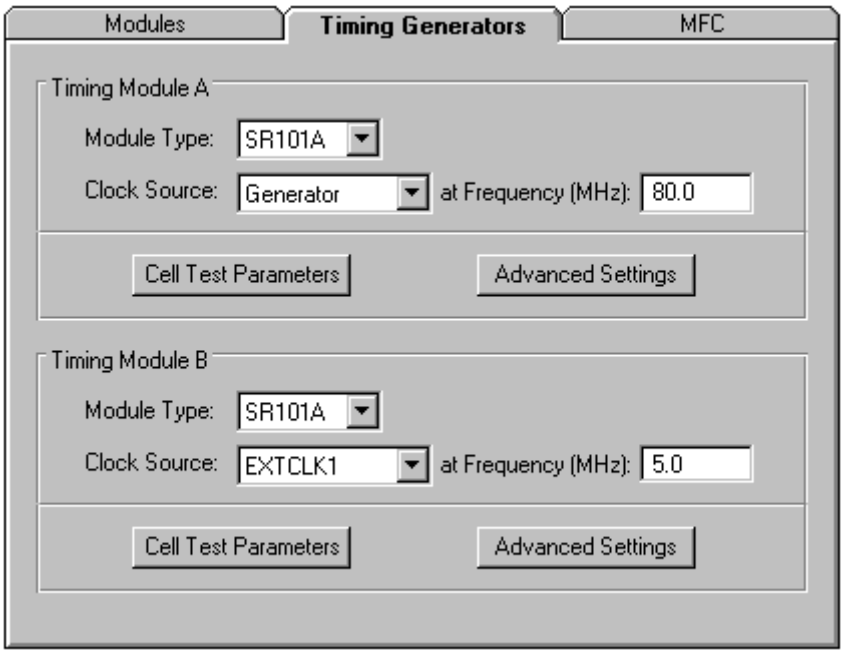
A single left mouse click, in the Module column, reveals a drop-down list of the modules appropriate for the particular slot. Once selected, the Description column is updated with information about the module. Timing Generators and MFC modules can also be selected on their corresponding tabs. Use a shortcut key (**Ctrl-P**) to print an image of this tab.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.2 Timing Generators

The Timing Generators tab, of the SR Module Configuration window, is used to define the timing generator hardware and settings for an SR192A. Timing Module A, in slot TSA, controls timing for channels 1-96. Timing Module B, in slot TSB, controls timing for channels 97-192. If the two generators are linked, a single collection of settings are used for all channels in the SR192A.

Once the module type is selected from the drop-down list, other parameters can then be set. These settings have a direct effect on the behavior of timing sets associated with the timing module. The Cell Test Parameters button activates a dialog box for defining timing set delays and test signal sources. The Advanced Settings button activates a dialog box with controls for external clock outputs, error memory, jump strobe and vector channels. Use a shortcut key (**Ctrl-P**) to print an image of this tab.



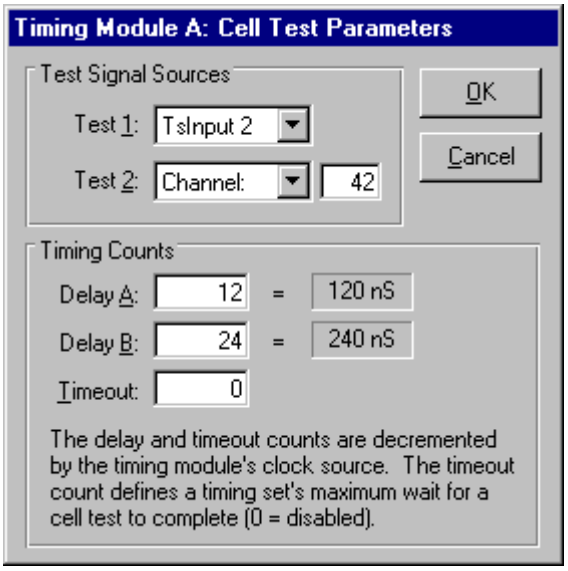
Note: When using an external clock source, enter its frequency (MHz) in the Clock Frequency field. This allows associated timing sets to be displayed and defined with timing values.

When an SR210 module is present, the Clock Source drop-down list includes a programmable clock option (PGMCLK). The specific programmable clock, and its frequency, is set on the MFC tab.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 Cell Test Parameters

The Cell Test Parameters dialog, available from the Timing Generators tab, defines test signal sources and timing set delays. The Test1 and Test2 signals are used by timing sets to synchronize execution with external signal sources (TsInput1, TsInput2, TsInputM), channel behavior or a user-defined trigger condition. The timing counts are also part of a timing set's cell test behavior. Delay A and B provide user-defined delays for timing set cells. The Timeout value controls how long a timing set will wait for a cell test to complete. If a clock frequency is defined on the Timing Generator tab, nonzero count values are automatically displayed in units of time. Use a shortcut key (**Ctrl-P**) to print an image of this dialog.

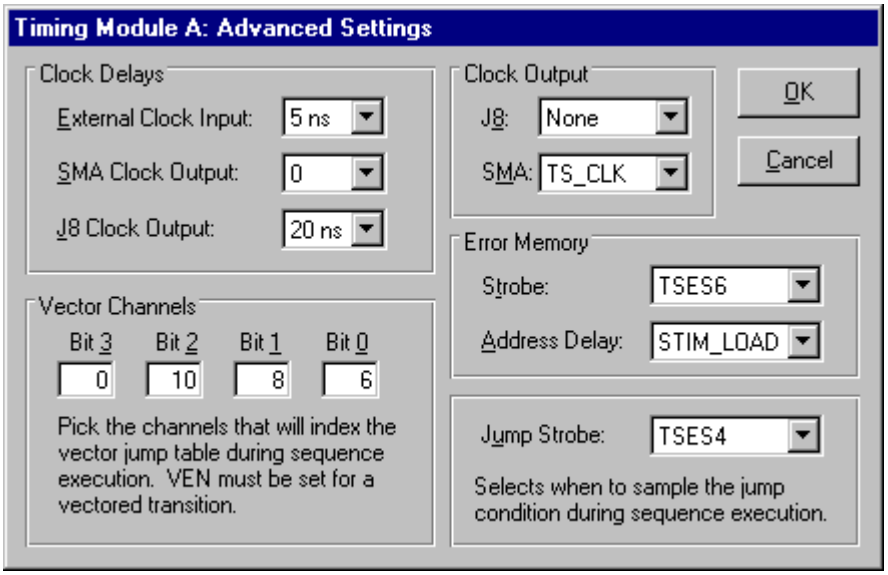


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4 Advanced Settings

The Advanced Settings dialog, available from the Timing Generators tab, defines external clock settings, error memory options, jump strobe conditions and vector channel selections. Delays from 1 to 20 nanoseconds are available for the external clock input and the two clock outputs (SMA and J8). The clock outputs can be disabled, set to TS_CLK or set to the clock generator. Use a shortcut key (**Ctrl-P**) to print an image of this dialog.

Control of the error memory is afforded by the selection of a strobe signal and optional delay signal. The strobe signal can be one of the enable-strobe signals defined in a timing set (TSES1-6), FCNTL1 or FCNTL2. Besides OFF, the error memory address delay can be set to STIM_LOAD, FMA or Both. The Jump Strobe signal controls when the jump condition is sampled during sequence execution. This signal can be one of the enable-strobe signals (TSES1-6), FCNTL1 or FCNTL2.

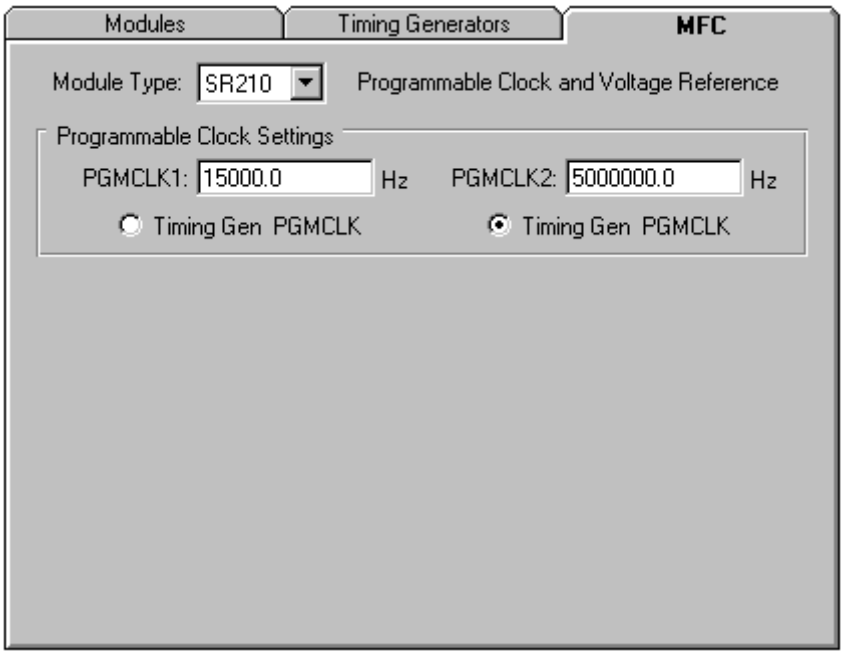


During sequence execution, dynamic transitions can be made via a vector table defined in the Sequence Editor. Up to four signal channels can be selected on this dialog to provide the index to the vector table. A value of zero disables the vector bit. Note that the vector enable (VEN) signal must be set with the Table Editor in order for a vectored transition to take place.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 MFC

The MFC tab, of the SR Module Configuration window, is used to define programmable clock references for an SR192A. A programmable clock can be selected as a clock source by certain types of timing modules. Use a shortcut key (**Ctrl-P**) to print an image of this tab.

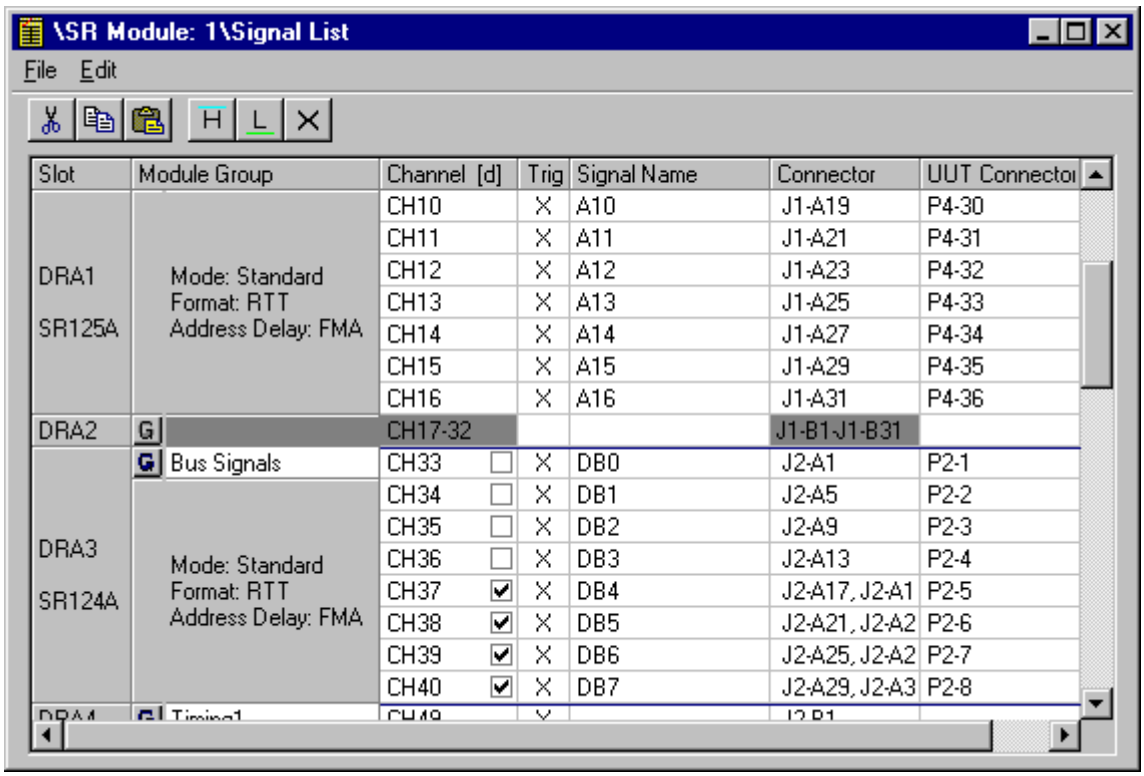


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. Signal List Editor

The Signal List Editor is used to assign signals to physical channels. It also handles the definition of module groups, trigger settings and the optional entry of UUT connector pins. Signals cannot be assigned until the pin modules have been selected with the SR Module Configuration window.

The behavior and names of digital signals are specified with the Table Editor. It is best to use logical or schematic names for signals rather than physical channel numbers. This isolates the signal data from potential changes in fixtures or instrumentation. The Signal List Editor is then used to match signal names to physical channels. If a data table contains signals that are not present in this list, they are automatically identified during project validation.



The leftmost column identifies the slot position (e.g. DRA3) and assigned pin module type (e.g. SR125A) for a group of channels. Pin modules typically have 8 or 16 channels. If no pin module is assigned to a slot, it is displayed as a single row with a dark gray background.

The next column specifies the module group associated with a set of 8 channels. Module groups are a collection of channels (8 to 192) that share behavioral properties (e.g. timing). The specific properties are dependent on the pin module type, but they can include output enables, input strobes and delays. Using named module groups allows many channels to easily share a common set of operational characteristics.

The top row of the Module Group column has a drop-down list of existing module groups that are also compatible with the slot's pin module. Alternately, a new module group name is entered in this location. The remaining rows in the column contain the settings for the specified module group. Once a module group is identified, its settings are modified by clicking on the '**G**' button to

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

the left of its name. This activates the Module Group Control dialog to edit the properties of the module group.

The Channel column identifies the SR192A channel represented by the row, and the Connector column is its corresponding physical location. For an SR124A pin module, the Channel column also has a checkbox. The checkbox sets the electrical characteristics of each individual channel to TTL (unchecked) or differential (checked). The checkboxes can be drag selected and set via the toolbar or Edit Menu (Trigger High = checked, Trigger Low = unchecked).

The pattern trigger column sets optional trigger conditions for timing set synchronization. The remaining two columns are for user-entered signal names and UUT connectors. The signal names provide the necessary mapping between digital behavior defined in a table and the physical channels of an SR192A. The UUT Connector column is provided for additional documentation that could be used when building interface adapters or cable harnesses.

Signal names and UUT connector entries can be cut, copied and pasted from other locations in the SR192A Development Environment, or from many other text sources. A whole column is selected by a left mouse click on its title. Multiple items are selected by holding the left mouse button down and dragging.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.1 Pattern Trigger

The Signal List Editor window supports the setting of channel trigger conditions. These conditions are collectively tracked by the SR192A hardware. An active high trigger signal is generated when the channel response data matches the specified pattern. The trigger signal can be selected, via the Cell Test Parameters dialog, as the source for timing set signal Test1 or Test2. Thus a timing set can synchronize its execution with the occurrence of a specific trigger condition.

Channel trigger levels are set in the **Trig** column of the Signal List Editor window. Double clicking the left mouse button causes these to toggle (e.g. X, L, H). By default, all channels are initially masked (X). High and low trigger levels are further indicated by blue and green backgrounds in the channel column. This is especially helpful when visually scanning a long signal list for trigger settings.

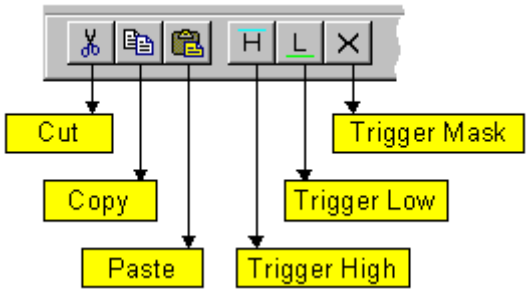
The trigger entries can also be drag selected and copied, pasted or set via the toolbar or Edit Menu. The whole trigger column is selected by a left mouse click on the **Trig** column heading.

Slot	Module Group	Channel [d]	Trig	Signal Name	Connector	UUT Connector
DRA1 SR126A	Grp123 Mode: Standard Response Strobe: TSES2	CH1	H	D0	J1-A1, J1-A2	
		CH2	L	D1	J1-A3, J1-A4	
		CH3	H	D2	J1-A5, J1-A6	
		CH4	L	D3	J1-A7, J1-A8	
		CH5	X	D4	J1-A9, J1-A10	
		CH6	X	D5	J1-A11, J1-A12	
		CH7	L	D6	J1-A13, J1-A14	
		CH8	X	D7	J1-A15, J1-A16	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.2 Toolbar

The toolbar, on the Signal List Editor window, provides quick access to commonly used commands. All of the buttons on the toolbar have corresponding menu entries. Many of them also have shortcut keys that perform the same command. The shortcut keys are shown to the right of their menu entries. Tooltips are displayed when the mouse cursor is held over a toolbar button for approximately two seconds. A tooltip box appears temporarily to identify the button's function.

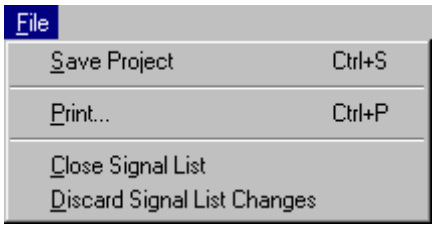


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 Menus

4.3.1 File Menu

The File Menu, on the Signal List Editor window, is used to update the project file with the latest changes. This causes all open windows to save their contents to the project file. Periodic saves are recommended as a precaution against editing errors or power failures. The Signal List Editor window is also closed from this menu. Use the Discard option to close the signal list without saving it.



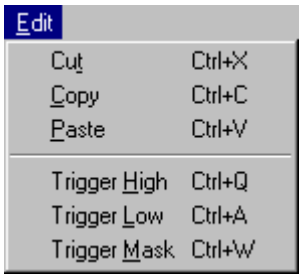
Menu Option	Description
Save Project	Causes all open windows to save their contents to the working project file. Then copies the working project file to the original project file.
Print…	Send an image of the current Signal List Editor to the printer. Also accomplished by pressing **Ctrl-P**.
Close Signal List	Close the Signal List Editor window and update the contents of the working project file.
Discard Signal List Changes	Close the Signal List Editor window and discard any changes that have been made.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3.2 Edit Menu

The Edit Menu, on the Signal List Editor window, is used to cut, copy and paste signal name and UUT connector information. These operations allow the information to be moved to or from other locations. These locations can be SR192A Development Environment windows or other text sources. This menu can also be accessed as a pop-up menu (right mouse click on the signal grid).

The trigger options set selected trigger cells to a specific pattern. They are also used to set channel type (TTL/Differential) for SR124A pin modules. A whole column is selected by a left mouse click on its title. Multiple items are selected by holding the left mouse button down and dragging.



