REFERENCE COPY
C. 2

SAND REPORT

SAND2002-1450
Unlimited Release
Printed May 2002

Controlatron Neutron Tube Test Suite Software Manual

Operation Manual (V2.2)

William P. Noel, Debra L. Wallace, Monica. L. Martinez, Robert J. Hertrich, and Keith Barrett

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2002-1450
Unlimited Release
Printed May 2002

Controlatron Neutron Tube Test Suite Software Manual Operation Manual (V2.2)

William P. Noel and Debra L. Wallace
Neutron Tube Development Department

Monica L. Martinez
Neutron Generator Development Department

Sandia National Laboratories
Albuquerque, New Mexico 87185-0516

Robert J. Hertrich and Keith Barrett
Primecore Systems, Inc.
Albuquerque, New Mexico 87111

Abstract

The Controlatron Software Suite is a custom built application to perform automated testing of Controlatron neutron tubes. The software package was designed to allowing users to design tests and to run a series of test suites on a tube. The data is output to ASCII files of a pre-defined format for data analysis and viewing with the Controlatron Data Viewer Application. This manual discusses the operation of the Controlatron Test Suite Software and a brief discussion of state machine theory, as state machine is the functional basis of the software.

Revision History

Rev #	Author	Date	Description
1.0	Keith Barrett	09\04\01	Original Layout and Formatting
1.4	Robert Hertrich	09\08\01	Began Software User Manual Input
1.5	Keith Barrett	09\14\01	First Rough Draft Completed
1.6	Robert Hertrich	09\20\01	Rough Draft Revision 1
1.7	Robert Hertrich	09\27\01	Rough Draft Revision 2
1.8	William Noel Monica Martinez	09\29\01	Rough Draft Revision 3\ Editing \ Format to SNL SAND specifications
2.0	William Noel	10\10\01	Final Draft Revision
2.1	William Noel	10\31\01	Peer Review Revisions
2.2	William Noel	03\02\02	Update figures and formatting

Figure 1. Revision History Table

Robert J. Hertrich
PrimeCore Systems, Inc.
8421 Osuna Rd, NE Suite C-1
Albuquerque, New Mexico 87111

Keith Barrett
PrimeCore Systems, Inc.
8421 Osuna Rd, NE Suite C-1
Albuquerque, New Mexico 87111

William P. Noel, EMT-B, RN
Neutron Tube Development Department
Sandia National Laboratories
PO Box 5800
Albuquerque, New Mexico 87185-0516

Debra L. Wallace
Neutron Tube Development Department
Sandia National Laboratories
PO Box 5800
Albuquerque, New Mexico 87185-0516

Monica L. Martinez
Neutron Generator Development Department
Sandia National Laboratories
PO Box 5800
Albuquerque, New Mexico 87185-0516

Contents

Figures

This page intentionally left blank.

1. Controlatron Software Overview

1.1 Overview

The Controlatron Software Suite is a custom built application to perform automated testing of Controlatron neutron tubes. The software package is capable of allowing users to design tests and to run a series of test suites on a tube. The data is output to ASCII files of a pre-defined format for data analysis and viewing with the Controlatron Data Viewer Application.

The software is designed to specifically control the system hardware defined by the Sandia documents; Controlatron Tube (Wire Reservoir) Test Procedure, drawing #10024353-000 and Product Specification, Controlatron Tube (U), drawing #PS704791. The software only controls the hardware designed with the system, but the source code was written with a modular design that is adaptable to rapid change.

This document provides operating instructions for the user (Section 2) as well as programming notes and descriptions for future development of the system (Section 3).

1.2 Software Specifications

Operating System:	Windows 2000
Programming Language\Version:	LabVIEW 6.02 (LabVIEW 6i with the 6.02 update patch)
NIDAQ Version:	NIDAQ 6.9 (National Instruments)
FieldPoint Version:	Fieldpoint 2.0 (National Instruments)
Message Manager Version:	v5.0.6
GPIB Version:	NI 488.2 Version 1.70
Executable Program Name:	Controlatron Test Suite Rev*X_X*.exe
Top Level Program Name:	CCC Top Level.vi
Data Viewer App Name:	Controlatron Data Viewer Rev*X_X*.exe
Minimum System Requirement:	Pentium II, 400MHz or better PC
Memory Requirement:	128 Meg Minimum, 256 Meg recommended
Network Requirement:	TC/PIP protocol network connection (for data storage)

1.3 System Administrators

Name	Phone	Email	Company
William P. Noel	505-845-8796	wpnoel@sandia.gov	SNL
Robert Hertrich	505-294-0010	rhertrich@primecoresystems.com	PrimeCore Systems, Inc.
Keith Barrett	505-294-0010	kbarrett@primecoresystems.com	PrimeCore Systems, Inc.

Figure 2. System Administrators

1.4 Hardware Considerations

This document does not address the hardware configuration of the Controlatron tube as it is placed or operated within the test fixture. Please note that the operation of the Controlatron Test Suite Software within the facility is considered a hazardous operation due to high voltages used and the production of radioactive flux. Appropriate safety procedures must be in place and adhered to during the operation a Controlatron neutron tube.

2. Users Guide

2.1 Launching the Controlatron Software

Prior to testing Controlatron neutron tubes using the Controlatron Test Facility and the Controlatron Test Suite Application, several operational safety precautions that must be adhered to:

1. Visually inspect the hardware components of both the Acquisition System (upstairs) and the PCS 01 001A Optically Isolated, Multi-Output Power System (Power Rack downstairs). If any hardware component of the Controlatron Test Facility is visibly damaged or missing, **DO NOT** operate this test facility. Immediately contact a system administrator (Section 1.3 of this manual) for assistance in restoring the test facility to a fully operational system.

2. Visually inspect the communications connections from each device to the computer to ensure that all connections are made. While inspecting communication connections to and from the Power Rack, ensure all fiber-optic connections are secure as well as connections from fiber optic modems to individual instruments.

3. Manually turn on power to the Pulizzi distribution center of the Power Rack, this will power-up the high voltage power supplies. **Note:** During the power up phase of the Pulizzi, power will be cycled to each outlet. Use caution while powering up the Pulizzi power strip. After each outlet has been turned on, immediately turn it back off manually. Power to the Pulizzi needs to be on, but for safety, no outlets should be left on when the rack is manually initialized.

4. Manually, turn on power to the computer and allow it to fully boot.

5. Manually, turn power to the Pulizzi distribution center of the Acquisition System. This power strip will cycle through all of its outlets as well, but there is no danger in allowing the acquisition system to fully power up and leaving the Pulizzi in the full powered up state.

Now that system instrumentation has been visually inspected and power has been supplied to all components in the test facility, the Controlatron Test Suite Application can be launched. Launching this application can be accomplished by double-clicking on the application's Icon (located on the desktop).

2.2 Logging In

After you launch the program it will prompt you to log onto the system. A dialog box shown below appears.

Figure 3. User Logon Pop-up Panel

Click on the user name and select account name. Type in password below and press OK. If your password is correct the program will proceed, if it is not then it will ask you to try again or cancel.

Note: If the cancel button is pressed on the User Logon screen, the application will be exited. Only users with 'Administrative' or 'Power User' privileges are permitted to adjust user accounts (see Section 2.5 Administrative Options).

2.3 Specifying Test Device

Once the user has successfully logged in, the Initialize Test Session panel appears. This panel is used to enter the specific definition related to the Controlatron being tested. Pb probe definition and positional information is captured in this panel, as well as system configuration variables.

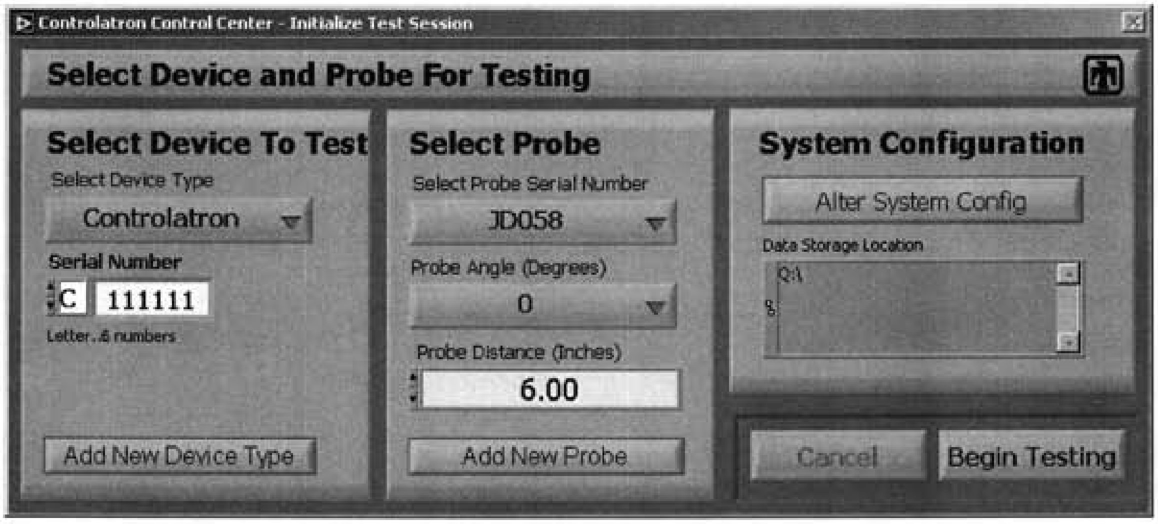


Figure 4. Initialize Test Session Pop-up Panel

This panel is segmented into four logical groups of controls:

2.3.1 Select Device to Test

The 'Select Device to Test' group of controls describes the type of device under test and the serial number (SN) of the device that is to be tested.

- **Select Device Type:** The user can use this pull down control to select the type of device that is to be tested from the list that appears. Device types are typically; Controlatron, TTA or probe.
- **Serial Number:** Specify the exact serial number of the device under test in these controls. This alphanumeric code consists of one letter (pull down selected) and six alphanumeric characters. To insert the alphanumeric characters, first left-click within the text box to make it active, and then type the desired alphanumeric characters into the control.
- **Add New Device Type:** This button is used when the type of device to be tested has never been tested on this system and there is not a selection for it within the 'Select Device Type' pull down menu. Pressing this button invokes the following pop-up panel for the user to specify the new device type.

Figure 5. Prompt for New Device Type

The user enters the new devices type name into the text control and presses the enter button or presses the <Return Key> on the keyboard. This action returns the user to the former screen (Initialize Test Session panel) and the new device type name will now be available in the pull down menu. If the 'cancel' button is pressed, or the <ESC> key on the keyboard is pressed, the user is returned to the Initialize Test Session panel and no adjustments will be made to the list of device types.

2.3.2 Select Probe

- **Select Pb Probe Serial Number:** The user can use this pull down control to select the serial number of the Pb Probe that is being used during the testing.