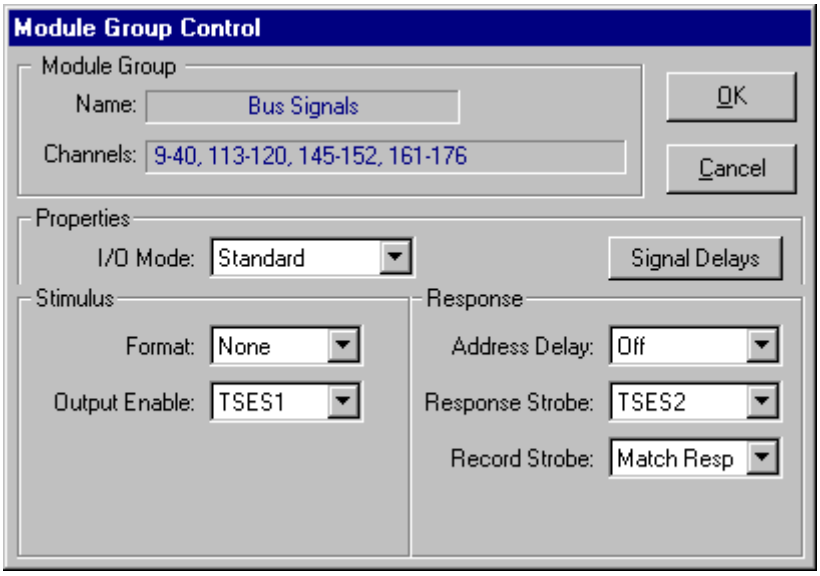
Menu Option	Description
Cut	Copy the selected rows to the system Clipboard and reset the row contents.
Copy	Copy the contents of the selected rows to the system Clipboard.
Paste	Paste the contents of the system Clipboard into the selected rows.
Trigger High	Set the selected trigger cells to a high level.
Trigger Low	Set the selected trigger cells to a low level.
Trigger Mask	Set the selected trigger cells to be masked (i.e. ignored).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4 Module Group Control

The Module Group Control dialog is used to view and edit module group settings. These are operational parameters that affect signal behavior. SR192A channels are assigned to module groups with the Signal List Editor. Different types of pin modules have different properties that can be controlled through a module group. These differences are reflected in the configuration of the Properties section of the Module Group Control dialog.

Pressing the OK button saves the current settings for the module group identified in the Name field. The Channels field identifies the SR192A channels that are associated with the referenced module group. If the channel list exceeds the size of the field, a tooltip box containing the whole list is displayed when the mouse cursor is held over the field. Use a shortcut key (**Ctrl-P**) to print an image of this dialog.

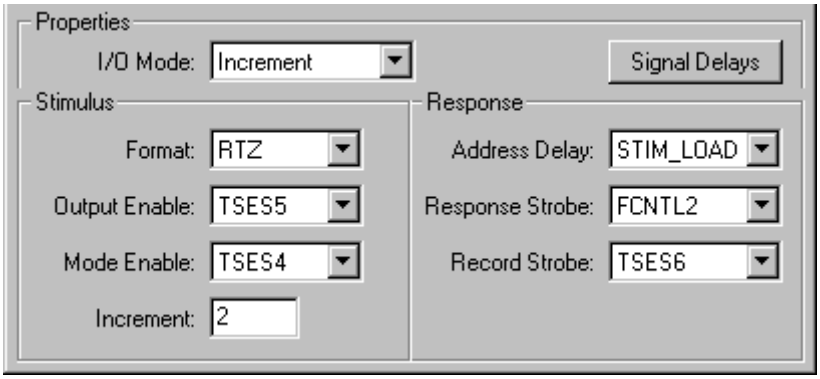


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4.1 Module Group Properties

The Properties section of the Module Group Control dialog contains the adjustable parameters for a set of SR192A channels. The contents and configuration of the Properties section is dependent upon the type of pin module associated with the channels. Pin modules are assigned to slots/channels with the SR Module Configuration window. The Signal Delays button activates a dialog for setting delays on certain control signals. If any signal delays are defined, the button caption is set to a **bold** font.

A set of drop-down lists are provided for selecting the operational characteristics of a module group. Note, some lists are dependent on the I/O mode and are only visible when applicable. Reference the SR192A documentation for further information about module group properties.

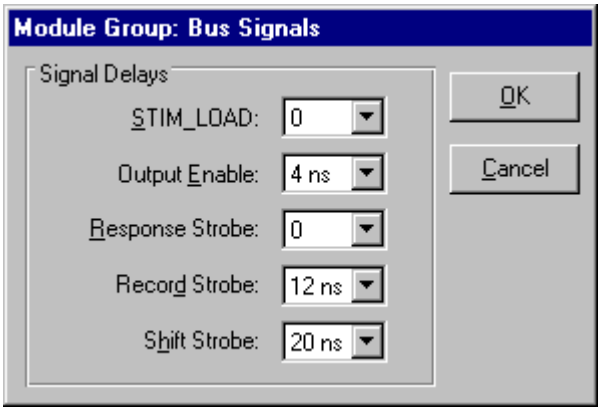


Note: After enable and strobe signals are defined, it is helpful to rename the signals in the Timing Set Editor to indicate their function. For example, based on the above dialog, TSES5 could be renamed "Output Enable" in the Timing Set Editor. This then provides a constant and useful reminder while editing timing set signals.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4.2 Signal Delays

The Signal Delays dialog is activated from the Module Group Control dialog in order to set control signal delays. Delays can be set from zero to 20 nanoseconds. If any delay is set greater than zero, the Signal Delays button caption is set to a **bold** font on the Module Group Control dialog. Use a shortcut key (**Ctrl-P**) to print an image of this dialog.



Note: Under most circumstances these delays can be left at zero. Also, some delays may not be applicable (e.g. Shift Strobe) depending on the contents of the Module Group Control dialog.

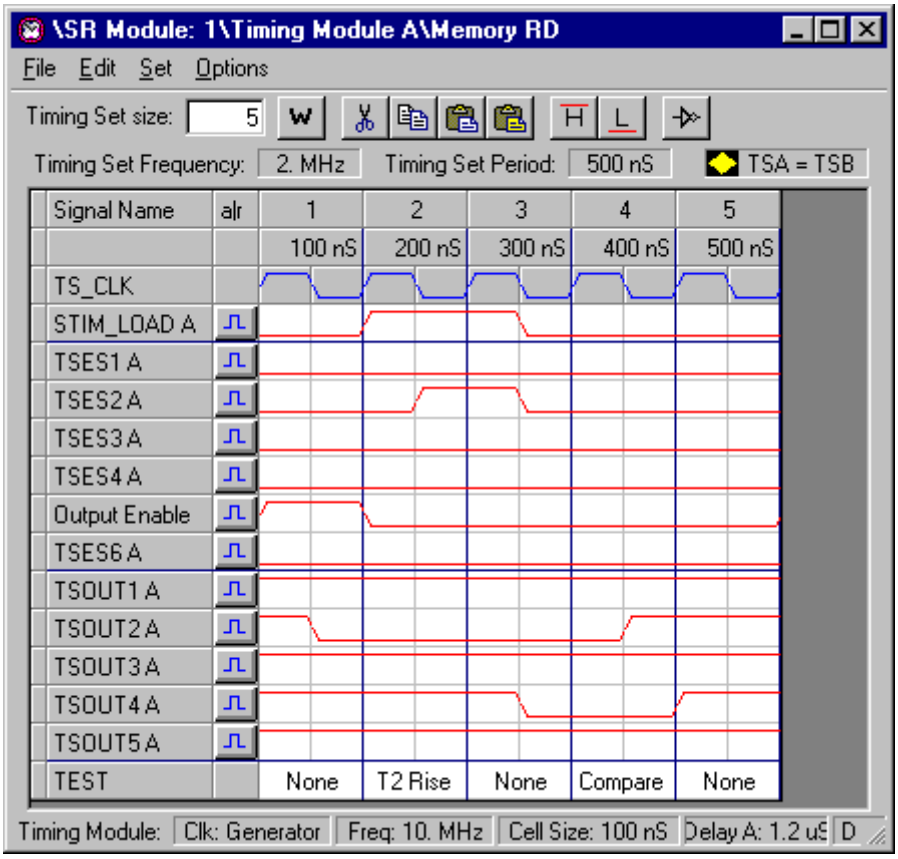
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5. Timing Set Editor

The Timing Set Editor is used to define the behavior of a timing set. A timing set controls digital data generation when an SR192A is executed. A timing set is cycled for each digital vector that is output. Behavior defined in a timing set controls when stimulus changes occur and when responses are sampled.

A common set of control signals are defined for a timing set. By default they are all initialized to an inactive state. Level changes are easily made by double clicking the left mouse button on a signal cell. Multiple cells are changed by drag selecting them and using the toolbar or Set Menu. A pop-up menu is also available (right mouse click on the signal grid).



Note: If timing generators A and B are linked, the control signals for both are combined into one timing set. The 'B' control signals are displayed with a gray background in order to differentiate them. To ensure proper operation, the 'B' control signals must be set appropriately. The Options Menu allows one set of signals to be used for both TSA and TSB. When this is active, a yellow diamond is displayed below the toolbar (as shown above).

The size (i.e. number of cells) of a timing set is entered in a text box on the toolbar. Pressing the Enter key causes the new size to be displayed. Below the toolbar are fields that show the frequency and period of the timing set. These indicate the rate at which the digital vectors will be

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

generated. Note that the rate also reflects any delays (Delay A or Delay B) that are specified within the timing set. A Cell Wizard is provided to assist in sizing a timing set based on clock frequency and the desired target frequency or period.

Along the top of the signal grid the columns are labeled with a cycle count and time period. TS_CLK, shown as a blue waveform, illustrates the phases of the timing set clock which aids the placement of other timing signals. Columns (phases) are selected by a left click on a top row. To the left of the signal names is a narrow column that selects a single row with a left click. Multiple rows are selected by holding the left mouse button down and dragging it along this column.

To the right of the signal names is a column of buttons labeled with a blue pulse. Pressing one of these buttons activates the Assert & Return Wizard. This wizard allows a sequence of assert and return times to be numerically entered for an associated signal. This is intended for users that are familiar with defining signals for simulation.

Timing signal names are edited by making a single left click in the appropriate cell. Pressing the Enter key accepts the changes and moves editing to the next lower cell. Signal names can be cut, copied and pasted from other locations in the SR192A Development Environment, or from many other text sources. A whole column is selected by a left mouse click on its title. Multiple items are selected by holding the left mouse button down and dragging.

The bottom row of the signal grid defines an optional test condition. When selected, a combo box presents the choices that are available for a timing cell test.

A status bar at the bottom of the window displays the clock source, clock frequency, computed cell size and delay counts. These parameters are all defined on the Timing Generators tab of the SR Module Configuration dialog. These fields are left blank if the information is not yet available.

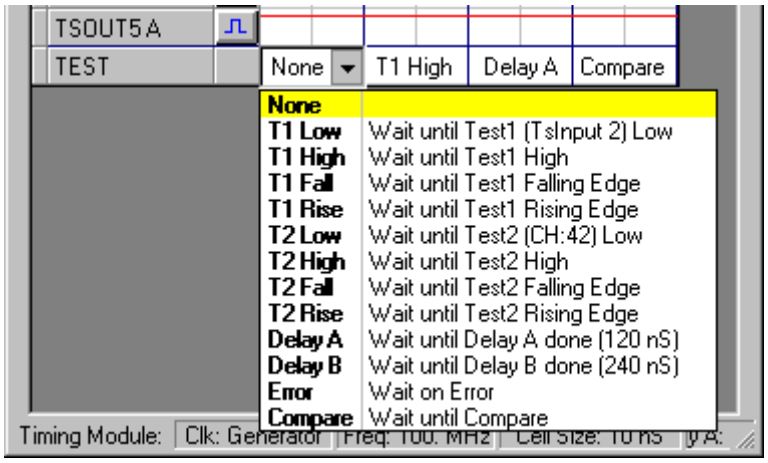
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.1 Timing Cell Test

The bottom row of the Timing Set Editor grid defines an optional test condition. A test condition can be set for each cycle of a timing set. These cause a timing set's execution to pause until the specified condition is met (e.g. Test1 at a high level) or a specified amount of time has elapsed (e.g. DelayA).

The behavior of these test conditions is dependent upon timing generator settings. Specifically, the test signal sources and delays are assigned via the Cell Test Parameters dialog accessed from the Timing Generators panel of the SR Module Configuration window.

Test conditions are easily selected via a combo list activated by a left mouse click. Note that the combo list contains the current delay values and the signals assigned to Test1 and Test2.

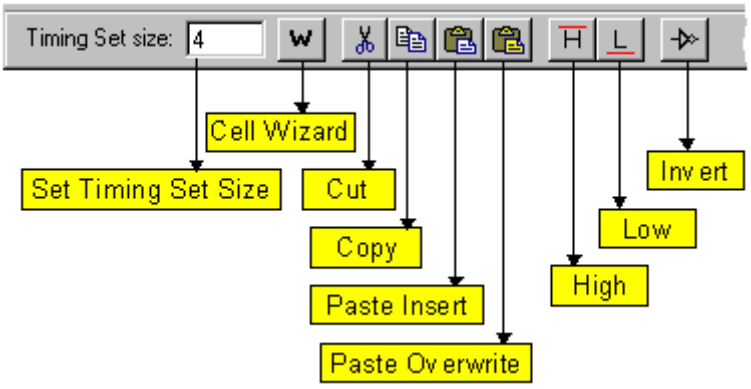


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.2 Toolbar

The toolbar, on the Timing Set Editor window, provides quick access to commonly used commands. It also contains a text box for changing the size (i.e. length) of the timing set.

All of the buttons on the toolbar have corresponding menu entries. Many of them also have shortcut keys that perform the same command. The shortcut keys are shown to the right of their menu entries. Tooltips are displayed when the mouse cursor is held over a toolbar button for approximately two seconds. A tooltip box appears temporarily to identify the button's function.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.3 Menus

5.3.1 File Menu

The File Menu, on the Timing Set Editor window, is used to update the project file with the latest changes. This causes all open windows to save their contents to the project file. Periodic saves are recommended as a precaution against editing errors or power failures. The Timing Set Editor window is also closed from this menu. Use the Discard option to close a timing set without saving it.



Menu Option	Description
Save Project	Causes all open windows to save their contents to the working project file. Then copies the working project file to the original project file.
Cell Wizard	Initiate a Cell Wizard dialog to adjust timing set size based on frequency or period.
Print…	Send an image of the current Timing Set Editor to the printer. Also accomplished by pressing **Ctrl-P**.
Close Timing Set	Close the Timing Set Editor window and update the contents of the working project file.
Discard Timing Set Changes	Close the Timing Set Editor window and discard any changes that have been made.

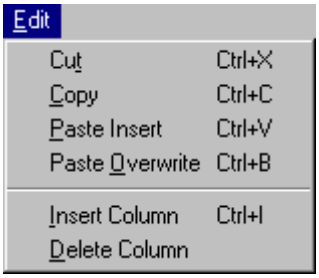
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.3.2 Edit Menu

The Edit Menu, on the Timing Set Editor window, is used to cut, copy and paste portions of a timing set. This can be used to adjust signal positions or to transfer behavior to another timing set. These operations can also copy all or part of the signal name column to move it to another timing set. Two types of paste operations allow copied signals to replace existing signals (overwrite) or be inserted between existing signals (insert). Only copied timing set information can be pasted in this window. Test conditions are not copied or pasted.

Individual signal cells are selected by a left mouse click. Multiple cells are selected by holding the left mouse button down and dragging. A whole column is selected by a left mouse click on its title. Multiple columns are selected by holding the left mouse button down and dragging it along the title row.

Several toolbar buttons and shortcut keys are provided to more easily initiate these operations. Some operations are also available on the pop-up menu (right mouse click on the signal grid).



Menu Option	Description
Cut	Copy the selected area to the system Clipboard, remove it and shift left the remaining area to its right.
Copy	Copy the contents of the selected area to the system Clipboard.
Paste Insert	Insert the complete contents of the system Clipboard into the grid, starting at the upper, leftmost selected cell.
Paste Overwrite	Paste the contents of the system Clipboard only into the selected area.
Insert Column	Insert a new, duplicate column to the left of the selected one.
Delete Column	Delete one or more selected columns. A column must be fully selected in order for it to be deleted.

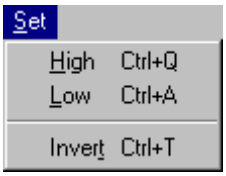
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.3.3 Set Menu

The Set Menu, on the Timing Set Editor window, is used to make selected signals high, low or inverted from their previous state. Signal states are also set by double clicking the left mouse button on an individual signal cell. Many of the signals in a timing set are active during a low state. Test conditions are only changed by double clicking the left mouse button.

Individual signal cells are selected by a left mouse click. Multiple cells are selected by holding the left mouse button down and dragging. A whole column is selected by a left mouse click on its title. Multiple columns are selected by holding the left mouse button down and dragging it along the title row.

Toolbar buttons and shortcut keys are provided to more easily initiate these operations. They are also available on the pop-up menu (right mouse click on the signal grid).



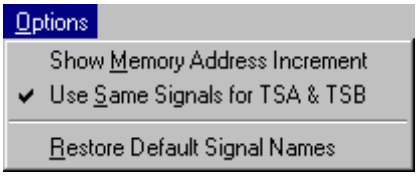
Menu Option	Description
High	Set the selected signal cells to a high level.
Low	Set the selected signal cells to a low level.
Invert	Toggle the level of the selected signal cells.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.3.4 Options Menu

The Options Menu, on the Timing Set Editor window, is used to control the display and operation of signals and labels. The Memory Address Increment (MA_INC) signal can be hidden because it is optional for most operating modes. Since timing set signal names can be edited, the last option on this menu allows them to be restored to their default labels.

When two timing generators are linked, a timing set contains signals for both TSA and TSB. Timing signals for TSB are displayed with a gray background in order to differentiate them. If the timing for TSA and TSB is the same, this menu allows a single set of timing signals to be applied to both. When this is active, a yellow diamond is displayed below the toolbar.



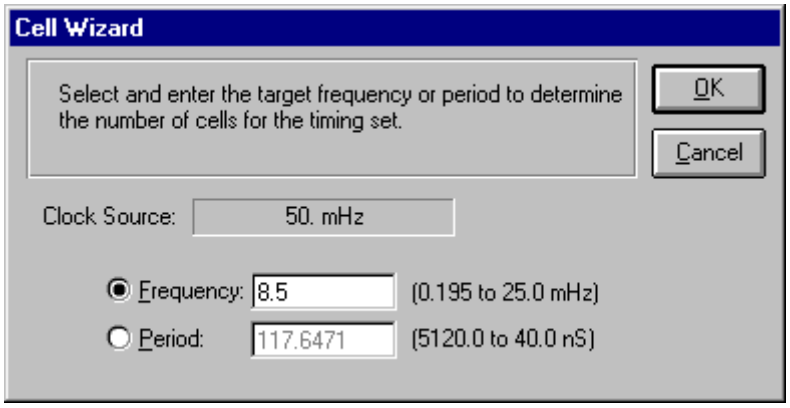
Menu Option	Description
Show Memory Address Increment	Show or hide the Memory Address Increment (MA_INC) signal.
Use Same Signals for TSA & TSB	For linked operation, use one set of timing signals for TSA and TSB.
Restore Default Signal Names	Replace modified signal names with their original labels.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4 Dialogs & Windows

5.4.1 Cell Wizard

The Cell Wizard is available from the Timing Set Editor to assist the user in achieving a specific data generation rate. It allows a target frequency, or period, to be entered and it adjusts the size of the timing set to meet that rate. The computation is based on the clock frequency specified for the associated timing generator. If the clock frequency is not yet specified, the Cell Wizard is disabled. Use a shortcut key (**Ctrl-P**) to print an image of this dialog.



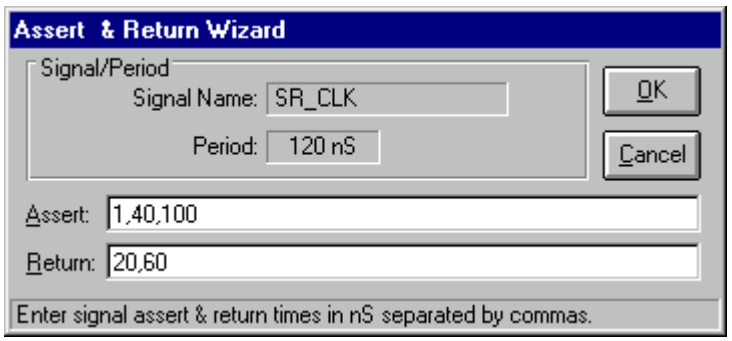
Note that valid timing ranges are displayed to the right of the text entry fields. These ranges are based on the current clock frequency and the range of timing set cells available (2 to 256).

> **Note: Using the Cell Wizard automatically deletes existing delay states (D) from the timing set.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4.2 Assert & Return Wizard

The Assert & Return Wizard is available from the Timing Set Editor to assist the user in specifying timing signal behavior. It uses a simulator-style format for defining when a signal transitions between active and inactive states. Intermediate values are rounded-down to the nearest cell edge. Use a shortcut key (**Ctrl-P**) to print an image of this dialog.



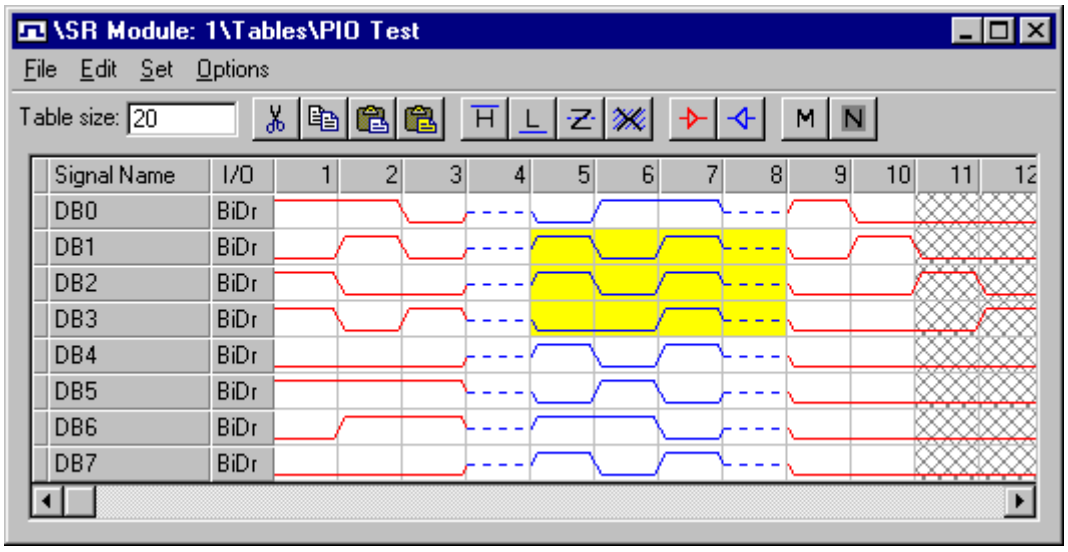
Note: Timing signals are active high. Therefore, an assert causes a signal to transition to a high state and a return sets it to a low state.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6. Table Editor

The Table Editor is used to define the behavior of a collection of digital signals. A data table is matched with a timing set when it is executed on an SR192A. The timing set is cycled for each digital vector that is output. The signals defined in a table are matched to SR192A channels with the Signal List Editor. Execution errors are automatically highlighted on a table when it is active during Test Manager operations.

By default, a table is set to 32 signals and 100 vectors. Vectors and signals are added or deleted with the Edit Menu. New signals are always initialized to an unknown state. Level changes are easily made by double clicking the left mouse button on a signal cell. Multiple cells are changed by drag selecting them and using the toolbar or Set Menu.



The size (i.e. number of vectors) of a table is entered in a text box on the toolbar. Pressing the Enter key causes the new size to be displayed. Along the top of the signal grid the columns are labeled with a vector count. If the count is too large for the column width, the full count appears when you hold the mouse cursor over the column heading. Columns are selected by a left click on the top row. You can quickly move to a distant vector with the '**Go to**' command on the Options Menu. This is also available on the pop-up menu (right mouse click on signal grid).

Signal conditions are identified by shape and color. Red is a stimulus (drive) signal and blue is a response (sense). The four signal states are high, low, tristate and unknown. The unknown state is used for signals whose behavior is indeterminate during a test. By default all signals are monitored (i.e. tested). Neglected signals are identified by a gray crisscross pattern in the cell.

Signal names are edited by making a single left click in the appropriate cell. Pressing the Enter key accepts the changes and moves editing to the next lower cell. Signal names can be cut, copied and pasted from other locations in the SR192A Development Environment, or from many other text sources. A whole column is selected by a left mouse click on its title. Multiple items are selected by holding the left mouse button down and dragging.

It is best to use logical or schematic names for signals rather than physical channel numbers. This isolates the signal data from potential changes in fixtures or instrumentation. The Signal List

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Editor is then used to match signal names to physical channels. When a project file is saved, a validation process is performed that matches table signal names to those on the signal list. If a matching signal is not found, a warning is generated. To avoid warnings on unused signals, preface those entries with two forward slashes (e.g. "// Ref Signal" or just "//"). It is often helpful to create unused signal rows for visual spacing or reference. A sync signal can be added to a table simply by naming a signal "**SR192A_Sync**". A signal with this name is automatically used by the ActiveX driver to program the sync memory.

Signals are easily repositioned by holding the left mouse button down on the cell to the left of the signal name. The cursor changes to indicate that you are dragging a signal. Release the mouse button to drop the signal in its new position. The signal grid automatically scrolls when you drag the signal near its top or bottom.

To the right of the signal names is a column of I/O indicators. Double clicking the left mouse button causes these to toggle (e.g. BiDr, Drv, Sens). This defines whether the signal is bidirectional, drive-only or sense-only. Setting the direction of a signal protects it from editing errors and clearly identifies its intended behavior. The whole I/O column is selected by a left mouse click on the column heading. Use the Unknown, Drive and Sense set commands to set to BiDr, Drv or Sens. When one or more complete signal rows are selected, changing their direction (Drive or Sense) also changes the I/O column contents.

A single bidirectional channel is capable of driving and sensing different levels during the same cell. This behavior requires that the drive and sense enables are at different points in the timing set. To define different drive and sense levels for a channel, a **doubled signal** should be created. This is accomplished by creating a drive signal and a sense signal that have the same name. This doubling then allows the drive and sense levels to be defined differently for the same cell.

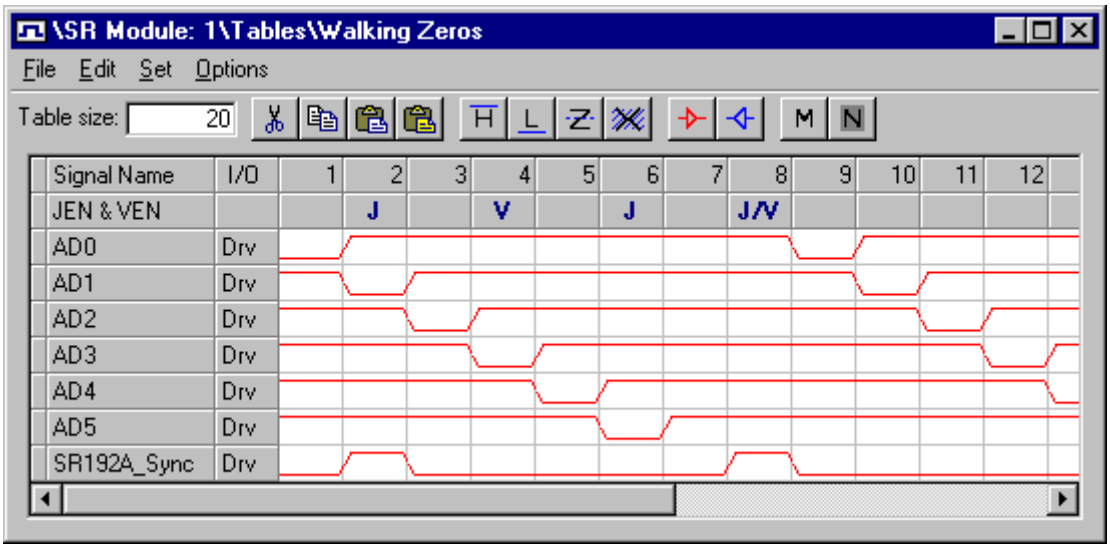
The flow of sequence execution is controlled by branching or indirect vector jumps. They occur at defined positions within a table based on the setting of a jump enable and/or vector enable bit. These are optionally displayed via the Options Menu.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.1 Jump Enable & Vector Enable

The Jump Enable (JEN) and Vector Enable (VEN) signals control branching during SR192A sequence execution. The JEN signal defines when branch conditions are evaluated. The VEN signal controls when branching is via the vector table defined with the Sequence Editor. Since these signals operate synchronously with a table's execution, they are viewed and edited with the Table Editor.

The JEN & VEN row is displayed or hidden via the Option Menu. Double clicking the left mouse button on a JEN/VEN cell causes it to toggle (None, J, J/V, V). The whole row is selected by a left mouse click in the I/O column. Use the High, Low, Tristate and Unknown commands to set the selected row to J, V, J/V or None.

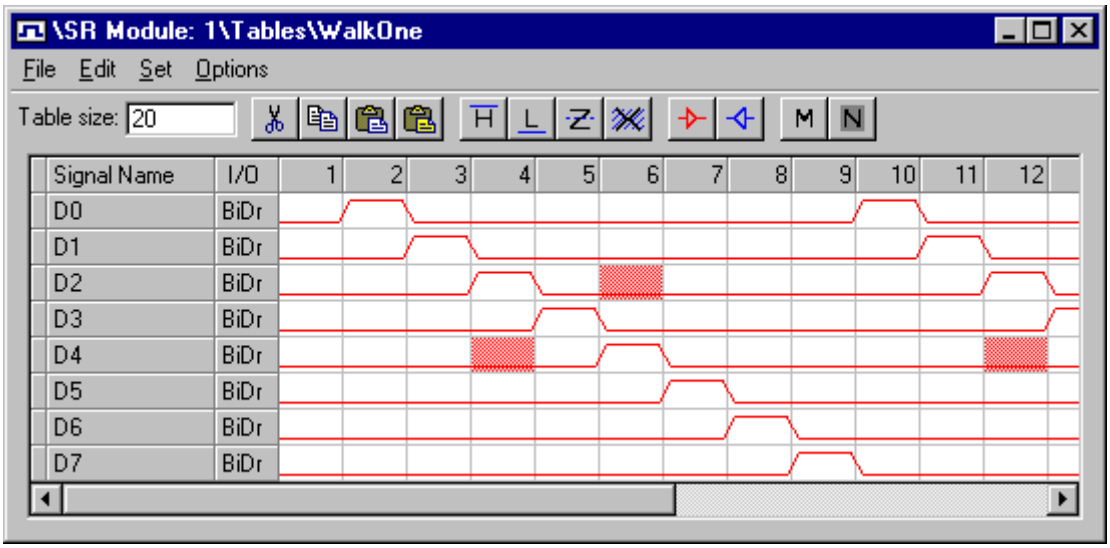


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2 Error Display

During test development and debug, the Table Editor automatically displays execution errors detected by the Test Manager. The execution errors are identified by a light red, or pink, background. The Errors submenu has options for finding, clearing and neglecting the highlighted errors.

After the Test Manager runs a test, an execution result file is created. This file contains the project file name and the detected data table failures. Based on this information, the Test Manager looks for an active SR192A Development Environment that has the same project file loaded. If one is found, it is informed that there are applicable execution results to display. Thus, active Table Editor windows are updated following each Test Manager run.



Note: this only displays the number of vector failures defined by the Test Manager's Execution Results dialog. It does not necessarily indicate the full scope of erring vectors.

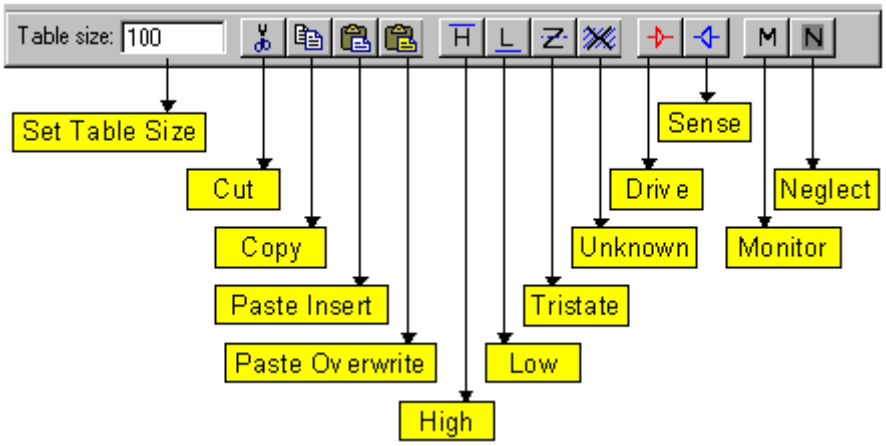
If the background color used to highlight errors is difficult to discern, it can be modified by editing the application's initialization file (SR192aDev.ini). Change the value for the "ErrorColor" entry to the desired RGB color code. A couple of possible standard colors include cyan (&H808000), light gray (&HC0C0C0), dark gray (&H808080) and light cyan (&HFFFF00). Delete the "ErrorColor" entry to return to the default background color. Be sure to edit the initialization file while the application is closed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.3 Toolbar

The toolbar, on the Table Editor window, provides quick access to commonly used commands. It also contains a text box for changing the size (i.e. length) of the table.

All of the buttons on the toolbar have corresponding menu entries. Many of them also have shortcut keys that perform the same command. The shortcut keys are shown to the right of their menu entries. Tooltips are displayed when the mouse cursor is held over a toolbar button for approximately two seconds. A tooltip box appears temporarily to identify the button's function.

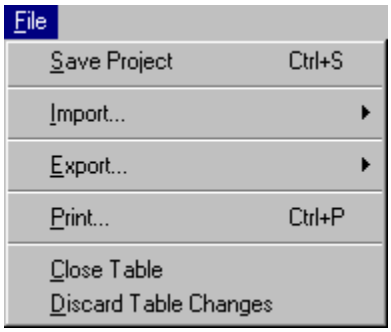


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4 Menus

6.4.1 File Menu

The File Menu, on the Table Editor window, is used to update the project file with the latest changes. This causes all open windows to save their contents to the project file. Periodic saves are recommended as a precaution against editing errors or power failures. The import and export options permit data to be moved in and out of the table. This allows data to be externally edited or automatically created. The Table Editor window is also closed from this menu. Use the Discard option to close a table without saving it.



Menu Option	Description
Save Project	Causes all open windows to save their contents to the working project file. Then copies the working project file to the original project file.
Import…	Import to the table, or selected cells, from the specified file type.
Export…	Export the table contents, or selected cells, to the specified file type.
Print…	Send an image of the current Table Editor to the printer. Also by pressing **Ctrl-P**.
Close Table	Close the Table Editor window and update the contents of the working project file.
Discard Table Changes	Close the Table Editor window and discard any changes that have been made.

Note: A progress bar is displayed while a large table is being saved. A Cancel button allows the save to be aborted and the Table Editor is subsequently reactivated. Use the Discard option whenever possible with large tables to avoid unnecessary updates to the working project file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.2 Import & Export Menu

The Import and Export Menus, on the Table Editor window, are used to read and write table data to various file formats. This allows table data to be created or consumed by third party or custom applications. The three formats supported are Value Change Dump (VCD), Simple Hex and FITS Expected State. VCD is a format that is often used by simulators and waveform viewers. The Simple Hex format is useful because it is easy to generate automatically or manually. The FITS Expected State format is used by the Serendipity Systems' Fault Isolation Tool Set for guided probe fault isolation.



Menu Option	Description
Value Change Dump…	Import or export the entire table from/to a VCD formatted file.
Simple Hex…	Import/Export the table, or selected cells, from/to a Simple Hex formatted file.
FITS Expected State…	Import/Export the entire table from/to a FITS Expected State formatted file.

Note: Large selection areas for Simple Hex import or export are easily achieved. First select one corner of the desired area with a left mouse click; then select the opposing corner with a Shift-left mouse click.

An optional utility is available for importing **LASAR** Tap files into an SR192A project file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

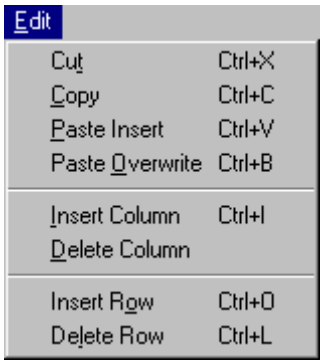
6.4.3 Edit Menu

The Edit Menu, on the Table Editor window, is used to cut, copy and paste portions of a data table. This can be used to adjust signal positions, duplicate behavior or to transfer data to another table. These operations can also copy all or part of the signal name column to move it to another table or to the Signal List Editor.

Two types of paste operations allow copied signals to replace existing signals (overwrite) or be inserted between existing signals (insert). Only copied table information can be pasted in the signal grid. Whole columns, or rows, must be selected before an insert or delete is allowed.

Individual signal cells are selected by a left mouse click. Multiple cells are selected by holding the left mouse button down and dragging. Large selections are quickly achieved by selecting one corner with a left mouse click and the opposing corner with a Shift-left mouse click. A whole column is selected by a left mouse click on its title. Multiple columns are selected by holding the left mouse button down and dragging it along the title row. A single row is selected by a single mouse click in the I/O column.

Several toolbar buttons and shortcut keys are provided to more easily initiate these operations.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

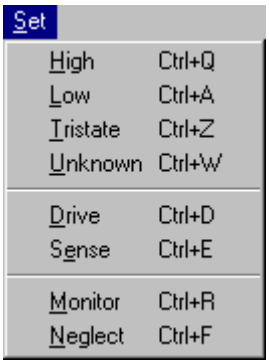
Menu Option	Description
Cut	Copy the selected area to the system Clipboard, remove it and shift left the remaining area to its right.
Copy	Copy the contents of the selected area to the system Clipboard.
Paste Insert	Insert the complete contents of the system Clipboard into the grid, starting at the upper, leftmost selected cell.
Paste Overwrite	Paste the contents of the system Clipboard only into the selected area.
Insert Column	Insert a new, duplicate column to the left of the selected one.
Delete Column	Delete one or more selected columns. A column must be fully selected in order for it to be deleted.
Insert Row	Insert an empty row above the selected one.
Delete Row	Delete the selected row. The signal below is moved up one row and selected. This allows rapid deletion of multiple rows when used with **Ctrl+L**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.4 Set Menu

The Set Menu, on the Table Editor window, is used to control level, direction and monitoring of digital signals. The set commands operate on the selected signal cells. Signal levels are also set by double clicking the left mouse button on an individual signal cell.

Individual signal cells are selected by a left mouse click. Multiple cells are selected by holding the left mouse button down and dragging. Large selections are quickly achieved by selecting one corner with a left mouse click and the opposing corner with a Shift-left mouse click. A whole column is selected by a left mouse click on its title. Multiple columns are selected by holding the left mouse button down and dragging it along the title row. A single row is selected by a single mouse click in the I/O column. Toolbar buttons and shortcut keys are provided to more easily initiate these operations.



Menu Option	Description
High	Set the selected signal cells to a high level.
Low	Set the selected signal cells to a low level.
Tristate	Set the selected signal cells to a mid level (tristate).
Unknown	Set the selected signal cells to an unknown (don't care) state. This becomes tristate/masked in the SR192A hardware.
Drive	Set the selected signal cells to drive (i.e. into the UUT).
Sense	Set the selected signal cells to sense (i.e. read from the UUT).
Monitor	Set the selected signal cells to be monitored (i.e. tested).
Neglect	Set the selected signal cells to be neglected (i.e. not tested).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Note: A progress bar is displayed during a large set operation. A Cancel button allows the operation to be aborted and the Table Editor is subsequently positioned to show the last row modified.

The Unknown state is used for signals whose behavior is indeterminate during a test. Driving channels are automatically set to tristate and sense channels are neglected (i.e. masked). This visually different state allows a user to uniquely identify indeterminate or untestable behavior.