- **Probe Angle (Degrees):** This pull down control permits the user to define the angle at which the Pb probe is oriented to the Controlatron neutron tube target.

- **Probe Distance:** This pull down control permits the user to define the distance between the probe and the Controlatron neutron tube target.

- **Add New Probe:** Functionality is similar to the 'Add New Device Type' in the 'Select Device to Test' portion of this screen, however this is to be used if the Serial Number of the presently used Pb probe is not available in the 'Select Probe Serial Number' pull down control above.

2.3.3 System Configuration

- **Configuration File Adjuster:** This panel is only available to users that have 'Administrative' or 'Power User' privileges. This panel is used to manage communication variables for all devices in the Test Facility as well as the Data Storage Location path and application paths.

  Note: Settings manipulated on this panel are very sensitive to the functionality of the Controlatron Test Suite Application. Use Caution when adjusting parameters available on this panel!

- **Hardware Address Settings:** Refer to hardware specific address settings and must match the hardware's settings correctly for the application to function. Running the *Measurement and Automation Explorer* before entering the Controlatron application, you can scan for instruments, then all the addresses found will be pull down options. If the desired address isn't there, type it in.

  Default Hardware Address Settings:

Acquisition System	Address	Power System	Address
Acquisition Pulizzi	COM 11	Power Pulizzi	COM 12
SR 430	GPIB 0:5	Bertan 1	GPIB 1:10
DG535 1	GPIB 0:6	Bertan 2	GPIB 1:11
DG535 2	GPIB 0:7	SR 1	GPIB 1:12
Scope 1	GPIB 0:8	SR 2	GPIB 1:13
Scope 2	GPIB 0:9	Xantrex 1	GPIB 1:14
VTR Com Port	COM 13	Xantrex 2	GPIB 1:15
		Xantrex 3	GPIB 1:16

Acquisition System	Address	Power System	Address
		Xantrex 4	GPIB 1:17
		Xantrex 5	GPIB 1:18

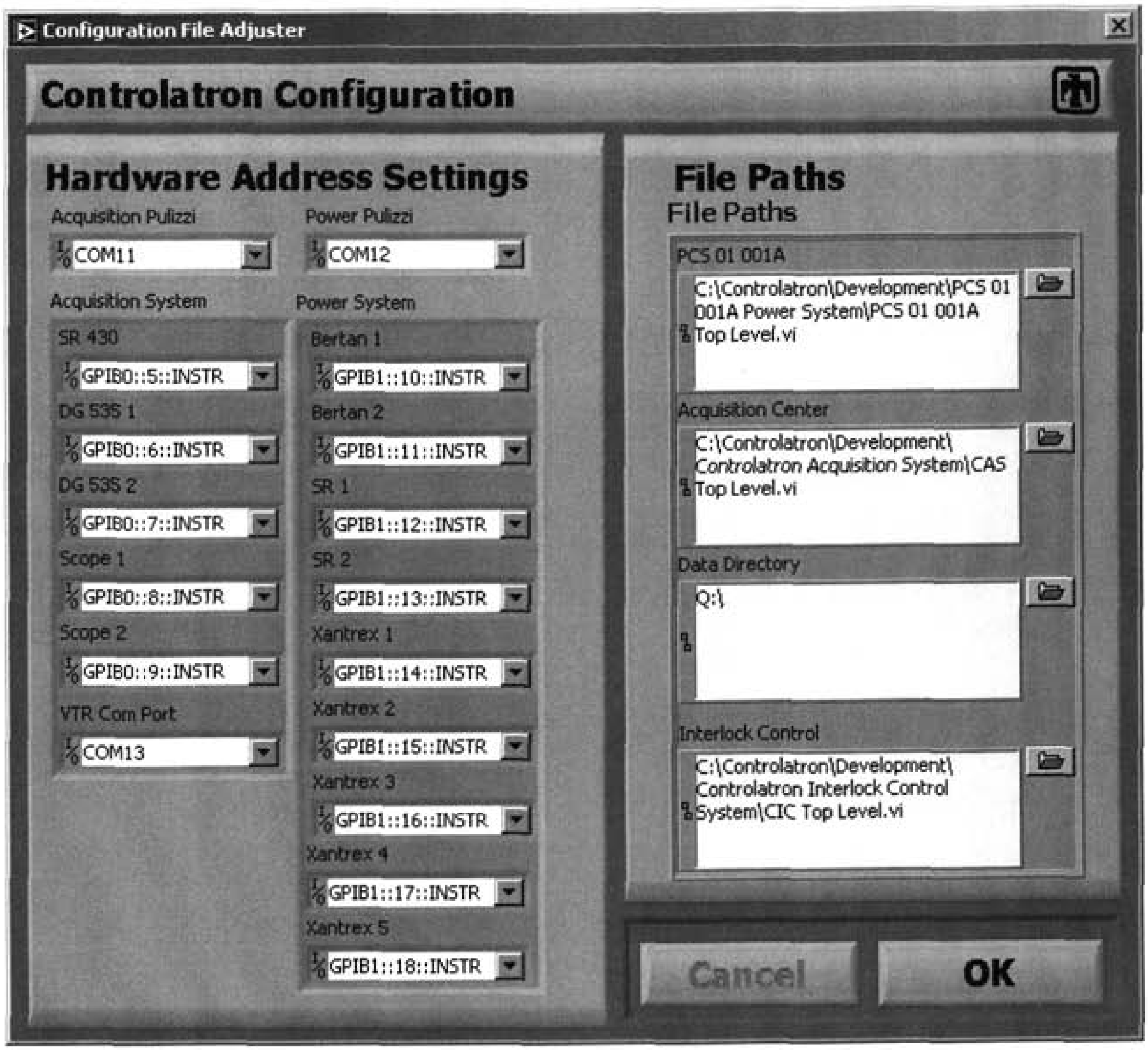


Figure 6. Configuration File Adjuster

- **File Paths:** File Paths refer to the locations in the system where module components of the Controlatron Test Suite Application are located as well as the top directory path where data will be stored. These data paths must be exact for the application to function properly.

  Default File Path Settings:

System Component	Path
PCS 01 001A	C:\Controlatron\Development\PCS 01 001A Power System\PCS 01 001A Top Level.vi
Acquisition Center	C:\Controlatron\Development\Controlatron Acquisition System\CAS\top Level.vi
Interlock Center	C:\Controlatron\Development\Controlatron Interlock Control System\Top Level.vi

- **Data Storage Location** This indicator shows the user where collected data will be stored. This value is determined by a system administrator and is not available for manipulation by a standard user.

  Default File Path Settings:

System Component	Path
Data Directory	Q:\ Network remote mount to: \CSU891\Cpq891d\807Data –OR– Local Data Storage: C:\Data

2.3.4 Action Buttons

The 'Cancel' button and the 'Begin Test' button cause application actions.

- **Cancel:** This button exits the user from the application without affecting the test device or the application.

- **Begin Test:** Pressing this button causes several verification processes prior to launching the complete Controlatron Test Suite Application. These verifications include:

  1. Checking that there is a proper path for storing test data collected for the specified device under test (DUT). If the data path is incomplete for the specified DUT, the system will prompt the user to choose if the path should be dynamically created, or if the 'Begin Test' action should be cancelled.

  2. Verifying that the specified probe has been calibrated on today's date. If the specified probe has not been calibrated today, the user will be prompted to enter a User Defined Background

     By canceling out of the Background Average Pop-up the user may calibrate the Pb probe dynamically by collecting data and calculating an average from a shot log file.

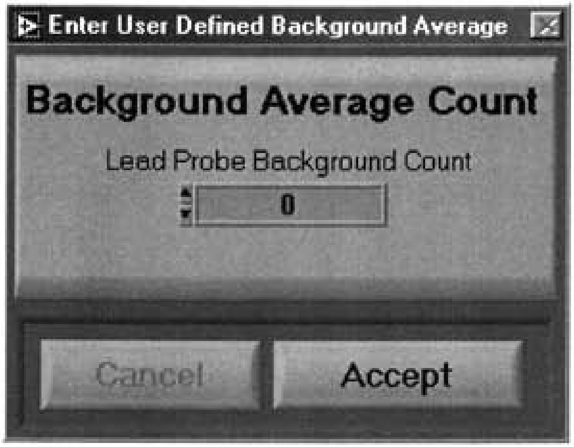


Figure 7. User Defined Background Average Pop-up

3. If the probe has been calibrated more than once today, the user will be prompted to select which calibration file (shot log) (or combination of calibration files) for calculating the background average count.

   Note: When multiple calibration files are located for a Pb probe, the most recent file in the list does not have a number in the file name. The earliest calibration file has *001 appended to it.*

   In Figure 8, A123456_Shot Log_001 was the first calibration attempt, A123456_Shot Log_002 was the second, and A123456_ Shot Log was the third and most recent calibration.

4. Pb background count will be in the 150 to 225 range, nominally 190. This value will change with Pb probe, scalar average, and associated equipment that is changed or calibrated.

5. If the selected test device type is a 'Probe' and the probe has already been calibrated for the day, the user will be given a choice to append this test's data to the current probe file or to create a new probe calibration file.

Note: If you have selected to calibrate a probe, you will be prompted to enter the Estimated Background Average Count. This value is used to generate actual background count averages. This value is not known by the system; therefore, it is the duty of an experienced test engineer to provide this value. For further information about this value, contact a system administrator.

Once the specifications and procedures defined in Sections 2.1, 2.2, and 2.3 have successfully been performed, the Controlatron Test Suite application is loaded into the system. This process takes a few moments as indicated on the 'Initializing Controlatron Test Suite Components pop-up panel.

Once the entire application is loaded the user can begin utilizing the Controlatron Test Facility to perform functional tests on the Controlatron neutron tube.