The direction of a signal is changeable if it is designated as bidirectional (BiDr in the I/O column). Otherwise, the direction commands are ignored. Set the I/O column for a signal if you need it to always drive or sense. This protects it from editing errors and clearly identifies its intended behavior. Driving signals are shown in red, sense signals are shown in blue.

By default, all signals are monitored (i.e. tested). Use the Neglect command to disable testing for all or part of a signal's operation. Neglected signals are identified by a gray crisscross pattern in the cell.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.5 Options Menu

The Options Menu, on the Table Editor window, is used to jump to a different location in the table or change the state of selected signal cells. Many of these commands are also available from a pop-up menu (right mouse click on the signal grid). This menu also provides a way to display the JEN/VEN row which controls branch conditions during sequence execution.



Menu Option	Description
Go to…	Prompts for a vector number and shifts the signal grid to display it. Also accomplished by pressing **Ctrl+G**.
Invert	Inverts the state (level) of the selected signal cells. Also accomplished by pressing **Ctrl+T**.
Fill	Activates a submenu with options for setting patterns in a selected block of signal cells.
Errors	Activates a submenu for locating, clearing or neglecting execution errors.
Show JEN & VEN	Display or hide the Jump Enable & Vector Enable row.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.6 Fill Menu

The Fill Menu, on the Table Editor window, is used to set selected signal cells to a specified pattern. These commands are also available from a pop-up menu (right mouse click on the signal grid).



Menu Option	Description
Increment	Set the level of the selected signal cells to an incrementing pattern (topmost signal is least significant). Prompt for increment step.
Ramp	Set the level of the selected signal cells to a ramp pattern.
Random	Set the level of the selected signal cells to a random pattern. Prompt for randomizing seed value.
Rotate	Set the level of the selected signal cells to a rotating pattern based on the leftmost selected column. Prompt for a rotational increment.
Repeat	Set the level of the selected signal cells to match the leftmost selected column.
Toggle	Set the level of the selected signal cells to a toggling (on/off) pattern.

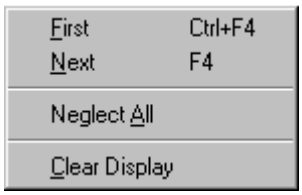
Note: A progress bar is displayed during a large fill operation. A Cancel button allows the operation to be aborted and the Table Editor is subsequently positioned to show the last column modified.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.7 Errors Menu

The Errors Menu, on the Table Editor window, is used to locate, neglect or clear displayed execution errors. The error display correlates execution results with a data table's contents in order to aid test development and debugging. Execution results and display activation are provided by the Test Manager.

When automatically locating an error cell, the cell to the right of the error is selected (i.e. yellow highlight). This provides an additional visual indication of which error has most recently been located. The error navigation commands are also available via shortcut keys (**Ctrl-F4** and **F4**).



Menu Option	Description
First	Shifts the signal grid to display the first execution error. Also accomplished by pressing **Ctrl+F4**.
Next	Shifts the signal grid to display the next sequential execution error. Also accomplished by pressing **F4**.
Neglect All	Sets all error cells to be neglected (i.e. not tested).
Clear Display	Removes error highlighting from table.

Note: Errors are only displayed for the number of vector failures defined by the Test Manager's Execution Results dialog. Displayed errors do not necessarily indicate the full extent of erring vectors.

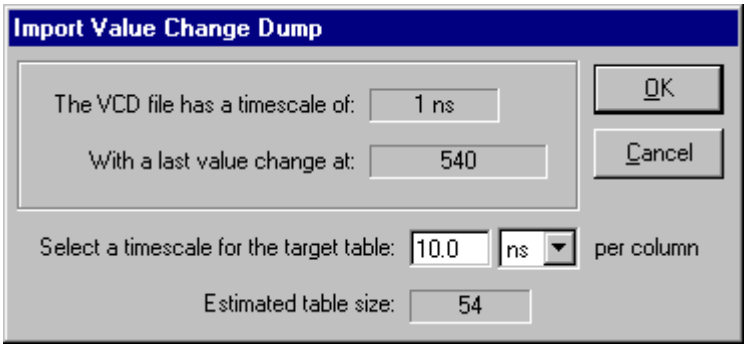
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5 Dialogs & Windows

6.5.1 Import Value Change Dump

When importing a Value Change Dump file, via the Import/Export Menu, this dialog is displayed after an input file is selected. The contents of the VCD file are scanned to determine and display its timescale and length. Based on the scanned information, a suggested scaling value is displayed along with the projected table size.

The default scaling parameter is based on the minimum time between successive value changes in the file. Typically this results in a reduced table size without loss of signal states. If the scale factor causes value changes to be written into the same table location, a warning is written to the Debug Log and the user is notified at the conclusion of the import.

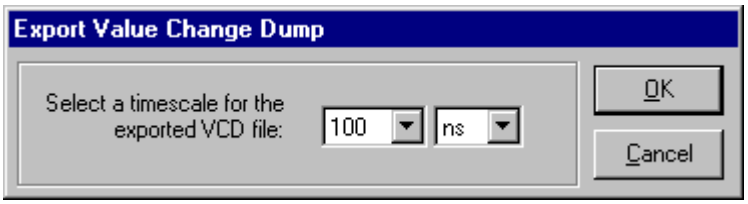


Note: In the Debug Log window, a double-click of the left mouse button on a VCD warning causes the Table Editor to display the corresponding signal and location. Moving up or down with cursor keys also accomplishes this.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.2 Export Value Change Dump

When exporting a Value Change Dump file, via the Import/Export Menu, this dialog is displayed after an output file is selected. Its purpose is to select a timescale for the VCD file being created.



Note: Only named signals are exported to a VCD file. Unused signal entries (i.e. "//") are ignored.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6 Import & Export File Formats

6.6.1 Value Change Dump File Format

A value change dump (VCD) file contains information about value changes on variables (i.e. signals). It is one of the formats supported for importing and exporting digital data from the Table Editor window. VCD files are often generated, or used, by simulation and waveform viewing applications. Many low-cost or free tools are available for defining or viewing VCD files.

The VCD file format is fully defined in Section 15 of IEEE Standard 1364-1995. It essentially consists of a definitions section and a value change section. The definitions section includes time/date information, VCD writer version, time scale and variable declarations. The value change section lists time increments (e.g. #500) and value changes corresponding to that time.

The VCD format allows a signal to change state within a single time increment (e.g. a momentary pulse or signal spike). This generates an import warning that identifies two states for a signal in the same vector (double-click warning on the Debug Log window to highlight corresponding vector in table).

> **Note: The occurrence of $dump commands can affect the import results. For example, a $dumpoff at the end of a VCD file turns all signals to unknown (X) for the last column of the table. It is always important to examine the contents of a VCD file when questions arise about the import result.**

The following example illustrates the format of a VCD file.

```
$date
      June 26, 2001 10:05:41
$end
$version
      VERILOG-SIMULATOR 1.0a
$end
$timescale
      10 ns
$end
$scope module top $end
$scope module m1 $end
$var trireg 1 *@ net1 $end
$var trireg 1 *# net2 $end
$var trireg 1 *$ net3 $end
$upscope $end
$scope task t1 $end
$var reg 32 (k accumulator[31:0] $end
$var integer 32 {2 index $end
$upscope $end
$upscope $end
$enddefinitions $end
$comment
      Note: $dumpvars was executed at time '#500'.
            All initial values are dumped at this time.
$end

#500
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
$dumpvars
x*@
x*#
x*$
bx (k
bx {2
$end
#505
0*@
1*#
1*$
b10zx1110x11100 (k
b1111000101z01x {2
#510
0*$
#520
1*$
#530
0*$
bz (k
#535
$dumpall 0*@ 1*# 0*$
bz (k
b1111000101z01x {2
$end
#540
1*$
#1000
$dumpoff
x*@
x*#
x*$
bx (k
bx {2
$end
#2000
$dumpon
z*@
1*#
0*$
b0 (k
bx {2
$end
#2010
1*$
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.2 Simple Hex File Format

The Simple Hex file format is used to import and export hexadecimal digital data from the Table Editor window. It is an uncomplicated ASCII format that can be viewed and modified with a variety of utilities and editors. Its straightforward syntax is conducive to automatic generation by CAD, simulation or translation applications.

The first line of a Simple Hex file contains a header that identifies its format. The header is enclosed in square brackets: **[SR192A Simple Hex]**

Each hexadecimal line in the file represents one column of a table. The rightmost bits correspond to the topmost signals. For increased readability, white space and blank lines may be used to separate the hexadecimal data. Comments are preceded by double slashes (//) and they can follow the data or be on a separate line.

The following is a portion of an exported Simple Hex file with the addition of comments and blank lines:

```
[SR192A Simple Hex]
AC 41 65 96
31 15 AD B7
56 40 59 1B
22 20 9A 7C
ED 8B ED F3
C7 E5 39 D3
0C B4 8E 7B
4B 7D 48 17
ED DF 06 61
B5 0A 6B BD

// Blank lines and comments are allowed
E5 D3 0B 9B
3C 20 1F FF
B2 2F E6 3A
D9 51 33 C5
99 7C 95 87   // comments are allowed here also
F1 96 BD B3
8B 48 AC 69
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.3 FITS Expected State File Format

When importing an Expected State file all signals are automatically defined as monitored, sense signals. If threshold levels are defined for a signal, they are included as part of the signal name (e.g. "Data2;4.2;1.3"). Thresholds encoded this way are understood by the Expected State export. The length of imported Expected State signals are limited to 256K due to the maximum table size of the SR192A.

When exporting an Expected State file, neglected signals are converted to the unknown (i.e. don't care) state. Thus the Table Editor can be used to easily define the portions of a signal that should not be evaluated during guided probe fault isolation. Measurement thresholds can be added to the signal name (e.g. "U13.2;4.0;0.8") in order to have them included in the resultant Expected State file.

The following section on the Expected State file format is extracted from the FITS User Manual.

Expected State File (EXP) Format

An expected state file contains node-specific data describing the correct behavior of a UUT for a given test. These files are created/modified via the FITS Guided Probe Browser. They are given an EXP extension and are typically named for the test used to generate them. For example, the expected state file created using the test program (DEMO.EXE) is named (DEMO.EXP).

The format of an expected state file describes the behavior of one node on each line. This line-oriented format includes thirteen semi-colon delimited fields that define the following eight types of node data. Six fields are not currently used, but are included to provide room for expansion:

Field	Node Data	Field	Node Data
1)	Node Label	8)	Field Not Used
2)	High Threshold	9)	Field Not Used
3)	Low Threshold	10)	Field Not Used
4)	Field Not Used	11)	Field Not Used
5)	Signature *	12)	Field Not Used
6)	First Transition Step **	13)	Waveform
7)	Starting Level		

* The Signature field contains an octal value that describes the node's behavior. It is typically a CRC that is calculated from the nodal activity. This field is optional and some hardware doesn't support it.

** The First Transition Step field contains an octal value representing the test step of the first transition (i.e. change in level). The first digit of this field represents the starting level of the node (0 or 1).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

By examining the expected state file format of these sample lines:

```
3C.2;2.0;.8;;004592;100010;1;;;;;;7H,3L,30X,10Z
3D.9;2.0;.8;;032467;100015;1;;;;;;12H,1L
4A.6;2.0;.8;;100005;000002;0;;;;;;1L,1H
4B.1;2.0;.8;; 100005;000002;0;;;;;;1L,1H
```

these observations can be made:

- All four nodes have a High Threshold of "2.0" and a Low Threshold of ".8".
- The node, '3C.2', has a Waveform of "7H,3L,30X,10Z". This means that the node's level was high for the first seven test steps, then low for three test steps, unknown for thirty test steps, and tristate for the final ten test steps.
- The node, '3D.9', has a Signature of "032467", a First Transition Step of "100014", and a Starting Level of "1". The waveform for this node (12H,1L) only describes behavior up to the first transition.

Note that the Waveform field either includes behavior for the complete test or behavior up to the first transition only. In the above example, node '3C.2' has the only complete waveform; the other three waveforms only have up to the first transition.

Guided probe isolation is the most efficient if there are complete waveforms for each node. Sometimes the expected state file is built from existing data sets which may not have complete waveforms. Under those circumstances, the guided probe operates as best as it can with the data available.

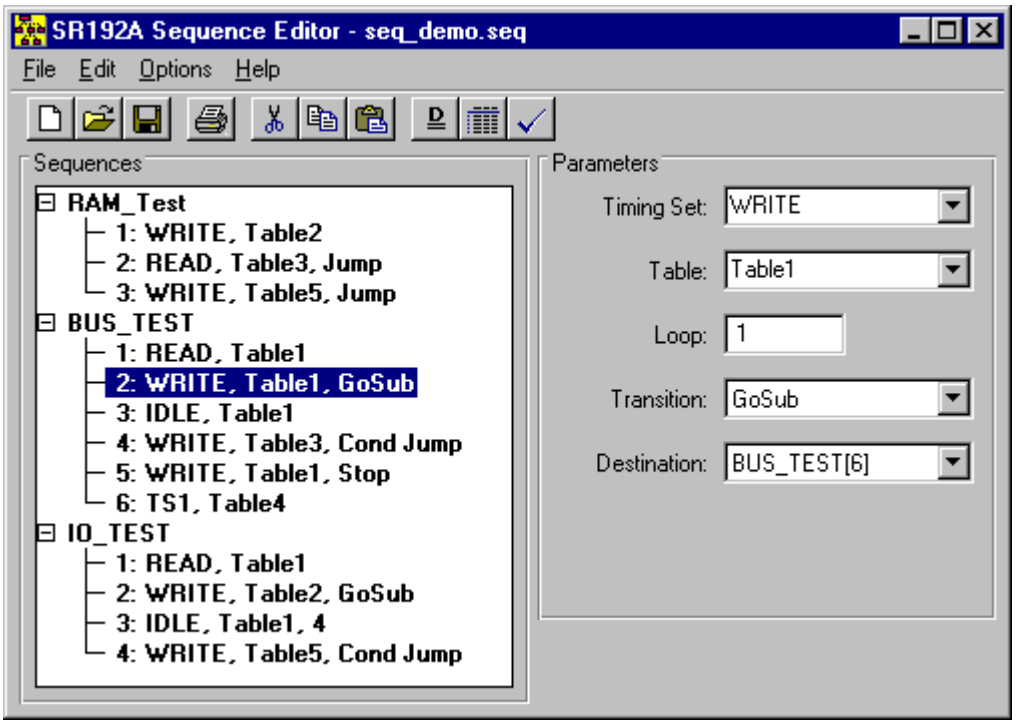
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7. Sequence Editor

The Sequence Editor is one of the software components that comprise the SR192A Development System. The Sequence Editor provides an easy-to-use interface for creating and editing execution sequences for the Talon SR192A. Sequences are loaded and executed by the Test Manager and programming interface.

Sequences are collections of timing set and table pairs (i.e. subsequences) that are executed in a defined order. Their execution order is controlled through conditional jumps and subroutine calls. Each subsequence has a separate count that allows it to be looped up to 32768 times. Building sequence definitions by hand can be tedious and error prone. The Sequence Editor is designed to quickly capture sequence requirements, even while providing a great degree of visibility and flexibility.



The Sequence Editor allows sequences to be edited, validated and stored to a file. Sequences containing one or more subsequences are presented in the Sequence List, a hierarchical list for viewing and editing. Subsequences are added, deleted, copied and pasted in the hierarchical list. Subsequences are each composed of a timing set name, table name, loop count and transition type. A branch (i.e. jump) transition also includes a destination and optional condition code. Defining a subsequence, with the Sequence Editor, is a simple matter of selecting these elements from validated lists in the Parameter Pane. In addition, lists of timing set and table names can be imported from SR192A project files.

When started, the Sequence Editor resumes the position it had during its previous execution. It also automatically reloads the last sequence file that was edited. The Sequence Editor can be moved, resized, minimized and maximized via standard Windows interface operations.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.1 Sequence Operation

Sequences are a very powerful mechanism for controlling the execution of an SR192A. This capability is provided by one or two Timing Set Modules that are installed in the SR192A.

The purpose of sequences is to coordinate the execution of multiple timing set and table pairs. This allows a series of operations to be executed, such as sequential read/write cycles on a bus. With the capability of conditional branching and subroutines, the sequences can generate complex series of patterns and behavior.

Sequences are composed of one or more subsequences. A subsequence represents the execution of a timing set and table pair. For every word in the table the timing set is executed once. Each subsequence can be repeated up to 32768 times.

> **Note: Table and timing set names are truncated to ten characters when they are downloaded to the SR192A hardware. Sequences must reference the truncated names.**

When a subsequence has completed its execution, there is an automatic transition to the next subsequence. During that transition, the **IDLE** timing set is executed. Alternately, a branch (jump) operation can be used to move execution to a different subsequence. A subsequence with a branch operation executes the first word of its corresponding table and then jumps to the destination subsequence.

Conditional branching is available for the following test conditions: Test1 level/edge, Test2 level/edge, Response Error, Timeout and Jump Enable. Branch conditions are only tested during the execution of words in the table whose jump enable bit is set. The jump enable bit can be set individually for each word in a table. This provides additional control over the sequence execution.

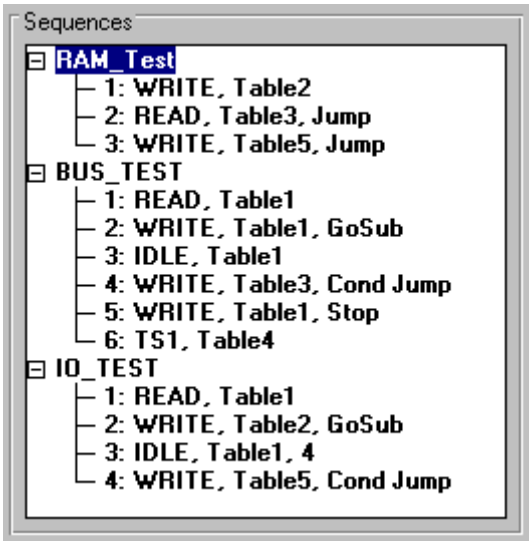
Another form of branching is a subroutine call (GoSub). A subsequence with a GOSUB executes a single table word and then branches to execute the destination subsequence. Upon completion of the destination subsequence, the next table word of the original subsequence is executed. The destination subsequence is executed after every table word in the original subsequence. Subroutine branching can be made conditional with the same conditions outlined above for jumps.

The SR192A is also capable of indirect branching via the vector table and specified vector channels. When the vector enable bit is active in a table, a branch or GoSub transition uses the vector channel levels to index into the vector table in order to locate a destination.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2 Sequence List

The Sequence List is used to manage the order of sequences and subsequences. It provides a hierarchical view; where each sequence is identified by a name and related subsequences are indented underneath. Subsequences are displayed with Timing Set name, Table name and transition type. If Display Details is checked on the Options Menu, subsequence entries also include destination and condition, when appropriate.



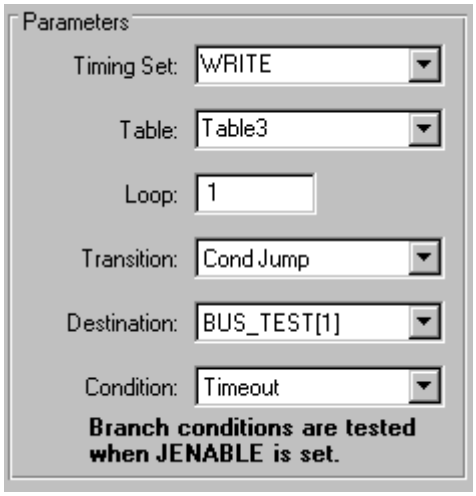
Scroll bars appear at the bottom and on the right side of the Sequence List if any text extends beyond those boundaries. Selecting an item in the Sequence List causes its contents to be displayed for editing in the Parameter Pane. Changes made in the Parameter Pane are automatically updated in the Sequence List when Enter is pressed or a selection is made from a drop-down list.

The Edit Menu has standard editing operations such as Cut, Copy and Paste. These are used to arrange the sequences and subsequences into their desired order.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.3 Parameter Pane

The Parameter Pane is used to view and edit the properties associated with a sequence or subsequence. The format of the Parameter Pane presented is dependent upon the item selected in the Sequence List. For a sequence, the Parameter Pane contains a single text box for changing its name. For a subsequence, the Parameter Pane has up to six fields for setting its properties.



For a subsequence, the top two fields define the Timing Set and Table names. These are selected from drop-down lists. Drop-down lists are activated by clicking the left mouse button on the down arrow button to the right of the text. These two name lists are loaded by importing names from timing and table files with the File Menu. New Table and Timing Set names can also be added to the list by typing them in their respective boxes and pressing Enter. To remove names from these lists, use the Remove Name command on the Edit Menu.

The next field defines the number of times to loop the subsequence. This value can be from 1 to 32768. Values outside of this range are automatically converted to the closest limit.

The Transition field defines how the sequence execution will proceed. To continue execution with the following subsequence, select a Next transition. To halt after executing one word from the table, select Stop. Branching to other subsequences is achieved by the Jump and GoSub selections. The Destination field contains a list of all of the subsequences to branch to. The UNKNOWN label is used if a subsequence has not yet been defined, or has been subsequently removed. If a conditional branch is selected, the Condition field appears. Branch conditions include input signals, a response error, a time-out and when allowed by the table's jump enable bit (JEN).

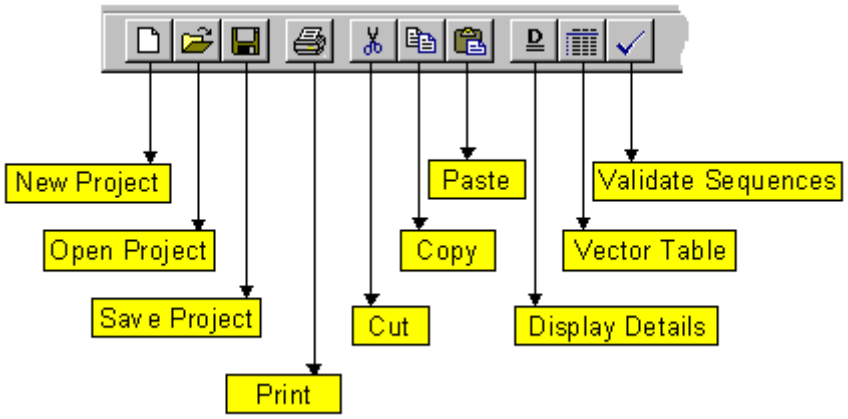
> **Note: Branch conditions are only tested during words whose jump enable bit is set. See the Jump Enable & Vector Enable section for more information on controlling tables' jump and vector enable bits.**

Sequence, timing set and table names can be up to ten alphanumeric and underscore characters. The first character must be an alpha character. The Sequence Editor automatically truncates these names to ten characters and replaces spaces by underscores. If a numeric or underscore is used as the first character, an 'N' is added to the beginning of the name.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.4 Toolbar

The toolbar, on the Sequence Editor window, provides quick access to commonly used commands. All of the buttons on the toolbar have corresponding menu entries. Many of them also have shortcut keys that perform the same command. The shortcut keys are shown to the right of their menu entries. Tooltips are displayed when the mouse cursor is held over a toolbar button for approximately two seconds. A tooltip box appears temporarily to identify the button's function.



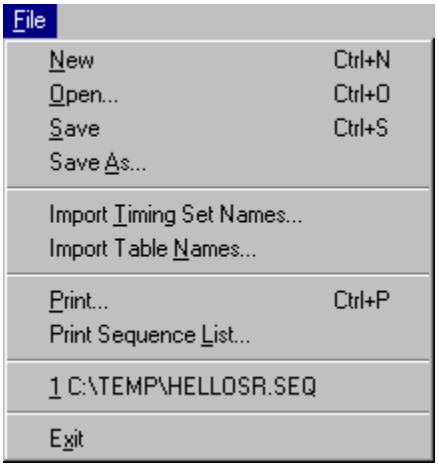
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.5 Menus

7.5.1 File Menu

The Sequence Editor File Menu supports selection and management of sequence files, timing set and table lists. It also provides a list of recently accessed sequence files and includes a way to exit the Sequence Editor.

Sequence files are stored in a simple ASCII text format. The imported timing set and table names are used in drop-down lists for configuring subsequences.



Menu Option	Description
New	Create and open a new sequence file.
Open…	Open an existing sequence (SEQ) file.
Save	Save the loaded sequence file.
Save As…	Save the loaded sequence file with a new name.
Import Timing Set Names…	Load Timing Set list with names extracted from a timing set file.
Import Table Names…	Load Table list with names extracted from a table file.
Print…	Send an image of the Sequence Editor to the printer. Also accomplished by pressing **Ctrl-P**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

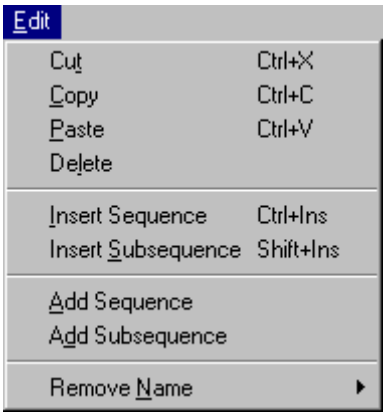
Print Sequence List…	Send an image of the Sequence List to the printer.
File History	Open one of the last four sequence files accessed by the Sequence Editor.
Exit	Close the application. If editing changes have been made the operator is prompted to save the sequence file. File history, display options and Window position are saved to SEQEDIT.INI.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.5.2 Edit Menu

The Sequence Editor Edit Menu allows the user to build and modify sequences. This includes cutting, copying and pasting sequences or subsequences. New items are added or inserted in the hierarchy. Additionally, the Timing Set and Table name lists are accessed from this menu. It is important to keep these lists current in order to best use the validate sequences command. References to unknown tables cause errors when a sequence is downloaded to an SR192A.

The copy and paste operations can also move sequences and subsequences between different files. Simply copy a sequence in one file and load another file to paste it into.



Menu Option	Description
Cut	Remove the selected items from the Sequence list and place them in the paste buffer. Cutting a sequence also includes its subsequences.
Copy	Copy the selected items in the Sequence list to a paste buffer. A copy of a sequence includes its subsequences.
Paste	Insert the contents of the paste buffer above the selected item in the Sequence list. This option is disabled when there is nothing in the paste buffer. Also disallowed is pasting a subsequence at the top of the Sequence list.
Delete	Permanently delete the selected items from the Sequence list. Deleting a sequence also deletes all of its subsequences.
Insert Sequence	Insert a new sequence above the selected item in the Sequence list.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

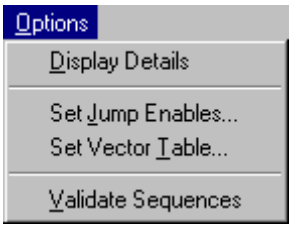
Insert Subsequence	Insert a new subsequence above the selected item in the Sequence list.
Add Sequence	Add a new sequence to the bottom of the Sequence list.
Add Subsequence	Add a new subsequence beneath the selected item in the Sequence list. If the selected item is a sequence, the new subsequence is added to the bottom of its subsequence list.
Remove Name	Select timing set or table names to delete with the Remove Name dialog.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.5.3 Options Menu

The Sequence Editor Options Menu allows the user to toggle a detail display option, set vectored destinations and validate sequences. These commands are also available on the toolbar.

Prior to loading an SR192A with sequences, it is highly recommended that the sequences be validated from this menu. Sequence validation includes verifying timing set and table names against imported lists. The existence of Jump and GoSub destinations are also verified. If a validation error is encountered, the operator is notified and has the choice of continuing validation or canceling.



Menu Option	Description
Display Details	Select this item to display all subsequence information in the Sequence list. When not selected, subsequence information is displayed in a minimal format.
Set Jump Enables…	Initiate a message box to inform user that jump enables are now set via the Table Editor.
Set Vector Table…	Initiate the Vector Table dialog to select destinations for vectored transitions.
Validate Sequences	Validate timing set and table names against imported lists. Also verify Jump and GoSub destinations.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

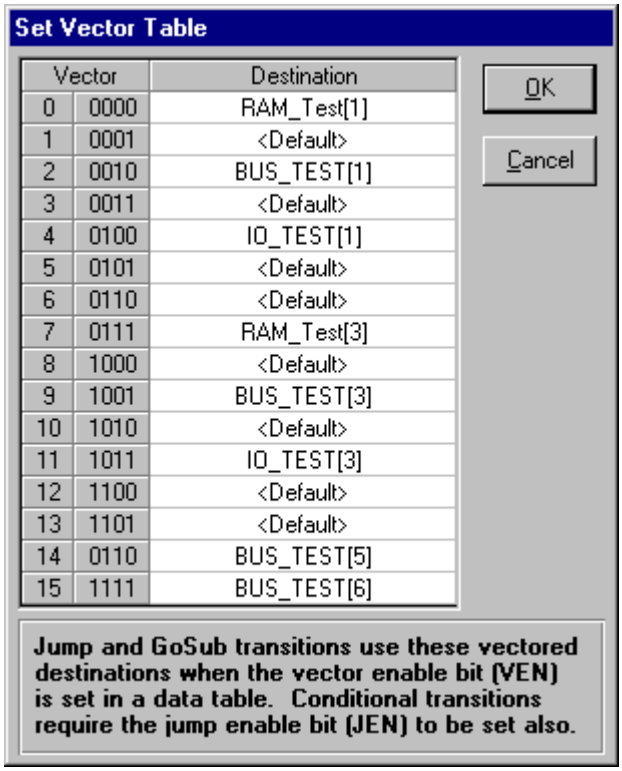
7.6 Dialogs & Windows

7.6.1 Vector Table

The Vector Table dialog box is for viewing and editing vectored destinations. It is activated from the Sequence Editor's Option Menu or toolbar. The vector table provides vector (indirect) branching for sequence execution. The subsequence destinations are selected from a drop-down list. Use a shortcut key (**Ctrl-P**) to print an image of this dialog.

Several steps are required to configure the SR192A for vector branching during sequence execution. The first step is to use the Advanced Settings dialog to select up to four channels for indexing into the vector table. If less than four channels are required, use a value of zero to disable the unnecessary bits.

The next step is to set vector enable bits in the table locations where vector branching is desired. Note that vector branching can be conditional if the jump enable bit is also set in the table. Finally, the vector table is used to set the indirect branch destinations.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.6.2 Remove Name

The Edit Menu presents the choice of deleting a name from the current Timing Set list or the current Table list. When either option is selected, a Remove Name dialog box appears with a drop-down list of the corresponding names. Pick a name and press the OK button to remove the name from the list.



Names are added to the Timing Set and Table lists by importing them from external files or by entering them in the Parameter Pane. This dialog allows names to be removed that are invalid, misspelled or that no longer apply. These lists are used during validation to ensure that each subsequence references a known timing set and table.

> **Note: It is very important to keep these lists matched to the tables and timing sets that will be present in the SR192A. Otherwise, a reference to a nonexistent entity can cause a sequence download to fail.**
>
> **Note: Table and timing set names are truncated to ten characters when they are downloaded to the SR192A hardware. Sequences must reference the truncated names.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.7 Sequence File Format

Sequence files (SEQ) are created and updated by the Sequence Editor. There is no requirement to edit or display these files directly. The following information is provided for completeness only.

The sequence file format is patterned after Windows' INI files. This is an ASCII text format that can be viewed or printed. Programmatic access to the contents of an INI file is quickly achieved through standard Windows' API functions (WritePrivateProfileString and GetPrivateProfileString).

The format of an INI file consists of sections and entries. A section is denoted by a bracketed name (e.g. [SequenceEditor]). Within a section, entries are labels that are assigned particular values. For example:

```
[SequenceEditor]
Version=2.0
TimingList=IDLE;READ;TS1;TS2;TS3;WRITE;
TableList=Table1;Table2;Table3;Table4;Table5;
```

For a sequence file there is one section devoted to general information and other sections represent sequences. The sequence sections are named in numerical order (e.g. S1, S2, S3…). Within each sequence section, there is an entry for the sequence name and a numeric list of the subsequences.

Keywords	Description
[SequenceEditor]	Section header for general sequence information.
Version =	File format version (2.0)
TimingList =	Semicolon delimited list of timing set names.
TableList =	Semicolon delimited list of table parameters.
[VectorList]	
V0 =	Destination for vector 0 branch
V1 =	Destination for vector 1 branch
:	Additional vector destinations
[S1]	Section header for first sequence.
Name =	Name of the sequence.
1 =	First subsequence definition.
2 =	Second subsequence definition.
:	Additional subsequence definitions
[S2]	Section header for second sequence.
Name =	Name of the sequence.
1 =	First subsequence definition.
2 =	Second subsequence definition.
:	Additional subsequence definitions

Only non-default destinations are listed in the vector list section.