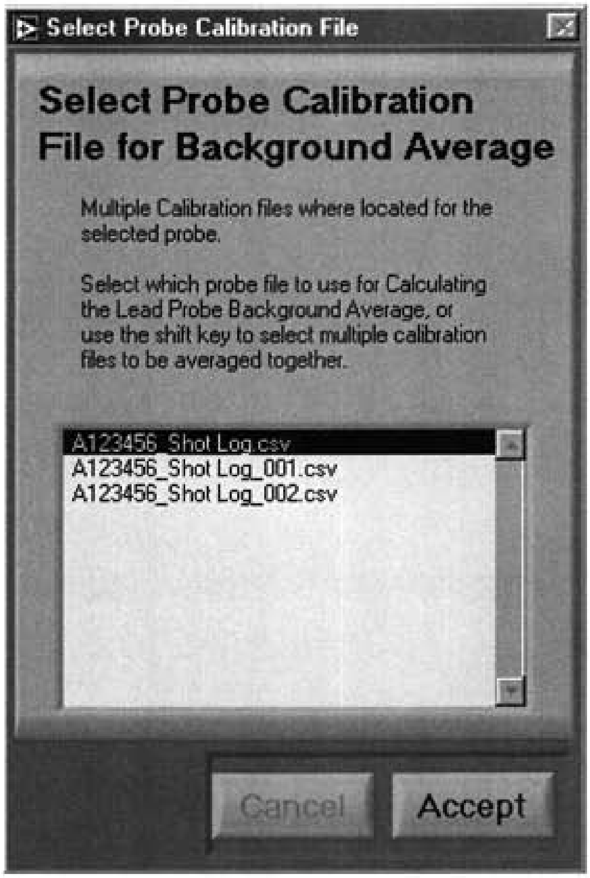


Figure 8. Select Probe Calibration File

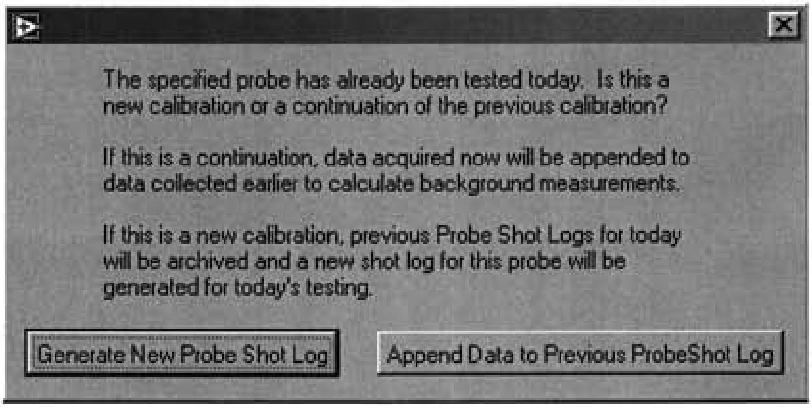


Figure 9. Generate New Probe Shot Log Pop-up Dialog

2.4 Controlatron Control Center

The Controlatron Control Center (CCC) is considered the 'Main Panel' of the Controlatron Test Suite application. From this panel, the user can select from predefined Power Rack configurations as well as Acquisition System configurations to set the hardware systems up, arm the system, trigger the system, and view data collected as the result of a test. Ideally, minimum use of configuration modules (Sections 2.9, 2.10, and 2.11 of this manual) will be required.

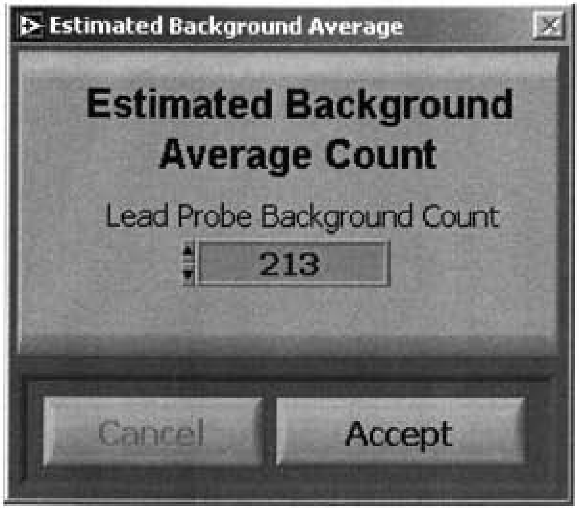


Figure 10. Estimated Background Average Count Dialog

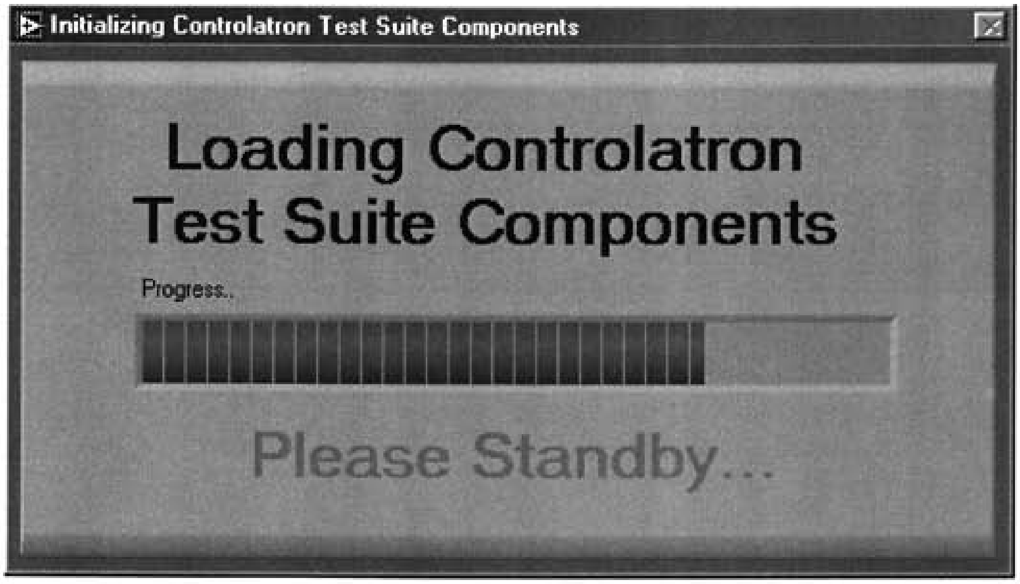


Figure 11. Initializing Controlatron Test Suite Components Pop-up Panel

This Panel is separated into five logical groups:

- **Pull Down Menus:** The pull down menus, available on the menu bar at the top of the screen, offer configuration options as well as functional application control.

- **Testing Controls:** Manage all of the automated testing procedures and high-level test variables.

- **Acquisition System:** Offers a quick status view of the current Acquisition System State as well as some functional control relating to the Acquisition System.

- **Power System:** Offers a quick status view of the current Power System State and the Interlock System as well as offering some functional control relating to the Power System (E-Stop).

- **Device Information:** Indicates device specific information as specified the by the user on the 'Initialize Test Session' panel (Section 2.3). This section also permits the user to make comments about the previous shot that was taken.

2.4.1 Pull Down Menus

Figure 13 shows the options available from the pull down menus.

2.4.1.1 File Menu

- **File – Print Processed Data**

  This option causes a printout of processed data to be generated when data is collected by the system. By default, this option is selected when the Controlatron Test Suite is started. If a processed data printout is not desired for each shot, the user must 'un-select' this menu item.

- **File – Exit**

  Exits the user from the application. Aside from canceling the Logon Screen (Section 2.2) or canceling while initializing a test session (Section 2.3), this is the only method for shutting the system down. While the system is being shut down the user is prompted with the option of not automatically shutting the power off to the Acquisition System.

  If the user selects to leave the Acquisition System's power on, the application will be exited but the power state of the acquisition will be left unaffected, otherwise, the entire system will be turned off.

  Note: There is not an option to leave the power state of the Power Rack on. The power system will be completely shut down (minus communication devices) when the Test Suite is exited.

2.4.1.2 Configuration\View Menu

- **Configuration\View – Change Test Object**

  Invokes the Initialize Test Session Screen (Section 2.2) so the user can re-define the test object and other variables regarding the next test session without restarting the application.

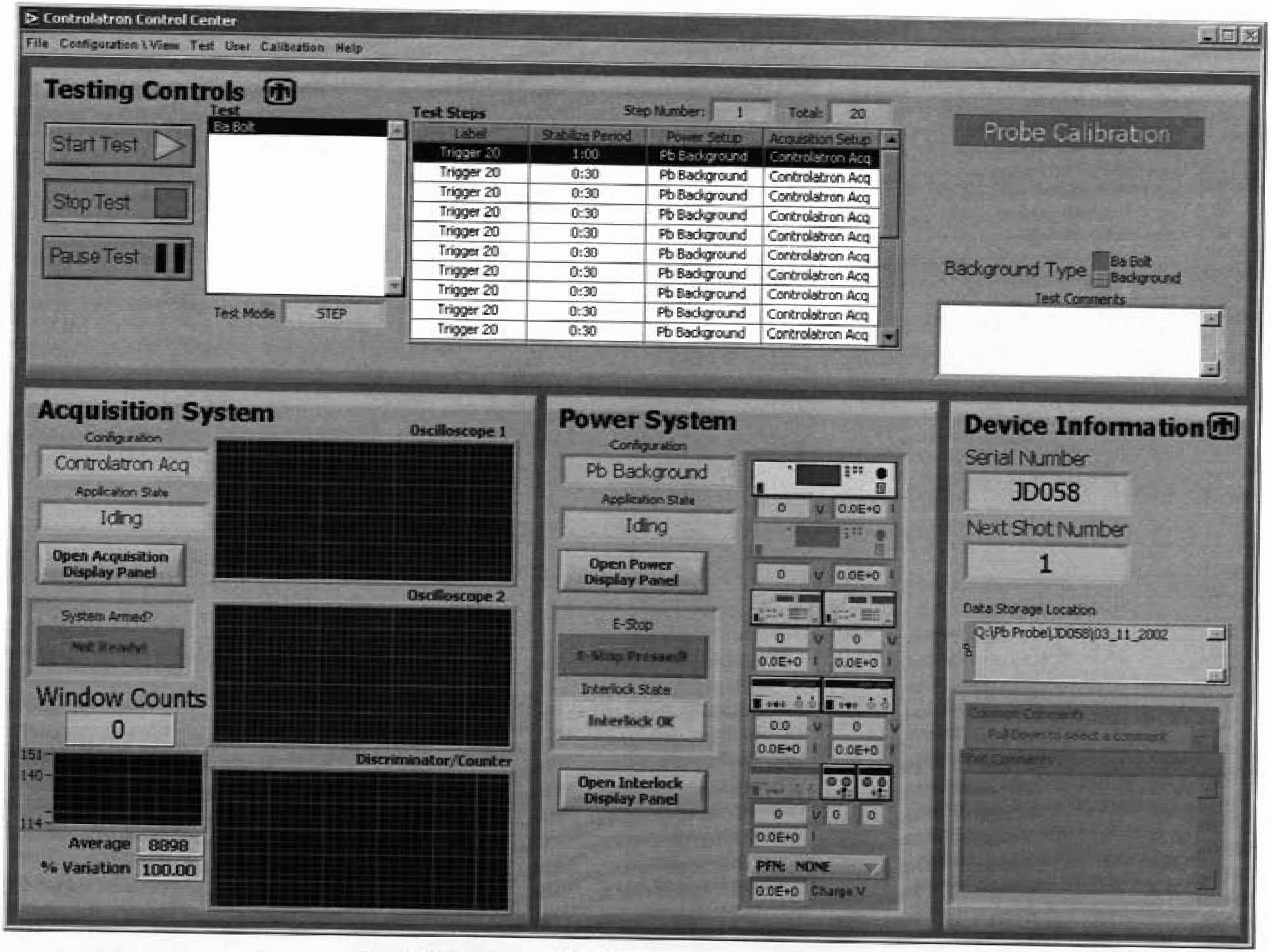
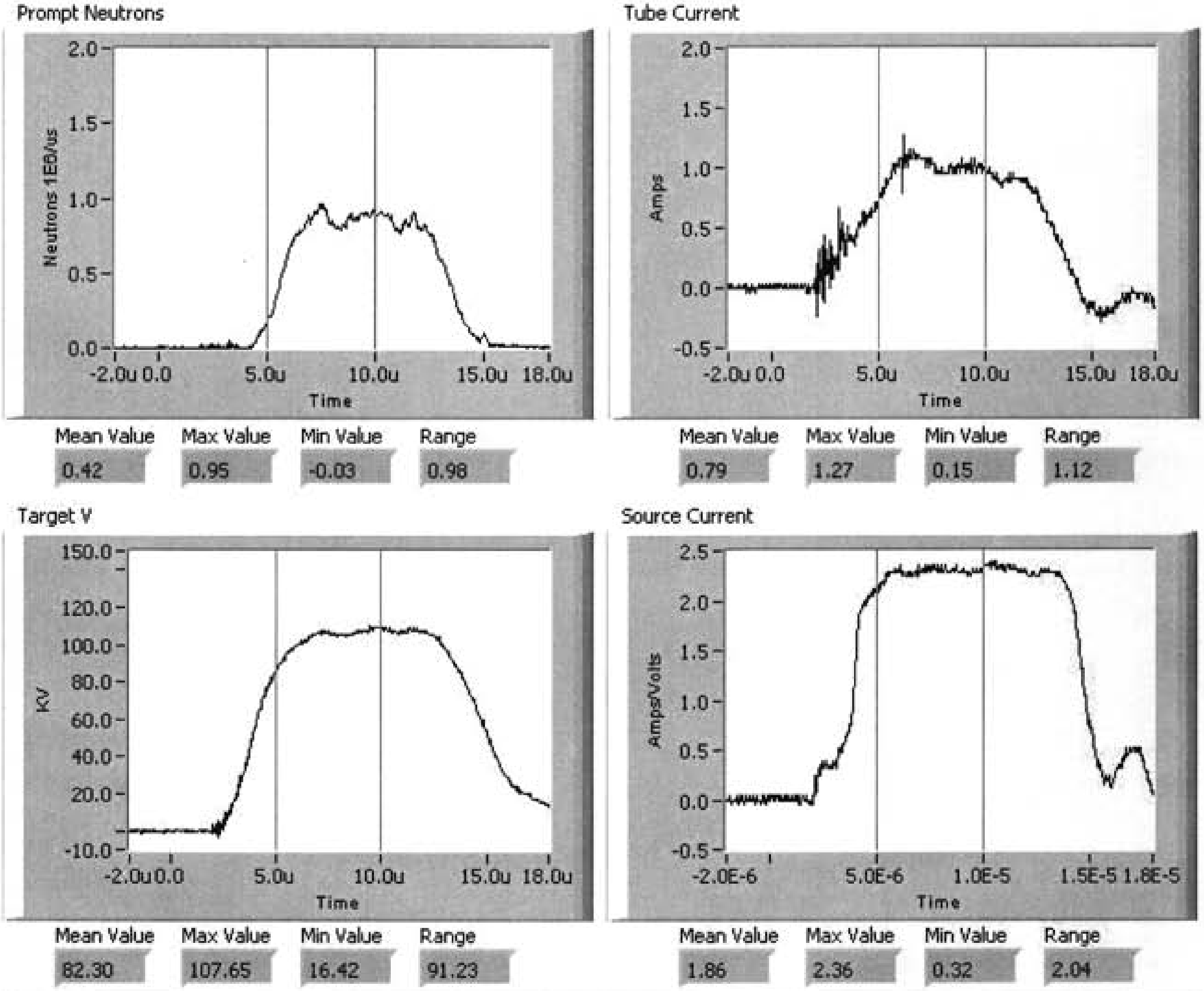


Figure 12. (Previous Page) Controlatron Control Center

File	Configuration \ View	Test	User	Calibration	Help
Print Processed Data	Change Test Object	Test Mode:	User Manager	Pb Probe Calibration	Show Context Help
Exit	Control Center Acquisition	Step	User Logon		About
	PCS 01 001A Power System	Auto			
	Interlocks	Modify Test			
	Probe Location	Add Test			
	View Shot Log	Remove Test			
	View Optimum Res Current Plot	Create New Test			
	Processed Data Plot Variables				
	Configure System Sounds				

Figure 13. Pull Down Menu Items available on Controlatron Control Center Menu Bar

Test Engineer	Serial Number	File Name
William P. Noel	C010547	Controlatron_C010547_218



Date	2/19/2002	Shot Time	12:51 PM
Device type	Controlatron	Serial Number	C010547
Shot Number	218	Test Engineer	William P. Noel
Total Count	2152	Scaler Background Average	190
Calculated Ba Bolt Counts	6673	Ba Bolt Decay Factor	0.890558
Charge Voltage	5.0	Set Charge Voltage	5000.0
Cal Factor	3496.9	Original BA 133 Cal Count	7280
Tube Voltage	NaN	Calculated Neutrons	6860872
Reservoir Current	2.05	Set Reservoir Current	2.05

Test Comments

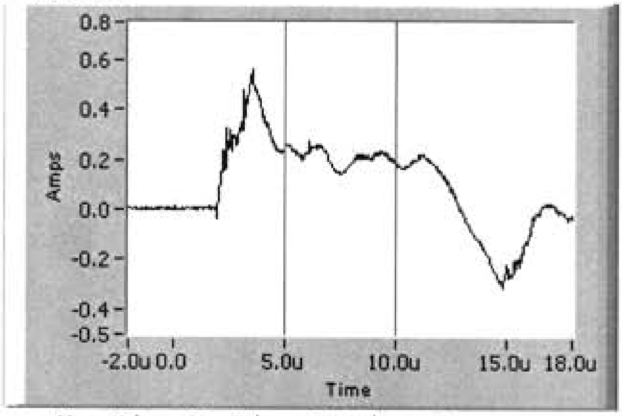
Page 1

Retest after error flag

Figure 14. Processed Data Printout generated when data is collected (Page 1)

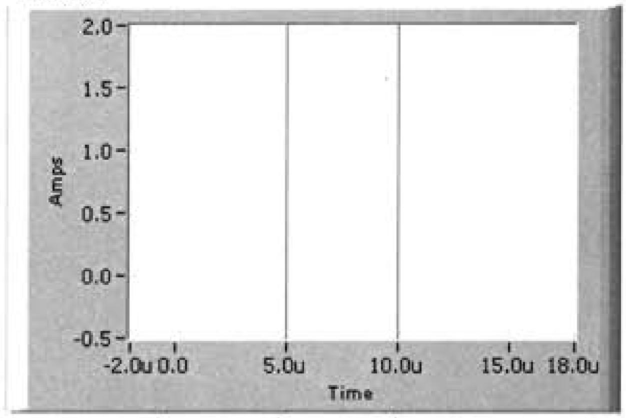
Test Engineer	Serial Number	File Name
William P. Noel	C010547	Controlatron_C010547_218

Grid Current



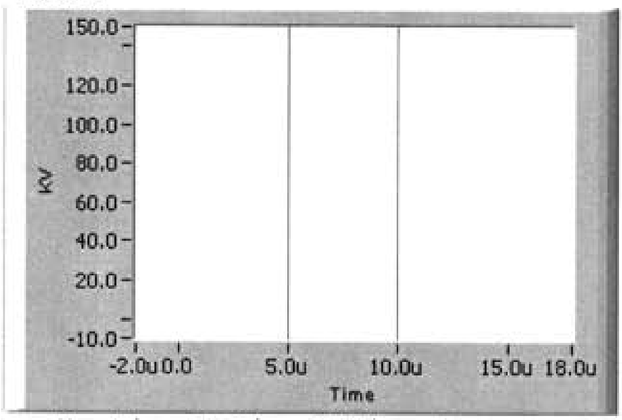
Mean Value	Max Value	Min Value	Range
0.27	0.56	0.14	0.42

Not Used



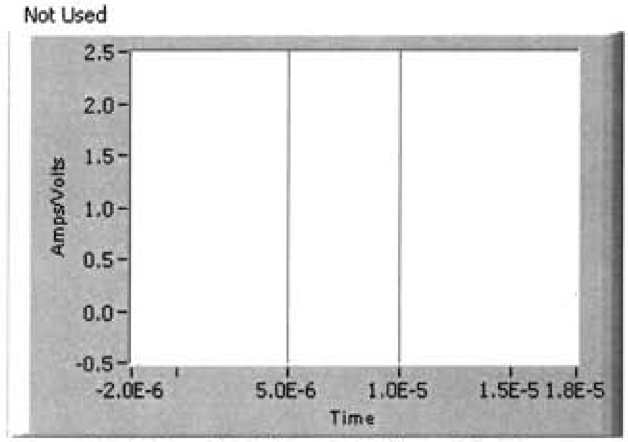
Mean Value	Max Value	Min Value	Range
0.79	1.27	0.15	1.12

Not Used



Mean Value	Max Value	Min Value	Range
82.30	107.65	16.42	91.23

Not Used



Mean Value	Max Value	Min Value	Range
1.86	2.36	0.32	2.04

Date	2/19/2002	Shot Time	12:51 PM
Device type	Controlatron	Serial Number	C010547
Shot Number	218	Test Engineer	William P. Noel
Total Count	2152	Scaler Background Average	190
Calculated Ba Bolt Counts	6673	Ba Bolt Decay Factor	0.890558
Charge Voltage	5.0	Set Charge Voltage	5000.0
Cal Factor	3496.9	Original BA 133 Cal Count	7280
Tube Voltage	NaN	Calculated Neutrons	6860872
Reservoir Current	2.05	Set Reservoir Current	2.05

Test Comments

Retest after error flag

Page 2

Figure 14. Processed Data Printout generated when data is collected (Page 2)

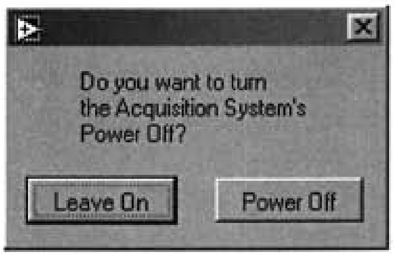


Figure 15. Pop-up option generated when system is exited

- **Configuration\View – Control Center Acquisition**

  Invokes the Control Acquisition System configuration panel (Section 2.9)

- **Configuration\View – PCS 01 001A Power System**

  Invokes the Power Control System configuration panel (Section 2.10)

- **Configuration\View – Interlocks**

  Invokes the Interlock System configuration panel (Section 2.11)

- **Configuration\View – Probe Location**

  Invokes the Probe Location pop-up panel. This pop-up panel permits the user to re-define how the probe is oriented in relation to the Controlatron neutron tube target end.