> **Note: The Sequence Editor does not automatically validate a sequence file before saving it. Consequently, a sequence file may contain invalid data (e.g. an unknown destination).**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following is the example sequence file (SEQ_DEMO.SEQ) included with the Sequence Editor:

```
[SequenceEditor]
Version=2.0
TimingList=Go_Fast;Go_Slow;IDLE;Vect_Count;
TableList=Ramp;Rotate;Thumper;Zing;

[VectorList]
V0=Seq_Two[1]
V2=Seq_Two[3]
V8=Seq_One[5]

[S1]
Name=Seq_One
1=Go_Fast;Ramp;1;Next;;;
2=Go_Fast;Rotate;1;Next;;;
3=Go_Fast;Thumper;1;Next;;;
4=Go_Fast;Zing;1;Next;;;
5=Go_Fast;Ramp;1;Stop;;;

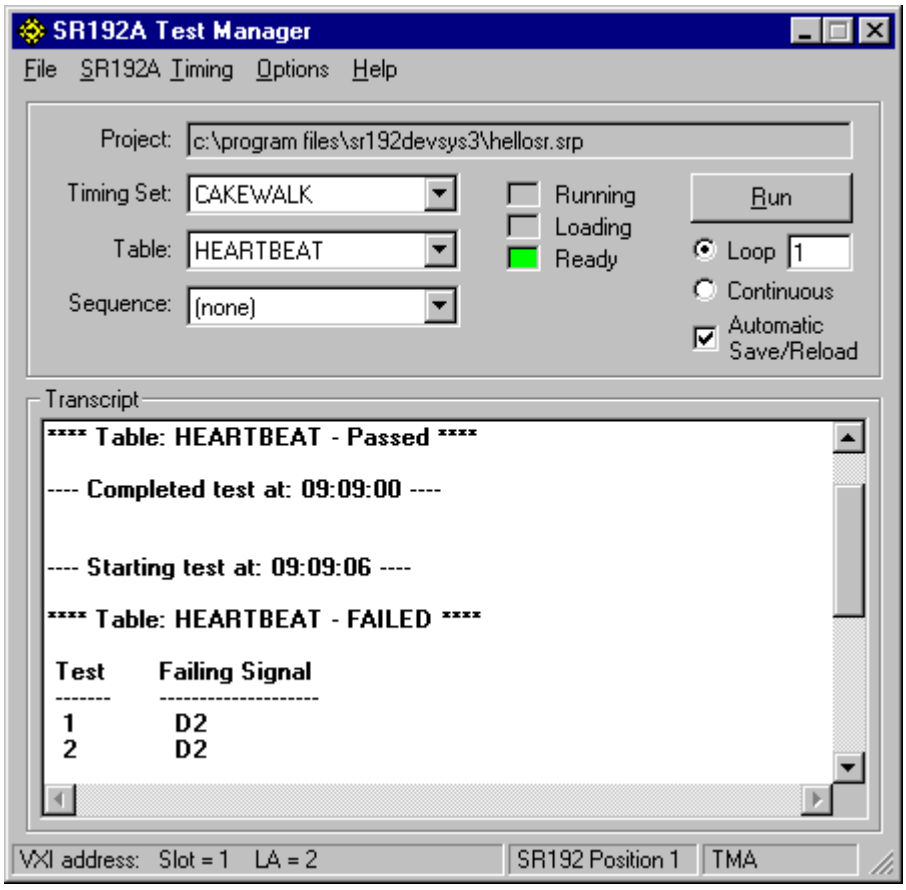
[S2]
Name=Seq_Two
1=Go_Slow;Ramp;4;Next;;;
2=Go_Fast;Rotate;1;Next;;;
3=Go_Fast;Thumper;1;Next;;;
4=Go_Fast;Zing;1;Next;;;
5=Vect_Count;Ramp;1;Stop;;;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8. Test Manager

The Test Manager is one of the software components of the SR192A Development System. The Test Manager handles loading, executing and test result reporting for project files created with the SR192A Development Environment. The Test Manager is used with the SR192A Development Environment during debug and it also runs standalone for production testing. Source code for the Test Manager is provided for customization.

The Test Manager has an easy-to-use interface for loading SR192A project and sequence files, executing timing set/table pairs and reporting test results. The Test Manager also supports reading SR192A test data and writing it to a project file. In addition, the Test Manager has looping and continuous run modes.



To use the Test Manager simply load a project file, select an appropriate timing set/table pair and press the Run button. Test results are reported in the Transcript window and operating status is indicated by LED-like controls.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.1 Operation

The Test Manager provides a bridge between the SR192A Development Environment and the SR192A hardware. It coordinates project loading, execution and test result reporting. The following describes the operation of the Test Manager interface.

The Project field shows the current path and name of the loaded SR192A project file. A project file is loaded, or reloaded, with the File Menu. The Timing Set and Table drop-down lists have the timing set and table names of the selected SR192A's specified timing generator. The current SR192A and timing generator are selected from the SR192A and Timing menus. The optional Sequence drop-down list has the sequence names from the selected SR192A.

Pressing the Run button causes the SR192A to run the selected timing set/table pair or sequence. While running, the button is labeled Stop. Pressing the Stop button immediately halts the SR192A. The Stop button is always used to halt the SR192A when it is running in Continuous mode. The Loop value defaults to one, but it can be set up to a count of 9999.

> **Note: A selected sequence has execution priority over a timing set/table pair. Set the sequence to "(none)" in order to execute an individual timing set/table pair.**

When a run is initiated, several checks are performed to ensure that the project and sequence files are current. The exact behavior of these checks is controlled by the Automatic Save/Reload checkbox. Following a run, the Test Manager distributes the execution results to any active SR192A Development Environments for use in their error displays.

The Transcript window is a continuous journal of test activity and test results. Its contents can be cut (**Ctrl-X**), copied (**Ctrl-C**), pasted (**Ctrl-V**) or cleared with the Options Menu. When the Transcript window buffer is full, a portion of the oldest data is deleted. During normal operation, the Transcript window logs test start, Pass/Fail status and displays detailed test failure information. When looping, only the last pass is tested for errors. In Continuous mode, or on early termination while looping, no Pass/Fail analysis is performed because the results would be inconclusive. The Transcript window also displays any error messages reported by the system.

There are three status LED's labeled Running, Loading and Ready. Initially, all three LED's are grayed out. Once an SR192A project file is loaded, the Ready LED turns green. While loading or reading a project file, the Loading LED turns yellow. The Running LED turns yellow while the SR192A is actively running.

The status bar at the bottom of the Test Manager window shows the currently selected SR192A's Slot, Logical Address, logical position, Timing Generator and operation mode.

The Test Manager is written in Visual Basic and its source code is provided (<Install_Dir>\Examples\TestMan*.*). Control of the SR192A is through the SR192A ActiveX Driver DLL. For production testing, the Test Manager may be used as supplied or customized to meet specific requirements. Alternately, the Test Manager can serve as a nontrivial example of interfacing to the SR192A ActiveX Driver DLL.

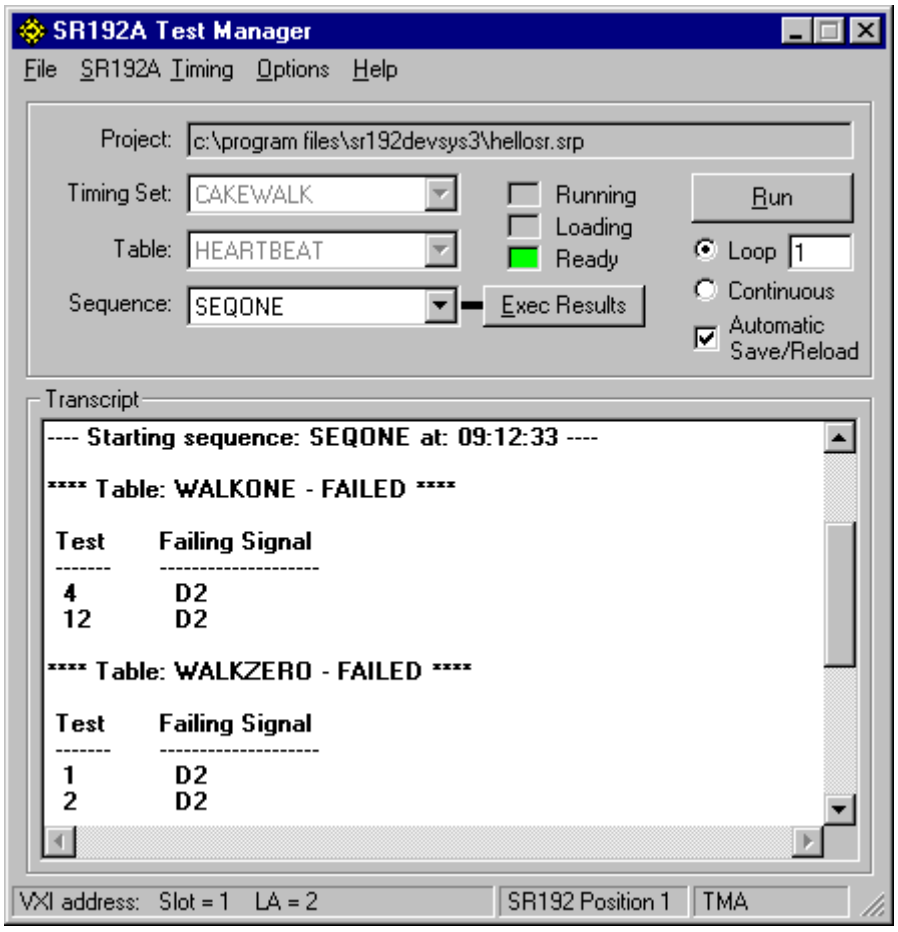
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.2 Sequence Execution

Sequences coordinate the execution of multiple timing set and table pairs. This allows a series of operations to be executed, such as sequential read/write cycles on a bus. With the capability of conditional branching and subroutines, sequences can generate complex series of patterns and behavior.

Once a sequence completes, its execution status (i.e. pass or fail) is determined by checking each of the executed tables for errors. Since sequences can be highly complex, the test developer must define the tables to query after a sequence executes.

When a sequence is selected, on the Test Manager window, the Execution Results ("Exec Results") button becomes visible. The appearance of this button is to emphasis that sequence errors are only reported for the tables selected on the Execution Results dialog.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.3 Automatic Save/Reload

The Test Manager monitors the status of project and sequence files in order to ensure that the latest changes have been downloaded to the hardware. This is particularly valuable during development and debug when a rapid edit-load-run cycle can be disrupted by forgetting to save or reload an important change. Equally important is to eliminate the suspicion that an update was missed (Did I…?).

The Automatic Save/Reload checkbox controls the save and reload process. If checked, the Test Manager automatically instigates any required save or reload of project or sequence files. If unchecked, the user is notified when unsaved or modified files are detected. The user then decides whether to save and/or reload the applicable files.

When a run is initiated, the Test Manager first queries active Sequence Editors and SR192A Development Environments to determine if editing changes have been made to the currently loaded sequence or project file. If project changes are detected, the applications are instructed to perform a file save.

As a second step, the Test Manager determines if any files have been modified since the last hardware download. This step is independent of the first because changes could have been manually saved by the operator prior to starting the run. Modified files are detected and downloaded to the SR192A.

> **Note: If a project file is reloaded, the sequence file is also reloaded in order to maintain synchronization with table and timing set locations.**

Under certain conditions, it may be best to temporarily disable (i.e. uncheck) the automatic save/reload mechanism. This might occur when a project is partially modified and you are not ready for it to be downloaded during the next run. Or, when the contents of the SR192A have been altered by another application and you want to test the modifications.

Even under normal circumstances it is still valuable to avoid extraneous editing of a project file while debugging with the Test Manager. Any editing of the project file can cause a save and reload to occur. This includes reordering signals, toggling a signal state, examining the contents of a drop-down list or selecting a signal name. To best avoid inadvertent changes to a project, you should use Cancel to close dialog boxes and discard changes when closing the Table Editor, Timing Set Editor and Signal List Editor.

Even when automatic save/reload is disabled, the user is still notified if a modification is detected and is given the opportunity to download the changes. To force a complete file reload, use the Options Menu to reset the SR192A.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

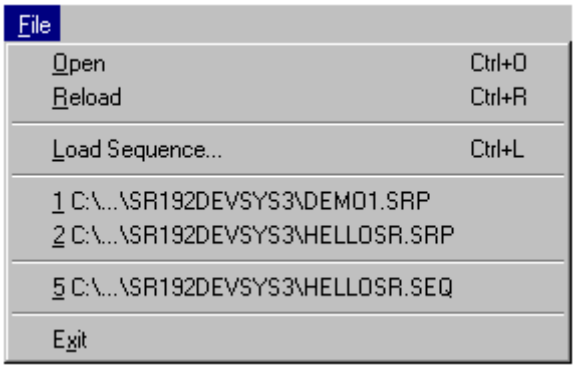
8.4 Menus

8.4.1 File Menu

The Test Manager File Menu supports selection and management of SR192A project files. Sequence files created with the Sequence Editor are downloaded via this menu also.

> **Note: Sequences are downloaded to the currently selected SR192A and timing generator.**

The manual project reload on this menu is often not necessary because the Test Manager has an automatic save/reload capability. The manual reload only updates the hardware if a file change is detected. To force a complete project reload, reset the SR192A with the Options Menu before reloading a project.



Menu Option	Description
Open	Opens a file browser to select a project file to download to the SR192A hardware.
Reload	Reloads current project file and updates the SR192A.
Load Sequence…	Select and download a file of sequence definitions to the SR192A hardware.
File History	Loads one of the last four project, or sequence, files. The file names are stored in a state file (SR192ATM.INI).
Exit	Closes the Test Manager. The contents of the SR192A are left intact.

> **Note: Timing sets and tables must be loaded into the SR192A before loading sequences that reference them. Errors are reported if a sequence references a nonexistent timing set or table.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.4.2 SR192A Menu

The Test Manager SR192A Menu contains a list of detected SR192A modules. If only one SR192A is installed, the menu shows only one entry. If multiple SR192A modules are present, the menu lists each module by logical position with a check next to the currently selected one. Only one module is selected at a time. When a selection is changed, the Timing Set, Table and Sequence drop-down lists are updated to reflect the new setting. The current setting is displayed on the status bar.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.4.3 Timing Menu

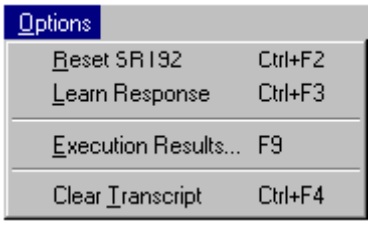
The Test Manager Timing Menu selects a timing generator module for operation. Selection is performed by clicking on the desired timing module. Once selected, the Timing Set, Table and Sequence drop-down lists are updated with the contents of the specified timing module. The current selection is also displayed on the status bar. Items on this menu are only enabled if they are detected in the SR192A hardware.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.4.4 Options Menu

The Test Manager Options Menu provides additional control over the user interface and SR192A hardware. It resets the selected SR192A, accesses the Execution Results dialog and clears the transcript window. If a complete project file reload is desired, use this to reset the SR192A before reloading the project.



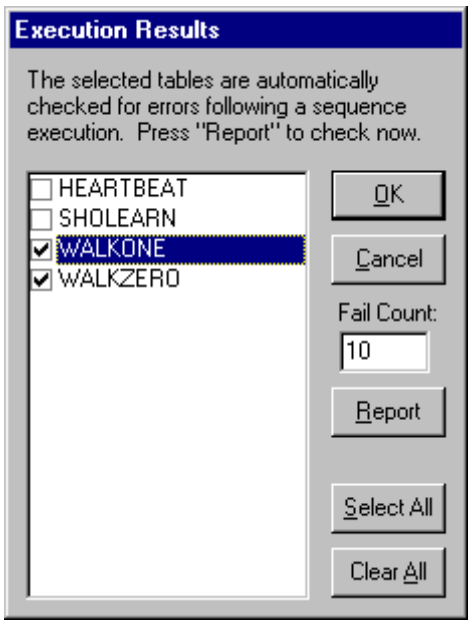
Menu Option	Description
Reset SR192A	Causes a hard reset of the SR192A. This initializes it to a power-on state. All timing sets, tables, module groups and sequences are deleted.
Execution Results…	Activates the Execution Results dialog to select tables for error reporting.
Clear Transcript	Clears the contents of the Transcript window.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.5 Execution Results

The Test Manager's Execution Results dialog is used to select data tables for error reporting. The selected tables are queried following a sequence execution, or immediately if the Report button is pressed. Since a sequence can cause one or more timing set/table pairs to be executed, it is necessary to select the tables that are to be queried for their test status.

Fail Count defines the number of failing vectors to detect and report. This greatly influences the Error Display, by the Table Editor, and the contents of the execution result file. The goal is to report a sufficient number of errors, for debugging and diagnostic purposes, without expending the time and resources required to report all errors.



Use the OK button to accept the changes and close the dialog. The Cancel button discards the changes and closes the dialog. The Select All and Clear All buttons quickly check, or uncheck, all the tables in the list. Use the Report button to immediately report the status of the selected tables. This allows a report to be created without an additional execution.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.6 Execution Result File Format

The Test Manager creates an execution result file (FAULT.RES) whenever a test is run. For a sequence, this is built based on the tables selected by the Execution Results dialog. This file is used by the SR192A Development Environment to set corresponding Error Displays. There is no need to edit or display these files directly. The following information is provided for completeness only.

The execution result file format is patterned after Windows' INI files. This is an ASCII text format that can be viewed or printed. Programmatic access to the contents of an INI file is quickly achieved through standard Windows' API functions (e.g. GetPrivateProfileString).

The format of an INI file consists of sections and entries. A section is denoted by a bracketed name (e.g. [FaultFile]). Within a section, entries are labels that are assigned particular values. For example:

```
[FaultFile]
Project = c:\program files\sr192devsys3\demo1.srp
```

For an execution result file there is one section that holds the test results for multiple tables. Each table is identified by name prior to the listing of test results. If a table entry is not followed by vector/signal failures, that means the table has no errors to report.

Keywords		Description
[FaultFile]		Section header for execution result information.
	Project =	SR192A project file name and path.
	FailCount =	Maximum number of vector failures being reported.
	TableX =	SR Module and table X name (matches window title of Table Editor).
	V1 S1	First failing vector and signal for table X.
	V2 S2	Second failing vector and signal for table X.
	V3 S3	Third failing vector and signal for table X.
	:	Additional failing vectors and signals for table X.
	TableX+1 =	SR Module and table X+1 name.
	V1 S1	First failing vector and signal for table X+1.
	V2 S2	Second failing vector and signal for table X+1.
	V3 S3	Third failing vector and signal for table X+1.
	:	Additional failing vectors and signals for table X+1.

The Test Manager creates this file by intermixing direct file writes with calls to ssSrGetExecResults. The direct file operations supply the necessary INI formatting and project information. The ssSrGetExecResults function appends the vector/signal failure results. The code for creating this file is included with the supplied Test Manager source (<Install_Dir>\Examples\TestMan*.*).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following is an example execution result file. Notice that Table1 reports two signal failures for the same vector, and Table2 has no failures to report.

```
[FaultFile]
Project = c:\program files\sr192devsys3\demo1.srp
FailCount = 10
Table1 = \SR Module: 1\Tables\HEARTBEAT
 3 D4
 3 D2
 10 D6
 17 D9
Table2 = \SR Module: 1\Tables\WALKONE
Table3 = \SR Module: 1\Tables\WALKZERO
 4 D2
 6 D4
 12 D2
 14 D4
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9. Programming Interface

Most production test scenarios involving the SR192A can be handled by the Test Manager. In some cases, small modifications to the Test Manager are all that is required. For that reason the source code for the Test Manager is supplied (<Install_Dir>\Examples\TestMan*.*). Some testing situations require a custom application or close integration with existing systems and procedures. To meet these requirements, a programming interface is included as part of the SR192A Development System.

The SR192A ActiveX Driver DLL (ssSr192AX.DLL) provides a high-level software interface and execution resource for the SR192A hardware. It contains a set of high-level function calls designed specifically to work with SR192A project files created with the SR192A Development Environment. These function calls internally manage SR192A low level operations such as selection, error checking and test result reporting, thereby freeing the application programmer from this burden. Since the ssSr192AX.DLL is ActiveX compliant, it may be used by any application development environment that supports ActiveX calling protocol. This includes the latest commercial versions of HP-VEE, National Instruments LabWindows/CVI, National Instruments LabView, Visual C++ and Visual Basic.

The SR192A ActiveX Driver application program interface (API) contains a set of function calls that operate on the currently selected SR192A module. The default module is one (1), the leftmost SR192A module in the VXI chassis. Other modules are selected with the ssSrSelectSR192 function. Returned error values are negative. They can be passed to ssSrGetErrorMessage for translation into a descriptive string.

ActiveX components are object oriented and thereby require object instantiation and reference for the API calls. The name of the SR192A ActiveX Driver object is **SR192A**. This name must be used when creating an instance of the object for accessing the API. Please refer to your application development environment's documentation for the proper instantiation and calling syntax. General guidelines for various popular environments are covered in a later section of this document, and examples are provided in a subdirectory to the SR192A Development System (<Install_Dir>\Examples). Additionally, Visual Basic and Visual C++ examples are included in this document for reference purposes.

The SR192A ActiveX Driver has comprehensive error checking and reporting. A description of the error code and the context that it occurred in is retrieved with the ssSrGetErrorMessage function. In addition, all errors are written to a log file (ssSr192AX.LOG**)**. A description of the log file and its entries are outlined in a subsequent section.

The following is an alphabetical list of the SR192A driver functions with a short description of each.

ssSrExecute (*timGen*, *timingSetName*, *tableName*, *loopCount*)
Starts execution of the specified timing set and table. Return immediately without waiting for execution to complete.

ssSrExecuteSequence (*timGen*, *sequenceName*, *loopCount*)
Starts execution of the specified sequence. Return immediately without waiting for execution to complete.

ssSrGetConfig (*iSlot*, *iLa*, *tgCount*, *tgLinked*)
Retrieves from the selected SR192A its slot location, logical address, number of installed timing generators and whether timing generators A and B are linked.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ssSrGetErrorMessage (*errNum*, *buffer*, *bufSize*)
Fills the specified buffer with a description of the error represented by the specified error value. The description is truncated to *bufSize* value.

ssSrGetExecResults (*timGen*, *tableName*, *failCount*, *faultFile*, *appendFlag*)
Returns the results of the specified table's execution (1 = Pass, 0 = Fail). If the *failCount* is greater than zero, writes individual signal/vector failures to the specified fault file.

ssSrGetList (*listType*, *timGen*, *listBuffer*, *bufSize*)
Fills the specified buffer with a comma-delimited list of items from the selected SR192A's timing generator page. The list is truncated to *bufSize*.

ssSrHalt ( )
Halts the execution of the current SR192A. Preserves the current state and contents of the SR192A.

ssSrLoadProject (*projectFile*)
Loads the SR192A hardware with the contents of the specified project file.

ssSrLoadSequence (*timGen, sequencetFile*)
Loads the SR192A hardware with the contents of the specified sequence file.

ssSrReset ( )
Resets the current SR192A to a power-up state.

ssSrSelectSR192 (*modulePosition*)
Selects an SR192A.

ssSrStatus ( )
Retrieves the status of the current SR192A.

The SR192A ActiveX Driver DLL utilizes both VXI Plug and Play (VPP) VISA and the Talon SR192A VPP-4 compliant A32 driver in its implementation. This makes it compatible with any other VPP-compliant application for the Win32 framework.

For operating efficiency, the ActiveX Driver makes significant use of low-level register based commands when communicating with the SR192A. In particular, all the table memory accesses are register based A32 read/writes to optimize load/unload times. The ActiveX Driver also employs cache technology to retain SR192A state information in order to reduce VXI bus traffic.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.1 Execution Result Reporting

Reporting execution results, by the SR192A ActiveX Driver, is handled via the ssSrGetExecResults function. Specifically, ssSrGetExecResults retrieves execution status for a named table. This allows it to be used following sequence executions (ssSrExecuteSequence) and direct *Plug&Play* driver operations. If multiple tables are executed during a sequence, ssSrGetExecResults should be called for each table.

Caveats and Concerns

Here are some things to check for if strange (goofy) errors are reported in the fault results file:

1. *Power Supply and I/O Thresholds:* Make sure that programmable thresholds are set properly and enough delay is programmed in the test application to allow power supplies to cycle and stabilize.

2. *Wait for SR192A to Complete Operation:* Make sure that the SR192A execution is completed before invoking the ssSrGetExecResults function. A good way of doing this is to use the ssSrStatus function in a loop and wait until both Timing Generators A and B have stopped running.

3. *SR192A Hardware Error Count:* The SR192A hardware error counter is global to all tables while the ssSrGetExecResults function only checks the specified table for errors. It is possible to have the hardware error counter set (global indicator) and the ssSrGetExecResults report no errors (local table indicator).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.2 Driver Updates

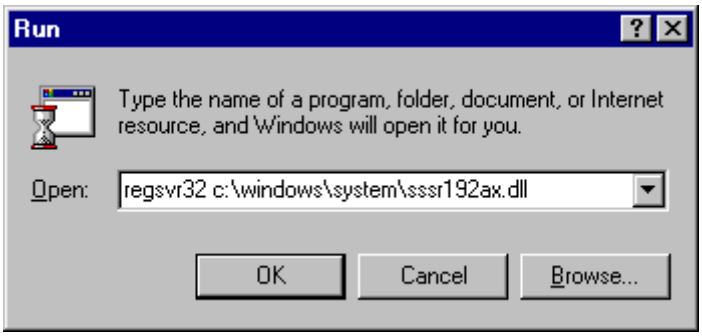
This section describes recent SR192A ActiveX driver updates and the changes that have been implemented in each. It is a good practice to keep your drivers updated in order to avoid diagnosing a problem that has already been solved.

Because the SR192A driver is an ActiveX component, each new version must be registered with the operating system. The following are the steps to take when you receive a new SR192A ActiveX driver.

1. Copy the driver to the Windows System directory. Depending on the specific operating system, this directory may be named differently (e.g. "c:\windows\system\" or "c:\winnt\system32\").