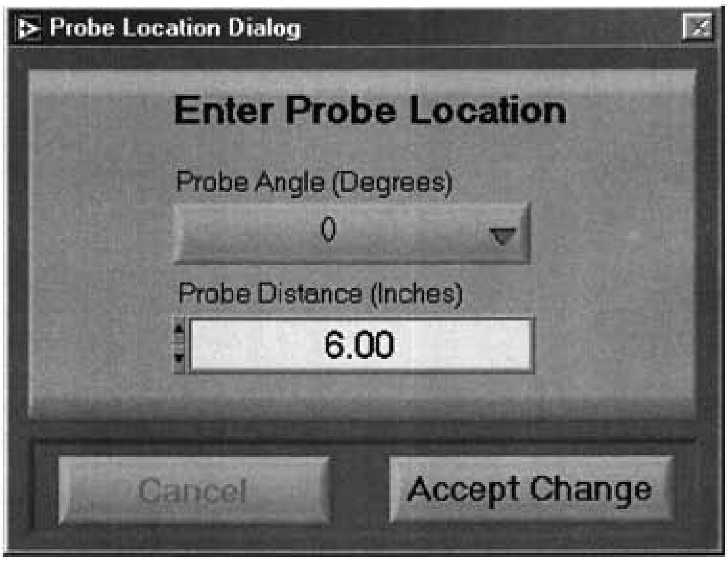


Figure 16. Probe Location Dialog Pop-up Panel

The variables manipulated on this panel are the same that were defined while initializing the test session (Section 2.3.4).

File Name

C010547_Shot Log.csv

Date	Shot Time	Device Type	Serial Number	Shot Number	Test Engineer	Total Count	Scaler Background Average	Ba Bolt Counts	Decay Factor	Charge Voltage	Set Charge Voltage	Cal Factor	BA 133 Calibration Count	Tube Voltage	Calculated Neutrons	Reservoir Current	Set Reservoir Current
12/5/2001	8:47 AM	Controlatron	C010547	1	William P. Noel	1416	225	226	0.000146	0.0	4000.0	3496.876	7280	-88402.1	4164780	2.00	2.00
12/5/2001	8:48 AM	Controlatron	C010547	2	William P. Noel	2200	225	226	0.000146	0.0	4000.0	3496.876	7280	-87663.9	6906331	2.00	2.00
12/5/2001	8:49 AM	Controlatron	C010547	3	William P. Noel	2181	225	226	0.000146	0.0	4000.0	3496.876	7280	-87903.6	6839890	2.00	2.00
12/5/2001	8:50 AM	Controlatron	C010547	4	William P. Noel	2186	225	226	0.000146	0.0	4000.0	3496.876	7280	-88267.9	6857375	2.00	2.00
12/5/2001	8:51 AM	Controlatron	C010547	5	William P. Noel	2193	225	226	0.000146	0.0	4000.0	3496.876	7280	-88450.1	6881853	2.00	2.00
12/5/2001	8:57 AM	Controlatron	C010547	6	William P. Noel	1845	225	226	0.000146	0.0	4000.0	3496.876	7280	-86302.6	5664940	2.05	2.05
12/5/2001	8:58 AM	Controlatron	C010547	7	William P. Noel	1895	225	226	0.000146	0.0	4000.0	3496.876	7280	-86801.1	5839784	2.05	2.05
12/5/2001	8:59 AM	Controlatron	C010547	8	William P. Noel	1878	225	226	0.000146	0.0	4000.0	3496.876	7280	-86475.1	5780337	2.05	2.05
12/5/2001	9:00 AM	Controlatron	C010547	9	William P. Noel	1984	225	226	0.000146	0.0	4000.0	3496.876	7280	-86350.5	6151006	2.05	2.05
12/5/2001	9:01 AM	Controlatron	C010547	10	William P. Noel	1898	225	226	0.000146	0.0	4000.0	3496.876	7280	-86686.0	5850274	2.05	2.05
12/5/2001	9:06 AM	Controlatron	C010547	11	William P. Noel	1547	225	226	0.000146	0.0	4000.0	3496.876	7280	-83723.6	4622871	2.10	2.10
12/5/2001	9:08 AM	Controlatron	C010547	12	William P. Noel	1495	225	226	0.000146	0.0	4000.0	3496.876	7280	-83877.0	4441033	2.10	2.10
12/5/2001	9:09 AM	Controlatron	C010547	13	William P. Noel	1538	225	226	0.000146	0.0	4000.0	3496.876	7280	-84049.6	4591399	2.10	2.10
12/5/2001	9:10 AM	Controlatron	C010547	14	William P. Noel	1478	225	226	0.000146	0.0	4000.0	3496.876	7280	-83819.5	4381586	2.10	2.10
12/5/2001	9:11 AM	Controlatron	C010547	15	William P. Noel	1472	225	226	0.000146	0.0	4000.0	3496.876	7280	-83762.0	4360605	2.10	2.10
12/5/2001	9:16 AM	Controlatron	C010547	16	William P. Noel	1130	225	226	0.000146	0.0	4000.0	3496.876	7280	-79687.5	3164673	2.15	2.15
12/5/2001	9:18 AM	Controlatron	C010547	17	William P. Noel	1128	225	226	0.000146	0.0	4000.0	3496.876	7280	-79476.6	3157679	2.15	2.15
12/5/2001	9:19 AM	Controlatron	C010547	18	William P. Noel	1091	225	226	0.000146	0.0	4000.0	3496.876	7280	-79476.6	3028295	2.15	2.15
12/5/2001	9:20 AM	Controlatron	C010547	19	William P. Noel	1189	225	226	0.000146	0.0	4000.0	3496.876	7280	-79371.1	3370989	2.15	2.15
12/5/2001	9:21 AM	Controlatron	C010547	20	William P. Noel	1119	225	226	0.000146	0.0	4000.0	3496.876	7280	-79227.3	3126208	2.15	2.15
12/5/2001	9:27 AM	Controlatron	C010547	21	William P. Noel	790	225	226	0.000146	0.0	4000.0	3496.876	7280	-73858.5	1975735	2.20	2.20
12/5/2001	9:28 AM	Controlatron	C010547	22	William P. Noel	744	225	226	0.000146	0.0	4000.0	3496.876	7280	-73618.9	1814879	2.20	2.20
12/5/2001	9:29 AM	Controlatron	C010547	23	William P. Noel	705	225	226	0.000146	0.0	4000.0	3496.876	7280	-73503.8	1678501	2.20	2.20
12/5/2001	9:30 AM	Controlatron	C010547	24	William P. Noel	755	225	226	0.000146	0.0	4000.0	3496.876	7280	-73360.0	1853344	2.20	2.20
12/5/2001	9:32 AM	Controlatron	C010547	25	William P. Noel	743	225	226	0.000146	0.0	4000.0	3496.876	7280	-73561.3	1811382	2.20	2.20
12/5/2001	9:41 AM	Controlatron	C010547	26	William P. Noel	1297	225	226	0.000146	0.0	4000.0	3496.876	7280	-97749.5	3748652	1.85	1.85

Create HTML File | Print Table | Exit

Figure 17. Shot Log Pop-up Panel

- **Configuration\View – View Shot Log**

  Evokes a pop-up panel that displays the shot log for the currently selected device. If the user chooses to, it is possible to create an HTML of this screen, or print the screen to the system printer.

- **Create HTML File**

  Creates an HTML file that is viewable from multiple browsers. Once pressed, a file dialog box will pop-up allowing the user to name the HTML file and the storage location for the file.

- **Print Table**

  Prints the table to the system printer.

- **Exit**

  Exits from this pop-up panel and returns the user to the main application panel.

- **Configuration\View – View Optimum Res. Current Plot**

  The Optimum Reservoir Current Plot is generated based upon a series of shots that were previously taken while the power output of the PFN is 4000 volts. The objective of this plot is to show the difference in the number of neutrons that were generated at different Reservoir Current levels while the output voltage remained a steady 4000 volts thus determining the optimum operational current for the DUT. This graph is designed to help test personnel determine the proper Reservoir Current level when the power output is increased to 5000 volts.

Figure 18. Select Date

- **Configuration\View Processed Data Plot Variables**

  Processed Data Plot Variables permit the user to assign cursor locations on each of the four plots located on the Processed Data Printout (Figure 13, page 1 or page 2). The page is selected by the switch at the top right corner of the window. Data located between the cursors is used to calculate the four variables indicated below each plot (Mean, Max, Min and Range). If the Display Cursor control is set to 'No', the cursors as well as the calculated variables will not appear on the printout.

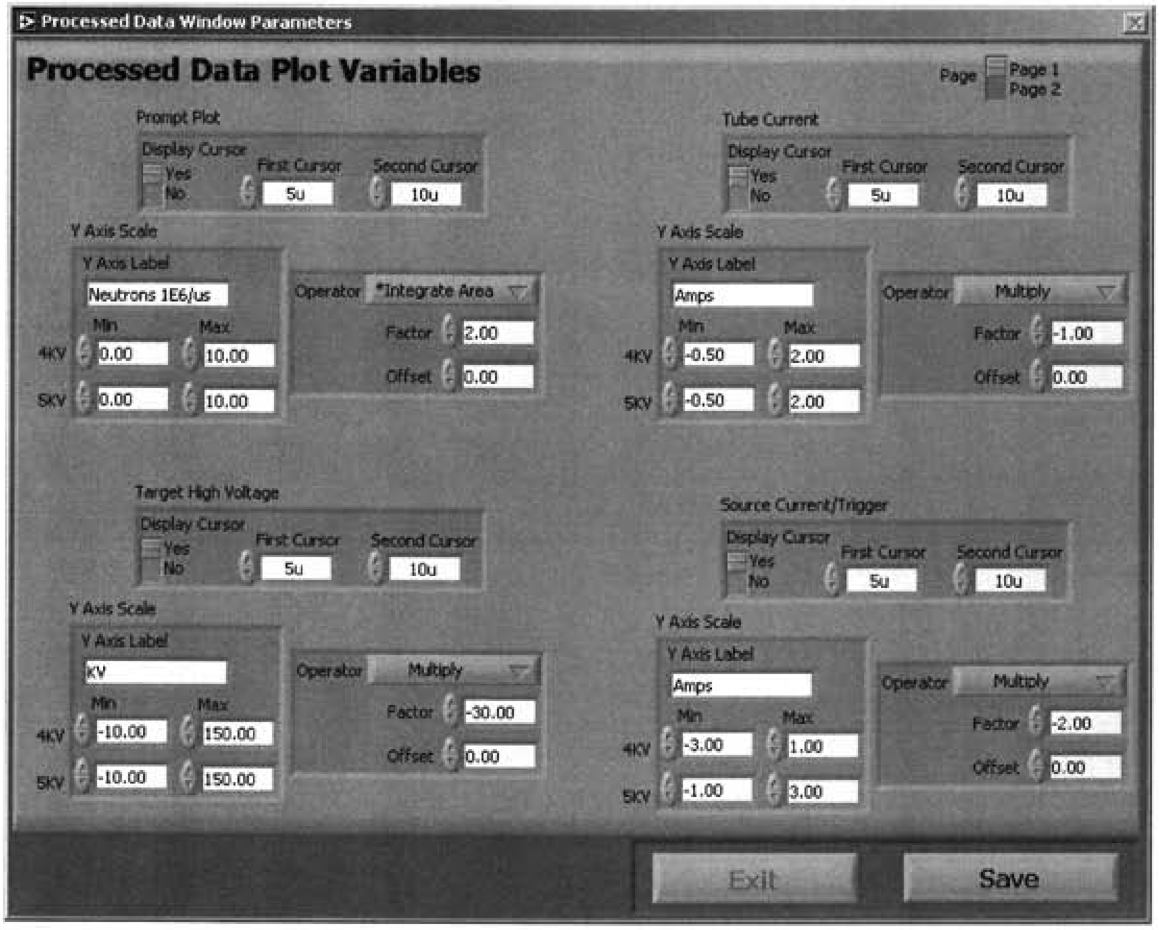


Figure 19. Process Data Plot Variables

The Y-axis scale (used for shot report only) can be set for two Y-axis scales, either the 4KV or the 5 KV. The Y-axis label is the label that will appear in the shot report printout. This is not tied to the scope channels specified on the scope channel setup display. The "Operator selection" will apply a factor and / or an offset to the captured waveform prior to generating the shot report. This feature is good for scaling the waveform to voltage or amps.

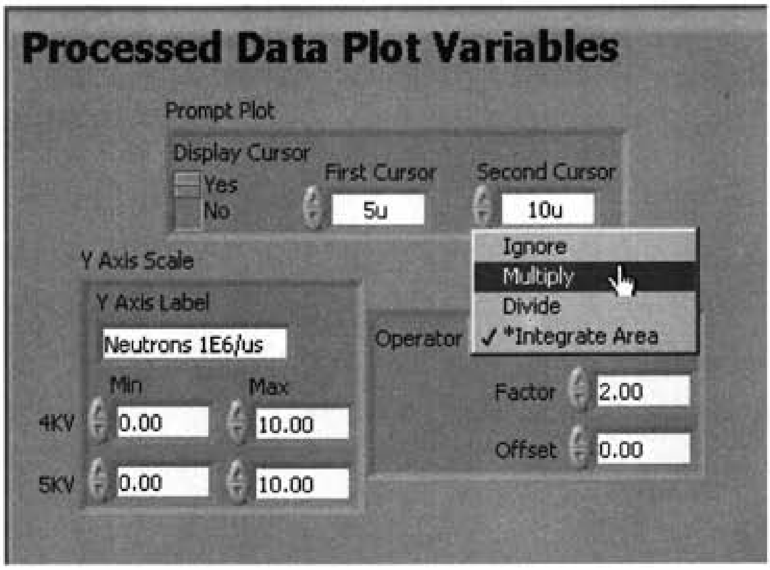


Figure 20. Processed Data Variables – Close up.

The available operators are:

Ignore	Nothing is calculated
Multiply	Waveform is multiplied by the factor specified
Divide	Waveform is divided by the factor specified
* Integrate Area	This setting is used to calculate the neutron rate of scope #1 channel 31, If this is selected, factor and offset are ignored.

- **Configuration\View Configure System Sounds**

The system sounds configuration panel permits the user to select .Wav files that will be played when the assigned action occurs in the software. This panel permits the user to select the absolute path to a desired wave file, and determine if the sound should be active or not. The Preview button located to the right of each file path control permits the user to test the sound file that has been selected.

2.4.1.3 Test Menu

- **Test – Test Mode – Step**

Test Mode relates to how the application will behave when the user has started an automated test sequence. While in the 'Step' mode, the test sequence will pause after each step has been completed. In order to proceed to the next step, the user must press the Start Test button.

Figure 21. Optimum Reservoir Current Plot Pop-up Panel

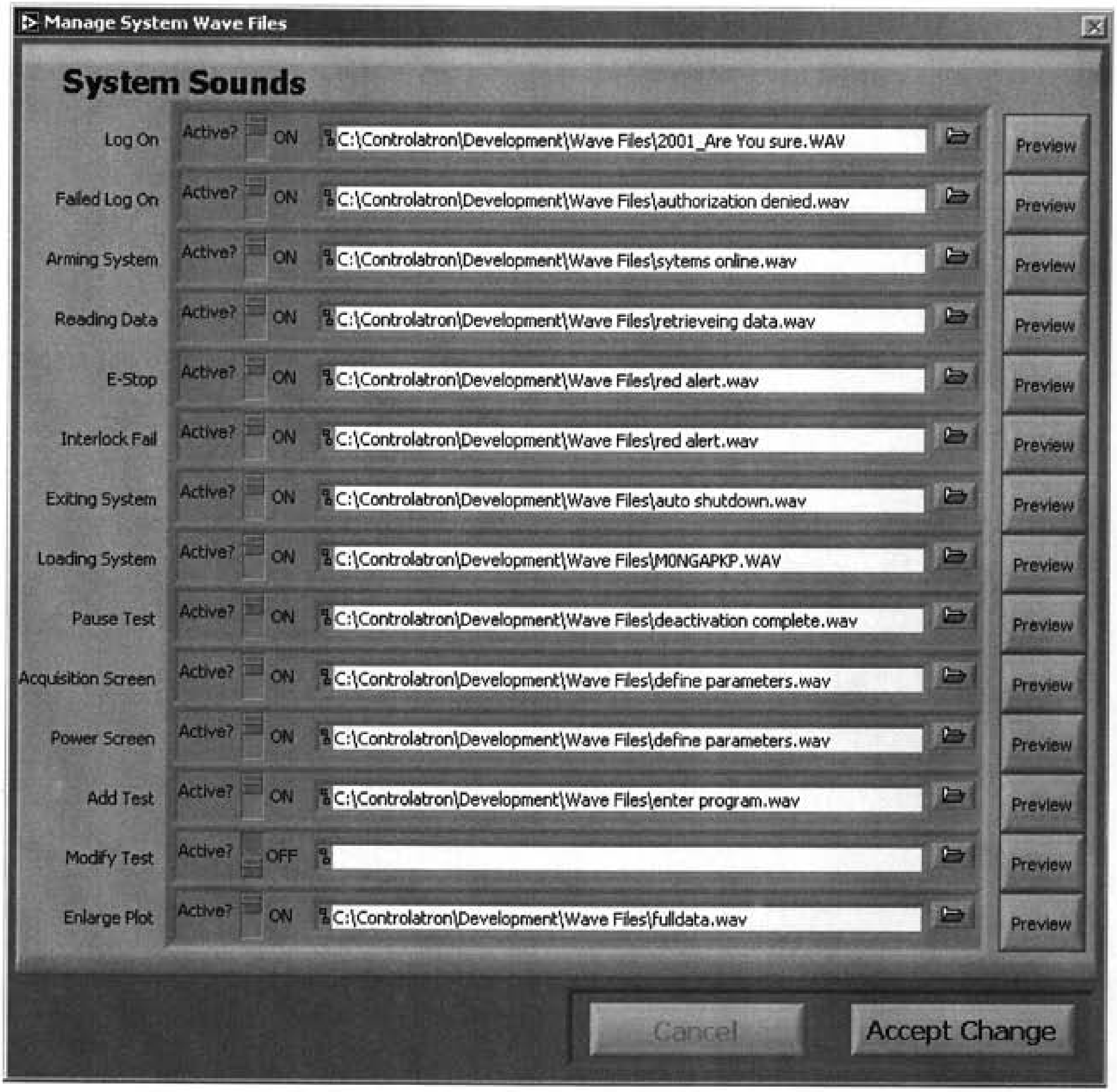


Figure 22. System Sounds Configuration Panel

- **Test – Test Mode – Auto**

  Test Mode relates to how the application will behave when the user has started an automated test sequence. While in the 'Auto' mode, the test sequence will sequentially execute without user intervention until all of the defined 'Test Steps' and 'Tests' have been completed.

- **Test – Modify Test**

Invokes a panel that is used the selected defined 'Test' (Section 2.4.2).

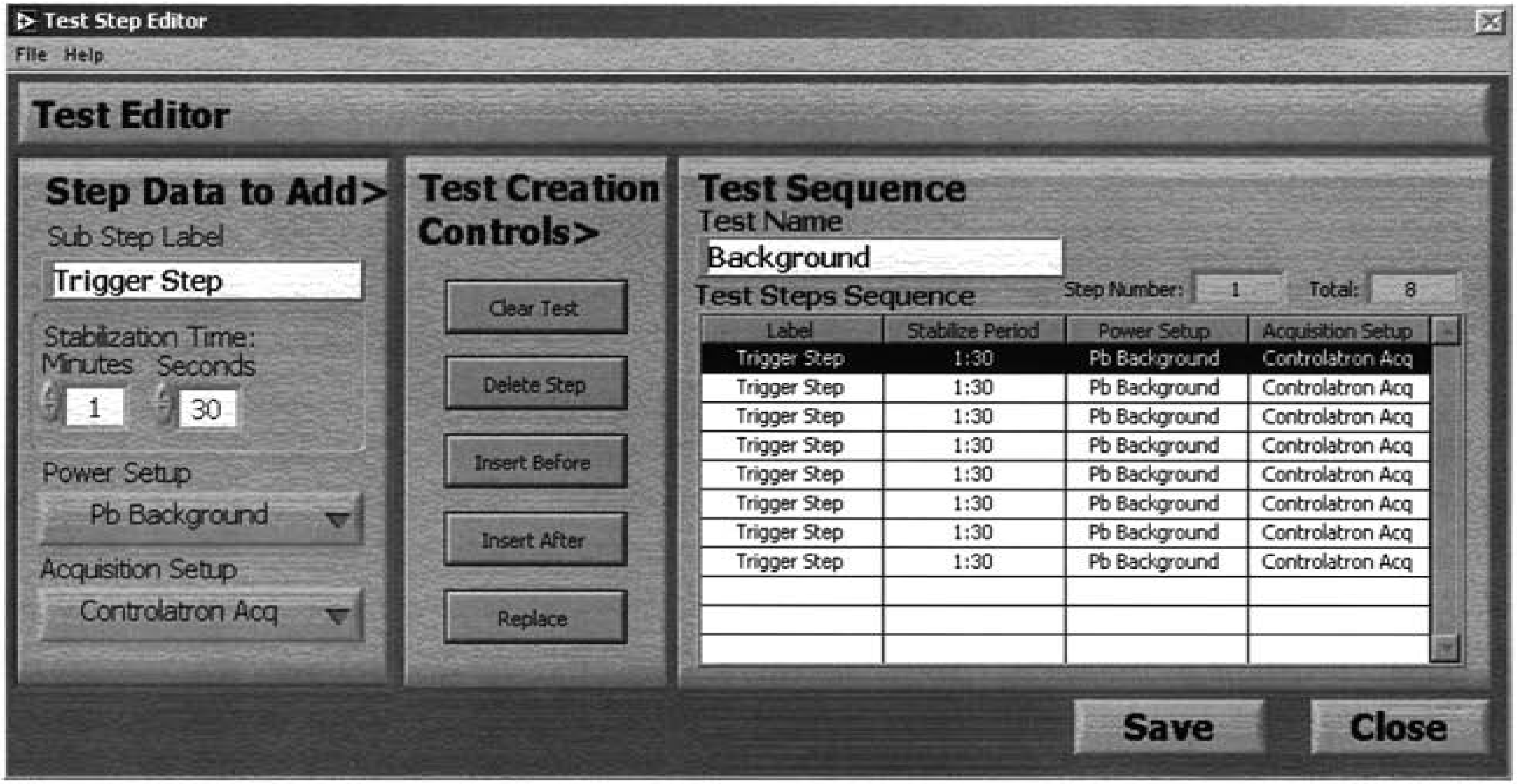


Figure 23. Test Step Editor Pop-up Panel

To manipulate Test Steps Sequence' table. This table will drive the automated test sequence when the test is executed from the main control panel. Variables adjusted within each test include the test step label, the stabilization period for the step (how long to wait between power adjustments and triggering the system), which Power System setup file to use to configure the Power Rack, and finally, which Acquisition System setup file to use for configuring the Acquisition System.or create a test sequence, the user defines individual Steps to be incorporated into the test sequence and then inserts them into the '

Once a test has been defined or altered, the user can 'Close' this panel. If changes to a configuration have been made and not saved the user will be prompted to save changes or ignore changes and return to the Main Panel. If the manipulated test sequence is saved, it will later be selectable by the 'Test Name' defined on this panel.

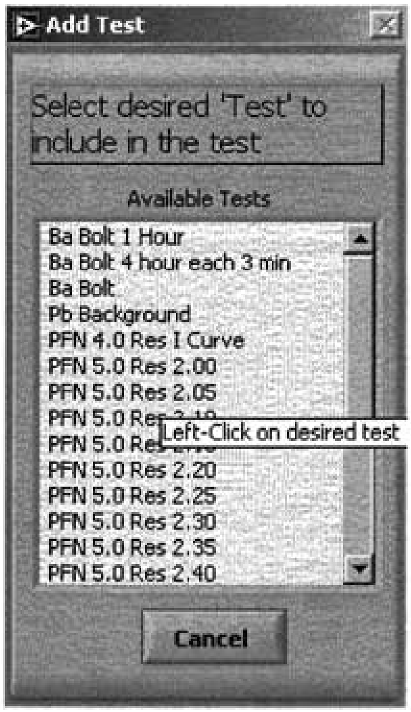


Figure 24. Add Test Pop-up Panel

- **Test – Add Test**

  Invokes the 'Add Test' pop-up panel allowing the user to select a pre-defined test sequence to execute.

  The test names available in this pop-up refer to tests that have been created using the 'Test Editor' previously discussed.

- **Test – Remove Test**

  This selection removes the selected test sequence from the 'Tests' control defined in Section 2.4.2.

- **Test – Create New Test**

  This selection invokes the same panel that was invoked by the '***Test** – Modify Test*' menu item invoked, but when it is invoked using this menu item, the panel will be empty rather than partially filled by the selected test.

2.4.1.4 User Menu

- **User – User Manager**

  The User Manager is only available to users that have been assigned 'Administrative' or 'Power User' privileges. The invoked panel is discussed in Section 2.5.

- **User – User Logon**

  Permits a new user to logon to the system without restarting the Controlatron Test Suite application. The invoked window is discussed in Section 2.2.

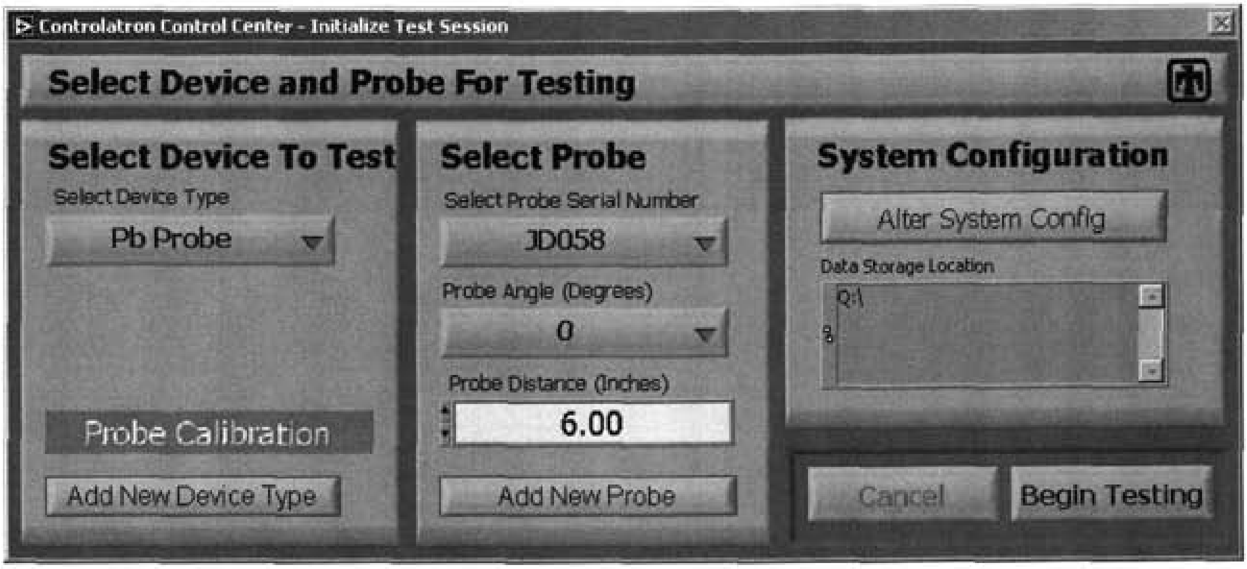


Figure 25. Probe Calibration

2.4.1.5 Calibration Menu

- **Calibration – Pb Probe Calibration**

  Invokes the 'Set Calibration Factors' popup panel. The calibration factors defined on this panel determine calculations that are used to calibrate the acquisition system and how data is processed within the application.

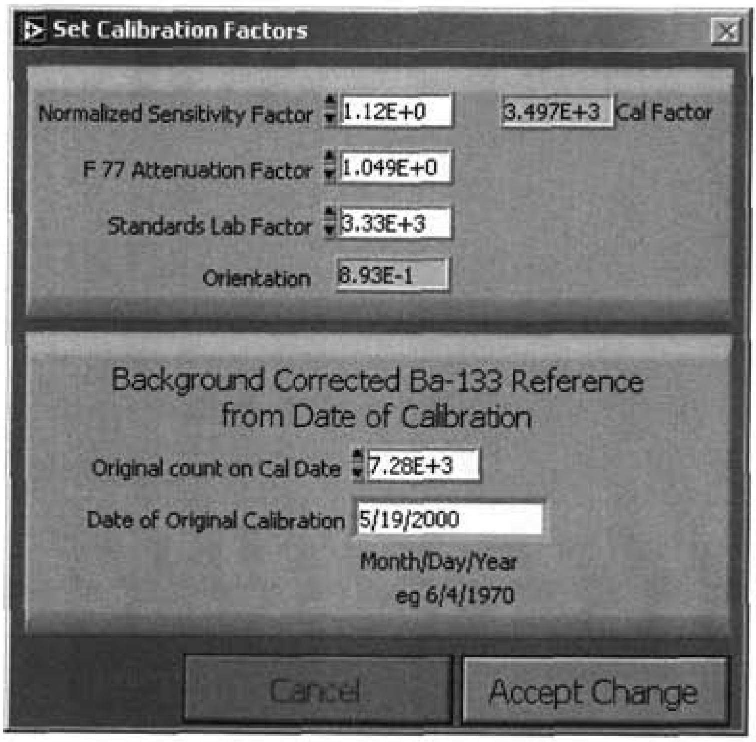


Figure 26. Set Calibration Factors Pop-up Panel

Calibration Factor (Cal Factor) is calculated using the following:

$$
\begin{aligned}
&CalFactor = f \times F \times AF \times OF \\
&where: \\
&\qquad f = Factor = 3.333 \\
&\qquad AF = F77\,AttenuationFactor = 1.049 \\
&\qquad F = NormalizedSensitivityFactor = 1.12 \\
&\qquad OF = OrientationFactor = .893 \\
&\qquad n = CalFactor(ScalarCnt - Bkd)
\end{aligned}
$$

Background corrected BA-133 reference from Date of Calibration

$$BA - \underset{\text{(on Cal Date)}}{133\text{count}} \times e - \frac{\ln 2(\text{time from cal date})}{\text{BA-133 Half Life}} = \text{Expected Pb probe count for day.}$$

NOTE: Time from Cal Date and BA-133 Half Life <u>must</u> be same units, i.e., day, week, year).

<u>BA-133 Half Life</u>
10.5 Years
126 Months
546 Weeks
3825.125 Days

- **Primary Standards |a| provides the following values:**

f = factor

F = normalized sensitivity factor

BA-133 count on Cal Date

2.4.1.6 Help Menu

- **Help – Show Help**

  This selection invokes a roaming 'Help' window. The purpose on the help window is to assist the user with simple descriptions of panel controls and indicators. Information in the help window directly relates to the panel object that the cursor is above. To gain information about a specific object, activate the help window and then move the cursor over the object of interest.

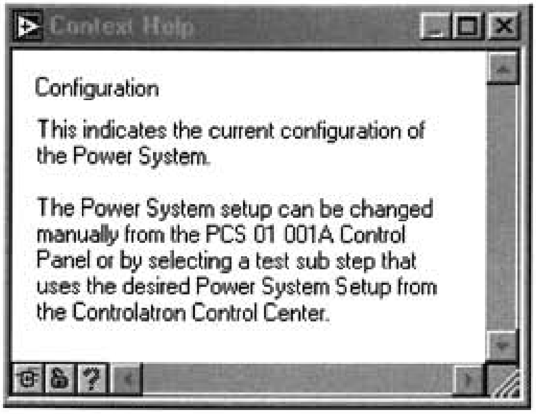


Figure 27. Roaming Dynamic 'Help' Window

- **Help – About**

  The 'About' window offers the user revision information about the Controlatron Test Suite Application and the application developers.

To modify the Revision information, manipulate the ReVIsion.txt file found in the application folder.