2. Register the driver by executing the following from the Run command on the Start Menu. The directory path must match where the driver was copied in step 1.

regsvr32 c:\windows\system\sssr192ax.DLL



Software Differences Between V1.00.0008 and V1.00.0005

1. Various improvements have been made to the SR192A ActiveX driver to support Master/Slave operation and to keep pace with SR192A firmware and Plug&Play changes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.3 Error Log File

The SR192A ActiveX Driver logs all errors to a running journal file named **ssSr192XA.LOG**. Entries to the log file are appended and roll over when the file reaches 10k in size. The entries are ASCII text and can be viewed using the Notepad or any text editor of choice. Since only errors are logged, the file is inactive during normal operation, thereby minimizing any operating overhead.

In addition to error entries, a session start and stop time stamp is logged every time the ssSr192AX.DLL is loaded and unloaded. This brackets any reported errors to a specific session and its relative time and date.

The messages written to the error log file conform to the following convention:

Error: <SSeCode>**:**<VPPeCode> <Error source> **reported** <"VISA/VPP Driver Error Message"> **while** <operational context description when error occurred>.

An example error message is shown below:

Error: BFFC0FFB:BFFF0015 SR192A reported "Timeout expired before operation completed" while reading Timing Set directory.

The VPPeCodes are errors reported by VISA and the VXI Plug&Play (VPP) Driver. A list of these codes and their descriptions are found in the VPP VISA and VPP driver documentation. The SSeCodes are generated by the SR192A ActiveX Driver and their descriptions follow:

Constant Label	Error Code	Description
SS_ERROR_PJOPEN	(BFFC0FFF)	Error opening/closing project file.
SS_ERROR_PJACCESS	(BFFC0FFE)	Error reading/writing project file.
SS_ERROR_OSFAIL	(BFFC0FFD)	Operating system error.
SS_ERROR_RMFAIL	(BFFC0FFC)	VISA Resource Manager error.
SS_ERROR_SRFAIL	(BFFC0FFB)	SR192A VPP/VISA error.
SS_ERROR_ESCAPE_TIMEOUT	(BFFC0FFA)	Escape timeout expired.
SS_ERROR_SELECT_FAIL	(BFFC0FF9)	Unable to locate SR192A in VXI chassis.
SS_ERROR_SRSYS	(BFFC0FF8)	SR192A SCPI error.
SS_ERROR_TABLEERR	(BFFC0FF7)	SR192A table assignment error.
SS_ERROR_PROGRAM	(BFFC0FF6)	SR192A application error.
SS_ERROR_VERSION	(BFFC0FF5)	Driver/Firmware version warning.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4 Functions

The following sections describe each of the functions available from the SR192A ActiveX Driver DLL. The descriptions include calling syntax, parameter lists, return values and operation.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.1 ssSrExecute

VB	**Function**	*object*.**ssSrExecute (** *timGen* As String**,** *timSetName* As String**,** *tableName* As String**,** ByVal *loopCount* As Long **) As Long**
C++	**long**	*object*->**ssSrExecute (** BSTR **timGen*, BSTR **timSetName*, BSTR **tableName*, long *loopCount* **);**

Parameter	I/O	Description
timGen	in	Timing generator select ("A" or "B").
timSetName	in	Name of timing set to run.
tableName	in	Name of table to run.
loopCount	in	Run loop count (1 to 65535). A negative value causes it to loop continuously.

Return Value

0 = Successful operation. Negative values are error codes. To get a description of the error, pass the error code to ssSrGetErrorMessage.

Remarks

Starts execution of the specified timing set and table pair on the currently selected SR192A. The loop count specifies the number of times for the cycle to repeat. Returns immediately without waiting for the SR192A to complete. Use the ssSrStatus function to monitor test completion. Once the test is complete, use the ssSrGetExecResults function to retrieve the table's execution status.

If the SR192A is running, this function waits for up to ten seconds for it to complete the previous operation before starting execution.

> **Note: Table and timing set names are truncated to ten characters when they are downloaded to the SR192A hardware. This function must reference the truncated names.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.2 ssSrExecuteSequence

VB	**Function**	*object*.**ssSrExecuteSequence (** *timGen* As String**,** *seqName* As String**,** ByVal *loopCount* As Long **) As Long**
C++	**long**	*object*->**ssSrExecuteSequence(** BSTR **timGen***,* BSTR **seqName***,* long *loopCount* **);**

Parameter	I/O	Description
timGen	in	Timing generator select ("A" or "B").
seqName	in	Name of sequence to run.
loopCount	in	Run loop count (1 to 65535). A negative value causes it to loop continuously.

Return Value

0 = Successful operation. Negative values are error codes. To get a description of the error, pass the error code to ssSrGetErrorMessage.

Remarks

Starts execution of the specified sequence on the currently selected SR192A. The loop count specifies the number of times for the cycle to repeat. Returns immediately without waiting for the SR192A to complete. Use the ssSrStatus function to monitor test completion. Once the test is complete, use the ssSrGetExecResults function to retrieve execution status for the tables associated with the sequence. Sequences are loaded with the ssSrLoadSequence function.

If the SR192A is running, this function waits for up to ten seconds for it to complete the previous operation before starting execution.

> **Note: Sequence names are truncated to ten characters when they are downloaded to the SR192A hardware. This function must reference the truncated name.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.3 ssSrGetConfig

VB	**Function**	*object*.**ssSrGetConfig (** *iSlot* As Long**,** *iLa* As Long**,** *tgCount* As Long**,** *tgLinked* As Long **) As Long**
C++	**long**	*object*->**ssSrGetConfig (** long **iSlot*, long **iLa*, long **tgCount*, long **tgLinked* **);**

Parameter	I/O	Description
iSlot	out	Reports the physical slot location in the VXI chassis.
iLa	out	Reports the VISA logical address.
tgCount	out	Reports the number of timing generators installed in the SR192A (0, 1 or 2).
tgLinked	out	Reports if timing generators A and B are linked. (0 = Not Linked, 1 = Linked)

Return Value

Returns the number of SR192A instruments detected in the VXI chassis. Negative values are error codes. To get a description of the error, pass the error code to ssSrGetErrorMessage.

Remarks

Retrieves from the currently selected SR192A its slot location, logical address, number of installed timing generators and whether timing generators A and B are linked.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.4 ssSrGetErrorMessage

VB	**Function**	*object*.**ssSrGetErrorMessage (** ByVal *errNum* As Long**,** *buf* As String**,** ByVal *bufSize* As Long **) As Long**
C++	**long**	*object*->**ssSrGetErrorMessage (** long *errNum***,** BSTR **buf***,* long *bufSize* **);**

Parameter	I/O	Description
errNum	in	Error number to evaluate.
buf	out	Output buffer for the error message return.
bufSize	in	The size of the output buffer.

Return Value

0 = Successful operation. Negative values are error codes. To get a description of the error, pass the error code to this function.

Remarks

Fills the specified string buffer with a description of the error represented by *errNum*. The description is truncated to *bufSize*. The SR192A ActiveX driver also maintains a dynamic error description stack. As errors are encountered, their descriptions and context (e.g. "loading sequence") are placed on the stack. These descriptions provide more information about a problem than a simple lookup table of canned error descriptions. To retrieve the top entry of the error stack, use an *errNum* of zero (0). For best results, retrieve all errors from the stack in a code loop. The error stack is empty when a negative one (-1) is returned.

When calling this function from Visual Basic it is recommended to use a fixed length string for *buf.* Otherwise, fill a string to the appropriate size with String$ or Space$.

Error messages are classified into three categories. Errors detected in the SR192A ActiveX Driver (ssSr192AX.DLL); errors attributed to the VXI Plug&Play (VPP) driver (tasr192_32.DLL); and errors attributed to the VXI Resource Manager and VISA.

The format of the returned error message conforms to the following convention:

Error: <SSeCode>**:**<VPPeCode> <Error source> **reported** <"VISA/VPP Driver Error Message"> **while** <operational context description when error occurred>.

An example error message is shown below:

Error: BFFC0FFB:BFFF0015 SR192A reported "Timeout expired before operation completed" while reading Timing Set directory.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The VPPeCodes are errors reported by VISA and the VPP Driver. A list of these codes and their descriptions are found in the VPP VISA and VPP driver documentation. The SSeCodes are generated by the SR192A ActiveX Driver and their descriptions follow:

Constant Label	Error Code	Description
SS_ERROR_PJOPEN	(BFFC0FFF)	Error opening/closing project file.
SS_ERROR_PJACCESS	(BFFC0FFE)	Error reading/writing project file.
SS_ERROR_OSFAIL	(BFFC0FFD)	Operating system error.
SS_ERROR_RMFAIL	(BFFC0FFC)	VISA Resource Manager error.
SS_ERROR_SRFAIL	(BFFC0FFB)	SR192A VPP/VISA error.
SS_ERROR_ESCAPE_TIMEOUT	(BFFC0FFA)	Escape timeout expired.
SS_ERROR_SELECT_FAIL	(BFFC0FF9)	Unable to locate SR192A in VXI chassis.
SS_ERROR_SRSYS	(BFFC0FF8)	SR192A SCPI error.
SS_ERROR_TABLEERR	(BFFC0FF7)	SR192A table assignment error.
SS_ERROR_PROGRAM	(BFFC0FF6)	SR192A application error.
SS_ERROR_VERSION	(BFFC0FF5)	Driver/Firmware version warning.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.5 ssSrGetExecResults

VB	**Function**	*object*.**ssSrGetExecResults** (*timGen* As String, *tableName* As String, ByVal *failCount* As Long, *faultFile* As String, ByVal *flag* As Long) **As Long**
C++	**long**	*object*-> **ssSrGetExecResults(** BSTR **timGen***,* BSTR **tabelName*, long *failCount***,** BSTR **faultFile***,* long *flag***);**

Parameter	I/O	Description
timGen	in	Timing generator select ("A" or "B").
tableName	in	Name of table to scan for test results.
failCount	in	Number of vector failures to report. If *failCount* is zero, do not report any errors in the *faultFile*.
faultFile	in	The file name and path for writing vector/signal failure information.
flag	in	*faultFile* overwrite or append flag. If the flag value is zero (0), this function overwrites any existing data in the *faultfile*. If the flag value is one (1), this function appends the test results to the existing data in the *faultFile*.

Return Value

1 = Test passed, 0 = Test failed. Negative values are error codes. To get a description of the error, pass the error code to ssSrGetErrorMessage.

Remarks

Returns a value for the results of the last test execution for a given timing generator and named table pair. A value of one (1) means the test passed, a value of zero (0) means the test failed. If *failCount* is greater than zero, the individual vector/signal failures are written to the specified *faultFile*. If *failCount* is zero, the failure information is not written; however, pass/fail information is still reported through the return value. Specifying a *failCount* of zero provides a quick pass/fail evaluation.

The fault file is an ASCII file containing detailed failure information about an execution. Previous copies of the fault file are overwritten or appended to depending on the value of the *flag* parameter. If there are no errors to report, an empty fault file is generated. The format of the fault file consists of a failed test vector and signal on each line. The first field is the number of the failed test vector; and the second field is the signal name associated with the failing SR192A channel. Test vectors are unique for each table and are numbered sequentially from one to the table size.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In the following example, failures are detected on SIG1 at test vectors 1, 8, 9, 11 and 25. Failures are detected on SIG2 at test vectors 1, 6 and 11; and on SIG3 at test vector 1, 3 and 11.

```
1  SIG1
1  SIG2
1  SIG3
3  SIG3
6  SIG2
8  SIG1
9  SIG1
11  SIG1
11  SIG2
11  SIG3
25  SIG1
```

The *failCount* parameter specifies the number of failing vectors to report. If a *failCount* of 3 was specified for the example above, only test vectors 1, 3 and 6 would be reported in the fault file. Note that *failCount* controls the reporting of detected vector errors. This could turn out to be vectors 1, 3 and 6; as in the example above, or vectors 12, 57 and 99.

The *flag* variable is used to control the generation of the fault file data. If *flag* value is zero (0) any existing data in the specified fault file is overwritten with the test results reported by this function. If *flag* value is one (1) the test results are appended to the existing data in the fault file. Append provides a way to combine the test results from multiple tables into a single fault file. This is particularly useful for reporting sequence execution results since sequences typically run multiple tables.

The Test Manager builds an execution result file by intermixing calls to this function and direct file operations. An execution result file is a multi-table fault file augmented by project and table information. The code for doing this is included with the supplied Test Manager source (<Install_Dir>\Examples\TestMan*.*).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.6 ssSrGetList

VB	**Function**	*object*.**ssSrGetList (** *listType* As Long, *timGen* As String**,** *listBuf* As String**,** ByVal *bufSize* As Long **) As Long**
C++	**long**	*object*->**ssSrGetList(** long *listType*, BSTR **timGen***,* BSTR **listBuf***,* long *bufSize* **);**

Parameter	I/O	Description
listType	in	Type of list to return: timing set names (0), table names (1), sequence names (2) or control mode (3).
timGen	in	Timing generator select ("A" or "B").
listBuf	out	Output buffer for comma-delimited list of specified items.
bufSize	in	Size of the output buffer.

Return Value

0 = Successful operation. Negative values are error codes. To get a description of the error, pass the error code to ssSrGetErrorMessage.

Remarks

Fills the specified output buffer with a comma-delimited list of names retrieved from the selected SR192A. Truncates the output buffer at *bufSize*. The control mode query returns how the SR192A is configured to operate ("Independent", "Master", or "Slave").

When calling this function from Visual Basic it is recommended to use a fixed length string for *listBuf.* Otherwise, fill a string to the appropriate size with String$ or Space$.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.7 ssSrHalt

VB	**Function**	*object*.**ssSrHalt ( ) As Long**
C++	**long**	*object*->**ssSrHalt (** *void* **);**

Parameter	I/O	Description
(none)		

Return Value

0 = Successful operation. Negative values are error codes. To get a description of the error, pass the error code to ssSrGetErrorMessage.

Remarks

Halts execution of the currently selected SR192A. The SR192A's contents and settings are preserved. This function can also be used to halt ssSrLoadProject.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.8 ssSrLoadProject

VB	**Function**	*object*.**ssSrLoadProject (** *projectFile* As String **) as Long**
C++	**long**	*object*->**ssSrLoadProject (** BSTR **projectFile* **);**

Parameter	I/O	Description
projectFile	in	SR192A project file name and path.

Return Value

0 = Successful operation. Negative values are error codes. To get a description of the error, pass the error code to ssSrGetErrorMessage.

Remarks

This function loads the contents of the specified SR192A project file into the currently selected SR192A. The project file defines which SR192A instruments are affected. Project files are created and viewed with the SR192A Development Environment.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.9 ssSrLoadSequence

VB	**Function**	*object*.**ssSrLoadSequence (** *timGen* as String**,** *seqFile* As String **) as Long**
C++	**long**	*object*->**ssSrLoadSequence (** BSTR **timGen*, BSTR **seqFile* **);**

Parameter	I/O	Description
timGen	in	Timing generator select ("A" or "B").
seqFile	in	SR192A sequence file name and path.

Return Value

0 = Successful operation. Negative values are error codes. To get a description of the error, pass the error code to ssSrGetErrorMessage.

Remarks

This function loads the contents of the specified SR192A sequence file into the currently selected SR192A. A sequence file defines executable sequences of timing set and table pairs. Sequences are executed with the ssSrExecuteSequence function. Sequence files are created by the Sequence Editor.

> **Note: Timing sets and tables must be loaded into the SR192A before loading sequences that reference them. Errors are reported if a sequence references a nonexistent timing set or table.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.10 ssSrReset

VB	**Function**	*object*.**ssSrReset ( ) As Long**
C++	**long**	*object*->**ssSrReset (** *void* **);**

Parameter	**I/O**	**Description**
(none)		

Return Value

0 = Successful operation. Negative values are error codes. To get a description of the error, pass the error code to ssSrGetErrorMessage.

Remarks

Stops the execution of the currently selected SR192A and resets it to a power-up state. All tables, timing sets and sequences are deleted.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.11 ssSrSelectSR192

VB	**Function**	*object*.**ssSrSelectSR192 (** ByVal *modPosition* As Long **) As Long**
C++	**long**	*object*->**ssSrSelectSR192 (** long *modPosition* **);**

Parameter	I/O	Description
modPosition	in	Relative position to other SR192A modules in VXI chassis.

Return Value

0 = Successful operation. Negative values are error codes. To get a description of the error, pass the error code to ssSrGetErrorMessage.

Remarks

This function selects an SR192A for operation by its relative position in the VXI chassis. Position one (1) is the SR192A instrument closest to slot 0. If the specified module position is invalid, the current selection is unchanged. The default module position is one (1).

The selected SR192A is the instrument that the other functions operate with.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.4.12 ssSrStatus

VB	**Function**	*object*.**ssSrStatus ( )** As Long
C++	**long**	*object*->**ssSrStatus (** *void* **);**

Parameter	I/O	Description

(none)

Return Value

0 = The selected SR192A is ready.
1 = The selected SR192A's Timing Generator A is running.
2 = The selected SR192A's Timing Generator B is running.
3 = Both Timing Generator A and B are running.

Remarks

Checks the execution status of the currently selected SR192A. Use this to watch for the completion of a test following an ssSrExecute command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.5 Application Development Environments

There are many application development environments (ADE) that support ActiveX Automation and are able to interface to the SR192A ActiveX Driver DLL (ssSr192AX.DLL). Unfortunately each one does so in a slightly different way. The following sections provide general information about interfacing to Visual Basic, Visual C++, HP-VEE, National Instruments LabWindows/CVI and National Instruments LabView. For additional information, refer to the documentation supplied with the specific application development environment. Also, check the examples directory (<Install_Dir>\Examples\) that is installed with the SR192A Development System.

Prerequisites

Before exercising the programming interface, make sure that all of the components of the SR192A Development System, VISA and the Talon SR192A Plug&Play driver (32-bit) are installed and operating properly. A quick check of this is possible by loading and running the example project (HelloSR.SRP) with the Test Manager.

> **Note: Review the section on driver updates to ensure that the latest driver is properly installed and registered.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.5.1 Visual Basic

When setting up a Visual Basic application, the ssSr192AX DLL must be included as a reference. This is accomplished with the Reference option on the Project menu. Then declare an instance of the **SR192A** object and simply access its methods:

```
Dim drv As New SR192A          '  Make an instance of SR192A Driver Object

  ' Load a project file and execute a timing set (Go_Slow) and table (Rotate)
drv.ssSrLoadProject("c:\Program Files\sr192aDevSys4\HelloSRa.srp")
drv.ssSrExecute("A", "Go_Slow", "Rotate", 1)
```

A complete Visual Basic example is in the Examples section and the source is included with the SR192A Development System installation.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.5.2 Visual C++

The easiest way to make the **SR192A** calls accessible to a Visual C++ application (V6) is by using the #import directive with the type library (ssSr192AX.tlb). An instance of the **SR192A** object is then created and function calls are made via a pointer to the instantiated object. Sample C++ source code is in the Examples section.

9.5.3 HP-VEE

To access the ssSr192AX methods from an HP-VEE application, perform the following steps.

1. From the main menu select **Device>>ActiveX Automation References**. HP-VEE must be in the standard compatibility mode.