2.4.2 Testing Controls

These controls manage all of the automated testing procedures, show the user what step the test is at and display the time until trigger.

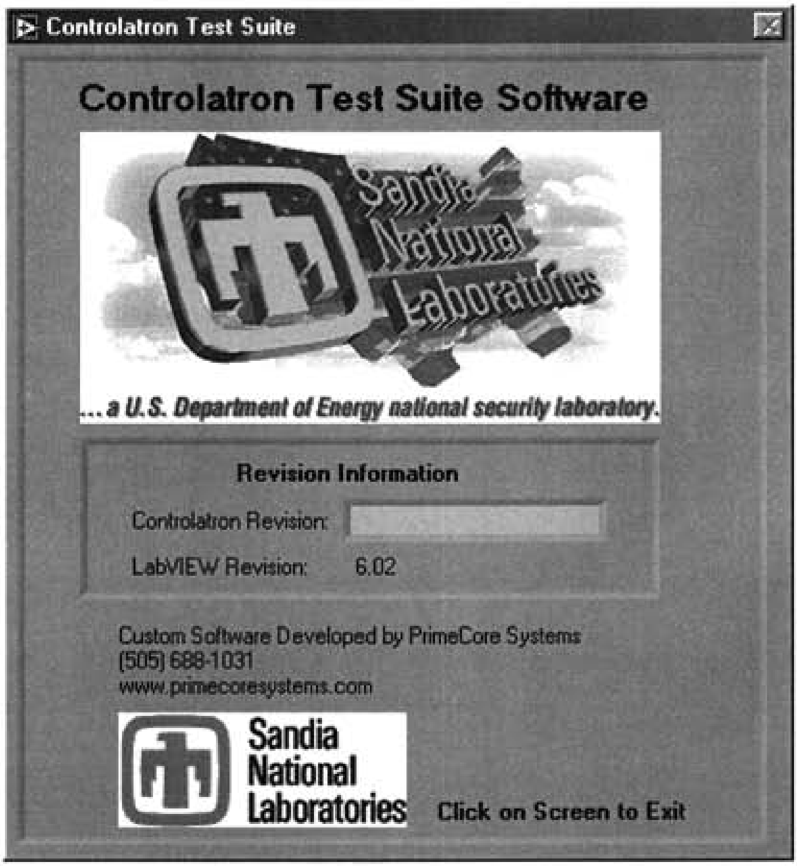


Figure 28. 'About' Pop-up Panel

- **Start Test**

Begins the highlighted 'Test' at the highlighted 'Test Step'. The selected 'Test Step' determines the Stabilization Period, specific Power Setup to employ, and the Acquisition Setup to use. When the 'Test Step' is selected, the Power System and the Acquisition System will immediately re-configure themselves to the proper settings as defined by the selected setups. When the 'Start Test' button is pressed, the 'Time to Trigger' countdown indicator will be reset to the time specified by the selected test step's 'Stabilization Period'. When there is less than 45 seconds remaining prior to a trigger, the Acquisition System will be automatically armed if it is not already in the armed state.

When the 'Time to Trigger' reaches zero (0), the Acquisition System and the Power System settings will be queried and prepared for creating a 'Settings' file that will reflect the setup of both systems prior to the trigger. Once the settings have been acquired, the Acquisition System will be sent the 'Trigger' Command starting the acquisition process.

Note: It is normal for the acquisition period to take several seconds. This delay is caused by the acquisition duration configured for the Stanford Research 430 Discriminator/Counter.

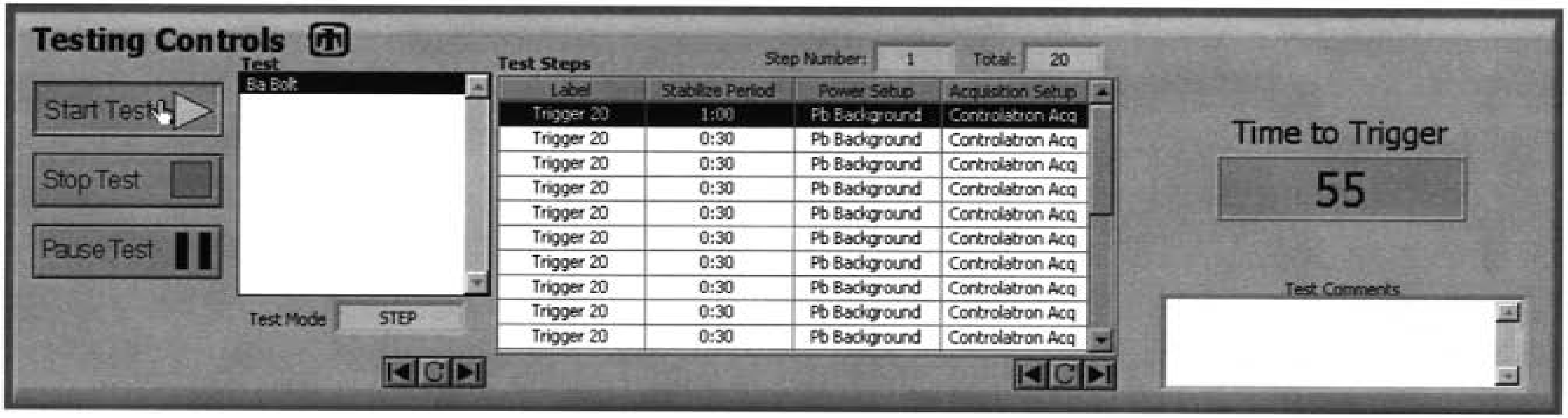


Figure 29. Testing Controls available on the Control Center Panel

When the data has been collected, all dynamic files are created and saved and the automated test procedure is continued. If there is another test to be performed in the test sequence, the next step in the sequence will be started.

Note: The next step will only be dynamically started if the Test Step Mode is set to 'Auto' (Section 2.4.1), other wise the test sequence will be paused until the user manually continues the sequence by pressing 'Start Test'

Stop Test

Stops the current test sequence. If the Acquisition System is armed, a 'Manual Trigger' button that appears in this section can trigger the system. To continue the test sequence, press the 'Start Test' button.

- **Pause Test**

  This pauses the test so that data and system settings can be reviewed prior to the next shot. Press 'Start Test' to take this out of Pause mode.

- **Time to Trigger**

  This indicates how much time is left in the Stabilization period before the system triggers.

- **Test**

  Lists all of the Tests that are currently included in the test sequence. Tests can be considered as a logical group of 'Test Steps' that have a defined order of execution. Multiple 'Tests' can be loaded into the 'Test' list and will be executed in a top-to-bottom order from the selected 'Test' when the 'Start Test' button is pressed. Flexibility is provided to the user by allowing the test sequence to be started at any test point.

- **Test Steps**

  Lists all of the steps included in the test sequence for the selected 'Test'. The collection of 'Test Steps' is the collection of variables that will specify the conditions of each shot taken.

- **Back 1 Step, Repeat Step, Skip 1 Step**

  The Step-Backward, Step-Forward and Repeat-Step buttons are located beneath the Test Steps and Test Displays. They allow the user to skip or repeat a step or test while a test is actively in progress.

- **Test Mode**

  Defines the currently selected step mode that the automated test is executing in. The Test Mode choices are 'Auto' and 'Step' as defined in Section 2.4.1.

- **Step Number**

  Indicates the Test Step number that is currently being executed for the selected Test. This indicator is related to the 'Total' indicator located to the right of it.

- **Total**

  Displays how many steps are included in the currently selected 'Test'

2.4.3 Acquisition System Display

This section offers a quick status view of the current Acquisition System State as well as some functional control relating to the Acquisition System.

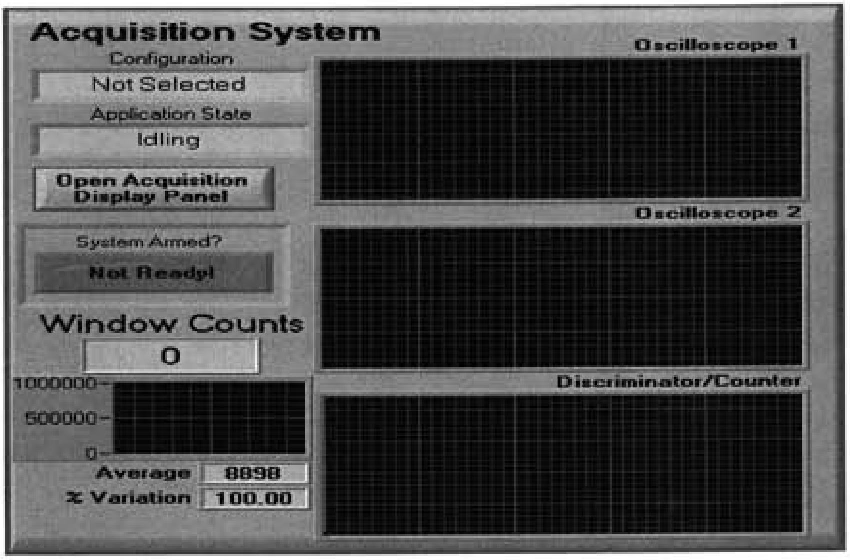


Figure 30. Acquisition Controls available on the Control Center Panel

2.4.3.1 Acquisition Control Section – Controls

- **Open Acquisition Display Panel**

  Pressing this will pop-up the control panel for the acquisition system.

- **System Armed?**

  This button behaves both as a control and an indicator. The user can manually arm the system by pressing this button or pressing the 'System Armed?' button located on the 'Controlatron Acquisition System' Panel.

- **Enlarge Plot**

  There are 'invisible buttons' that cover the plot area of the three visible plots in this section. The intention is that if the user would like to view the data presented in any of the three plots with more detail, he must simply left-click when the cursor is over the plot of interest.

  The enlarged plot offers the user powerful analysis tools as well as a blown-up view of the plot that had been selected for enlargement. Anomalies in the data being plotted can be zoomed in on using the graph tools (left of the plot legend). Traces may be made inactive (not visible) or hidden using the 'Show Plot' controls next to each trace in the plot legend (right of plot legend). To regain the 'full-scale' of the data, the 'Auto-Scale X' and 'Auto-Scale Y' buttons are provided to the left of the graph tools palette.

  Once the data has been scaled by the user, or zoomed in on, the entire panel can be printed to the system printer or an HTML can be created of the panel. The default location for the HTML file is within the selected shot's HTML directory, however, the user will be prompted to name the file and if he desires the directory where the file is saved can be altered.

  The indicators at the top of the panel provide shot, device, and test engineer information. Information displayed in these controls is provided by the system and cannot be altered on this panel.

2.4.3.2 Acquisition Control Section - Indicators

- **Configuration**

  Displays the name of the configuration that the Acquisition System is currently in. Creating Acquisition Configuration Setups are discussed in Section 2.