2. Select **References** and select the **ssSr192AX.TLB** file.

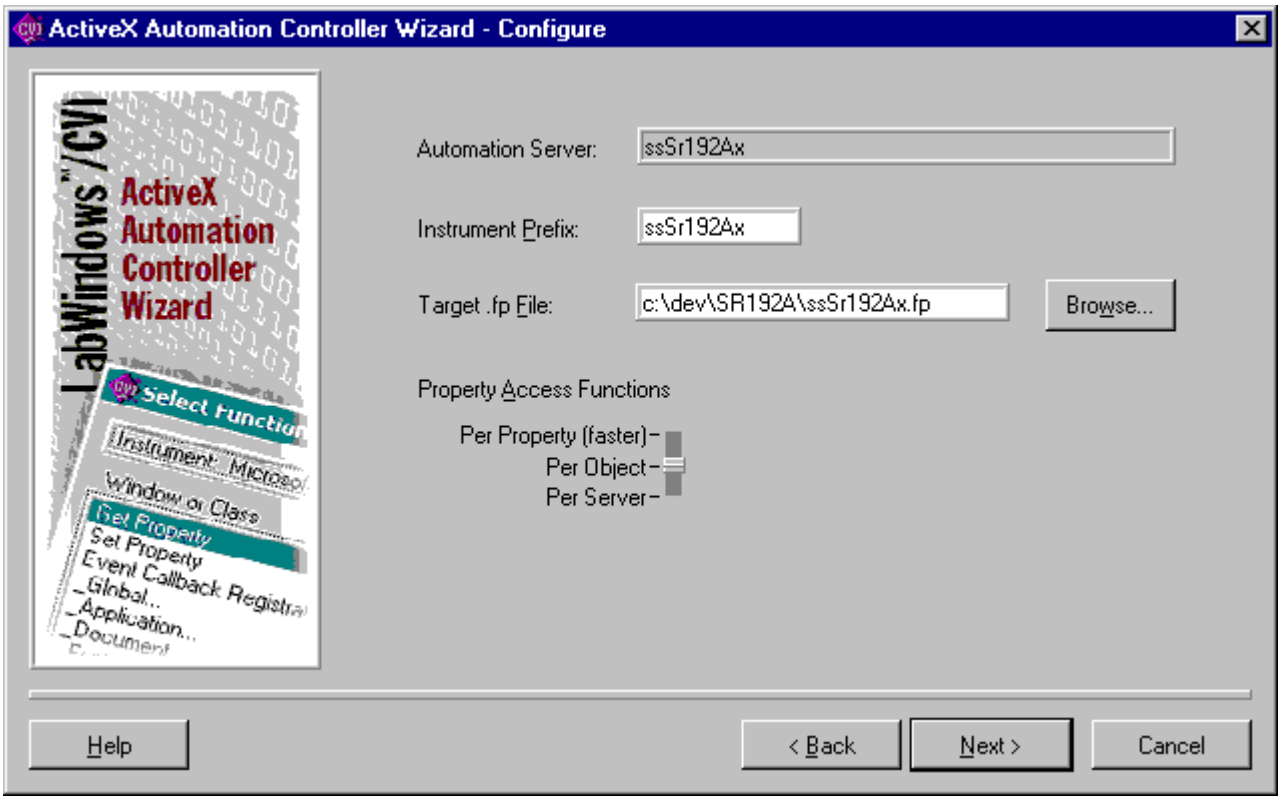
3. To use an ssSr192AX method, perform a **Select>>Function and Object Browser** from the main menu. Pick, place and connect the desired method in the application program.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.5.4 National Instruments LabWindows/CVI

The LabWindows/CVI software (V5) has a built-in tool to convert the supplied type library file (ssSr192AX.TLB) to a Function Panel (FP) file. The conversion tool is accessed from the Tools menu, **Tools>>Create ActiveX Automation Controller**, of LabWindows/CVI. Browse to locate the ssSr192AX.TLB file and select Generate to product a function panel file. To use the function panel in an application, select **Instrument>>Load>>ssSr192AX.FP** from the main menu.

LabWindows/CVI version 5.5 has additional settings in the **ActiveX Automation Controller Wizard**. For version 5.5, set the "Property Access Functions" selection to the "Per Object" position on the "ActiveX Automation Controller Wizard - Configure" form (see below).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.5.5 National Instruments LabView

To access the ssSr192AX methods from a LabView application (V5), perform the following steps.

1. Open the LabView diagram window. Make sure that the default palette set is selected. This selection is located in the **Edit>>Select Palette Set>>Default** menu.

2. Open the Functions Palette, if not already open. This selection may be found in the **Window>>Show Functions Palette** menu.

3. From the Functions Palette select **Communication>>ActiveX**. The **ActiveX** palette has five icons. Select the **Automation Open** icon and place it in the diagram window.

4. Right mouse click on the **Automation Open** icon in the diagram window. A pop-up menu will appear. Select the **ActiveX Class>>Browse** entry. Browse, locate and select the type library file (ssSr192AX.TLB). This file can be found in the **"c:\Program Files\SR192aDevSys4"** directory. Once selected, an **"SR192A(ssSr192Ax.SR192A)"** entry appears in the Objects text pane. Select this entry and click OK. The ActiveX Open icon in the diagram window now has an open handle assignment attached.

5. Next, select an **Invoke Node** icon from the **ActiveX** palette and place it in the diagram window. Open up a pop-up menu by right clicking on the newly placed **Invoke Node** icon. Select the **ActiveX Class>>ssSr192Ax._SR192A** entry in the pop-up menu. The Automation caption in the **Invoke Node** icon now changes to read **_SR192A**.

6. Make sure that the Finger tool is selected, then left click on the Method field on the **_SR192A Invoke Node** icon. This opens a list of available methods for the ssSr192Ax ActiveX Automation DLL. Choose a method from this list to transform the **Invoke Node** icon to the selected method with argument fields.

Once the ActiveX Automation icons are placed and specialized, they must be "wired" into the LabView application. This wiring and sequencing of Virtual Instruments (VI) and functions is the standard application development paradigm for LabView.

Keep in mind that all arguments and return values must be appropriately wired to input/output values. Some arguments are passed by reference. These items will have both input and output terminals associated with the entry. Both terminals must be connected for the VI to execute.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.6 Examples

9.6.1 Visual Basic Example

The following program example shows the minimum code required to load a project file, execute a timing set/table pair and report any test failures. The key program calls are bolded to make them stand out from the comments and error check code. For a more comprehensive application example for a Windows environment, look at the Test Manager source code.

```
'*********************************************************************************************
'  modSimpl.bas   v2.0
'    This code module contains a very simple example of interfacing with the
'    ssSr192AX driver DLL.  It is intended as a starting point for exploring the
'    driver's functionality.  Consequently, it has no user interface elements or
'    error resolving mechanisms.
'
'    Make sure that the SR192A object is checked in the References window.
'    (The References window selection is on the Project Menu)
'
'  Note VB conventions:
'    Comments begin with a single quote;
'    Line continuation is space/underscore/return.
'
'*********************************************************************************************

Option Explicit
Option Compare Text
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
'*************************************************************************************
' Main program module.
'  This module loads an SR192A project file, executes a timing set/table pair
'  and checks for errors.  The project file in the example is included with the
'  SR192A Development System and is run using this example code.
'
'  This example was written in VB 5.0.
'    Make sure that the SR192A object is checked in the References
'    window (The References window selection is on the Project Menu).
'*************************************************************************************
'
Sub Main()

Dim status As Long          ' Status variable
Dim drv As SR192A           ' Defines variable "drv" to be a SR192A Object

Set drv = New SR192A        ' Make an instance of SR192A Driver Object

' The following call loads a demo project file into the SR192A.  Since no SR192A was
' selected prior to the call, the call automatically defaults to module 1.
status = drv.ssSrLoadProject("c:\program files\sr192aDevSys4\HelloSR.srp")
If status <> 0 Then
   Debug.Print status         ' Test for error, print error code in debug window
End If

' Start execution.  Use timing generator A; timing set named "Go_Fast";
' table named "Ramp" and run for one full cycle.
status = drv.ssSrExecute("A", "Go_Fast", "Ramp", 1)
If status <> 0 Then
   Debug.Print status         ' Test for error, print error code in debug window
End If

' Test for error. Report the first 10 errors only and store
' them in a file named "fault.res"
status = drv.ssSrGetExecResults("A", " Ramp", 10, "fault.res", 0)
If status = 1 Then Debug.Print "Pass"     ' Print "Pass" in debug window
If status = 0 Then Debug.Print "Fail"     ' Print "Fail" in debug window

Set drv = Nothing             ' Destroy instance of SR Driver Object

End Sub
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.6.2 Visual C++ Example

The following source sample shows the minimum code required to load a project file, execute a timing set/table pair and report any test failures. The key program calls are bolded to make them stand out from the comments and error check code. Note that a project file and additional support files are required before this will execute in the Visual C++ (V6) development environment. This code and its support files are included in the examples directory (<Install_Dir>\Examples\Visual_C++) that is installed with the SR192A Development System. For a more comprehensive example of a Windows application, look at the Test Manager Visual Basic source code.

```
//*****************************************************************************
//
//   This code module contains a very simple example of interfacing with the
//   ssSr192Ax driver DLL.  It is intended as a starting point for exploring the
//   driver's functionality.  Consequently, it has no user interface elements
//   and limited error resolving mechanisms.
//
//   The ssSr192Ax function references are made accessible by importing a
//   type library (ssSr192Ax.TLB)
//
//   This code was developed as a console application using Microsoft C++ v6.0
//
// Note C++ conventions:
//   Comments begin with a double right slash (//).
//
//*****************************************************************************

// This header is automatically generated by console application wizard.
#include "stdafx.h"

// Program includes.
#include <iostream>
#include <string>

// Import ssSr192Ax.DLL interface type library.
#import "c:\Program Files\Sr192aDevSys4\ssSr192Ax.tlb"

// Global BSTR
BSTR fileName;
BSTR wtgp;
BSTR wts;
BSTR wtable;
BSTR faultFileName;

#define BSIZE 255      // buffer size constant

// Declare ssSr192Ax.dll namespace.
using namespace ssSr192Ax;
using std::cout;
using std::cin;

// Protos
void CleanExit(void);
void GetErrorAndQuit(_SR192APtr pSR, long stat, char* context);

int main(int argc, char* argv[])
{
    long status;

    // First, initialize the COM library; if error, report it and exit.
    if (FAILED(CoInitialize(NULL)))
    {
        cout << "COM Library initialization failed" << "\n";
        return 0;
    }

//*****************************************************************************
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
// Main program module.
//  This module loads an SR192A project file, executes a timing set/table pair
//  and checks for errors.  The project file in the example is included with the
//  SR192A Development System and is run using this example code.
//
//  This example was written in Microsoft C++ V6 as a console application. .
//
//*****************************************************************************

    cout << "GO -->>>\n";
    // The try/catch isolates unexpected COM errors from ssSr192Ax.DLL errors
    try
    {
        // The following makes an instance of the SR192A object and
        // defines variable "pSR" to be a SR192A Object pointer.
        _SR192APtr pSR(__uuidof(SR192A));

        // Make a UNICODE string containing project file name.
        fileName = SysAllocString(L"c:\\program files\\sr192aDevSys4\\HelloSRa.srp");

        // Load a demo project file into the SR192A.  Since no SR192A was selected
        // prior to this call, the operation automatically defaults to module 1.
        status = pSR->ssSrLoadProject(fileName);

        // Test for load project error. If error, output
        // error message to console window.
        if ( status < 0 )
            GetErrorAndQuit(pSR, status, "Error loading project: ");

        // Start execution.  Use timing generator A; timing set named "Go_Fast";
        // table named "Ramp" and run for one full cycle.

        // First, make UNICODE strings of variables.
        wtgp = SysAllocString(L"A");
        wts = SysAllocString(L"Go_Fast");
        wtable = SysAllocString(L"Ramp");

        // Run SR
        status = pSR->ssSrExecute(wtgp, wts, wtable, 1);
        if ( status < 0 )
            GetErrorAndQuit(pSR, status, "Error SR Execute: ");

        // Wait for the SR192A to finish running
        do
        {
            status = pSR->ssSrStatus ();
        } while ( status != 0 );

        // Test for error.  Report the first 10 errors only
        // and store them in "fault.res" file.
        faultFileName = SysAllocString(L"fault.res");
        status = pSR->ssSrGetExecResults(wtgp, wtable, 10, faultFileName, 0);
        if ( status < 0 )
            GetErrorAndQuit(pSR, status, "Error GetExecResults: ");

        // Check for I/O channel errors and report test result
        if ( status == 0 )
        {
            FILE *fp;
            char inBuf[BSIZE];
            cout << "FAIL: Error report follows...\n";
            fp = fopen("fault.res","r");          // Open file for read
            // Read a line and print a line; loop until end of file.
            while ( !feof( fp ) )
            {
                // Read a line; max size 254; NULL is returned for empty line
                if ( fgets(inBuf, BSIZE, fp) != NULL )
                    // Print a line if there is something there
                        cout << "Test Step/Signal Name -> " << inBuf ;
            }

            fclose(fp);                       // Close the file
        }
        else
        {
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
            cout << "PASS: No errors to report...\n";
        }
    }

    // Something went wrong with COM error handler
    catch(const _com_error& Err)
    {
        cout << "COM Error: "<< Err.ErrorMessage() << "\n";
    }

    // Go to common exit
    CleanExit();
    return 0;
}

//*****************************************************************************
// Retrieve and display SR192A driver error message then halt program.
//*****************************************************************************

void GetErrorAndQuit(_SR192APtr pSR, long stat, char* context)
{
    BSTR wStringBuf;
    int anInt;
    short ccs;
    char asciBuf[BSIZE];

    // Initialize variables
    wStringBuf = SysAllocStringLen(NULL, BSIZE);

    pSR->ssSrGetErrorMessage(stat, &wStringBuf, BSIZE);
    ccs = SysStringLen(wStringBuf);          // Get string length

    // Convert UNICODE to string
    anInt = WideCharToMultiByte(CP_ACP,0,wStringBuf,ccs,asciBuf,BSIZE,NULL,NULL);
    asciBuf[ccs] = NULL;                 // Null terminate it

    // Display error message and quit program
    cout << context << asciBuf << "\n";
    CleanExit();
}

//*****************************************************************************
// Clean up object handle, servers, allocated memory and bail out.
//*****************************************************************************

void CleanExit()
{
    int inChar;

    // Free previously allocated strings
    SysFreeString(fileName);
    SysFreeString(wtgp);
    SysFreeString(wts);
    SysFreeString(wtable);
    SysFreeString(faultFileName);

    // Destroy instance of SR Driver Object
    CoUninitialize();
    cout << "Done!\n";

    // Pause to keep the console I/O visible.
    cout << "Press the Enter key to continue...\n";
    inChar = getchar();

    exit(0);
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

10. Appendix A - Version Changes

The following are lists of changes that have occurred from previous versions of the SR192A Development System.

Changes from Version 4.0 to Version 4.05

1. The SR192A Development System now includes an updated on-line help file.

2. An error when copying timing sets between A and B timing modules has been corrected.

3. Corrections have been made to the Table Editor for importing VCD and Expected State files.

4. An example Visual C++ project has been added to the installation.

5. Most dialog boxes can now access help (**F1**) and print a copy of themselves (**Ctrl-P**).

6. The Signal List Editor has been updated to improve its response to dynamic configuration changes.

7. Various improvements have been made to the SR192A ActiveX driver (**v1.00.0009**) to support different configurations and error recovery. Also, several obsolete functions were removed from its public interface.

Changes from Version 4.0 Beta to Version 4.0

1. The SR192A Development System now supports setting a 100 MHz clock

2. Error Memory control and Clock Output selection are now supported by the timing generator's Advanced Settings dialog. On the same dialog, Vector Channels can now be set to zero (0) in order to disable them.

3. The Test Manager now displays an Operation indicator on its status bar. The three possible entries are Independent, Master or Slave.

4. The Signal List now allows the SR124A differential checkboxes to be drag-selected and set as a group. Use the toolbar “H” button to set them for differential and the “L” button to set them for TTL.

5. Various improvements have been made to the SR192A ActiveX driver to support Master/Slave operation and to keep pace with SR192A firmware and Plug&Play changes.

Version 4.0 Beta

1. This is the initial release of the SR192A Development System software. Since it was developed from version 3.0 of the original SR192 software, it is being released as version 4.0.

2. The on-line help files have not yet been updated to include new features of the SR192A. Many of the primary software operations are already covered in the help file. New features are generally self-explanatory. For information on the new capabilities of the SR192A, consult Talon’s reference manuals.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11. Sales & Support

For more information contact:

Talon Instruments

150 East Arrow Highway
San Dimas, CA 91773

909-599-0690
909-599-6529 Fax

sales@taloninst.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Index

/

// 8, 41

A

Advanced Settings 19
Appendix A - Version Changes 119
Application Development Environments 109
Assert & Return Wizard 40
Auto Save/Reload 80

C

Cell Test Parameters 18
Cell Wizard 39
check project 8
Context-Sensitive Help 6
copy tables 11

D

DAC 20
Debug Log 14
Delay A/B 18
Development Environment 5
Documentation and Printing 6
doubled signal 41
Driver Updates 92

E

Error Display 44
Error Log File 93
error memory 19
Error Reporting 91
ErrorColor 44
errors 14
Errors Menu 54
Execution Result File Format 86
Execution Results 85
Export Value Change Dump 56
external clock 19

F

Fail Count 44, 85, 86
file formats 14, 59, 75, 93
fill functions 53
FITS Expected State File Format 60

H

help 6
Help Menu 13
hexadecimal 59
HP-VEE 89, 111

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

I

Import & Export Menu 47
Import Value Change Dump 55
instrument 6
Introduction 1

J

JEN 43
Jump Enable & Vector Enable 43
Jump Enables 43
Jump Strobe 19

L

LabView 89, 113
LabWindows 89, 112
LASAR Import 47
log file 14, 93

M

Module Group Control 27
Module Group Properties 28
Modules 16

N

National Instruments LabView 113
National Instruments LabWindows/CVI 112

O

Operation 78
Options Menu 38

P

Parameter Pane 66
Pattern Trigger 23
pin group 21
power failure 10
printing 6
Programming Interface 89
Project Browser 7
  Edit Menu 11
  File Menu 9
  Help Menu 13
  Options Menu 12
  Toolbar 8
project file 9, 10
Project Validation 8

R

recover 10
register 92
regsvr32 92
reload 80
Remove Name 74

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

S

Sales & Support 120
save/reload 80
SCP 72, 81, 105
SCPI file 72, 81, 105
SEQ 68, 75
sequence 63, 64, 96, 102, 105
Sequence Editor 63
  Edit Menu 70
  File Menu 68
  Options Menu 72
Sequence Execution 79
Sequence File Format 75
Sequence List 65
Sequence Operation 64
Signal Delays 29
Signal List Editor 21
  Edit Menu 26
  File Menu 25
  Toolbar 24
Simple Hex File Format 59
SR Module Configuration 15
SR100 64
SR101 ....63, 64
SR192A 6
SR192A Development Environment 5
SR192A Hardware 6
SR192A Menu 82
SR192A_Sync 41
SR192DEV.LOG 14
SRP 9
ssSr192Ax 89, 92, 109
ssSrExecute 95
ssSrExecuteSequence 96
ssSrGetConfig 97
ssSrGetErrorMessage 98
ssSrGetExecResults 100
ssSrGetList 102
ssSrGetTableNames 102
ssSrGetTimingSetNames 102
ssSrHalt 103
ssSrLoadProject 104
ssSrLoadSequence 105
ssSrReset 106
ssSrSelectSR192 107
ssSrStatus 108
Support 120
sync 42
System Architecture 3
System Operation 4

T

Table Editor 41
  Edit Menu 48
  File Menu 46
  Fill Menu 53

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Options Menu 52
Set Menu 50
Toolbar 45
table list 74, 85
table names 11, 64, 95
Talon Instruments 120
Test Manager 77
File Menu 81
Options Menu 84
SR192 Menu 82
Timing Menu 83
Test Signal Sources 18
Timeout 18
Timing Cell Test 33
Timing Counts 18
Timing Generators 17
Timing Menu 83
Timing Set Editor 31
Edit Menu 36
File Menu 35
Set Menu 37
Toolbar 34
timing set list 64, 74, 95
Toolbar 67
trigger 18, 21, 22, 23, 31
truncate name 11, 64, 95, 96

U

unknown signal state 50

V

validate project 8
validate sequences 72
Value Change Dump File Format 57
VCD 55, 56, 57
Vector Channels 19, 73
Vector Table 73
VEN 19, 43, 73
Visual Basic 110
Visual Basic Example 114
Visual C++ 111
Visual C++ Example 116

W

warnings 14
working project file 9

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP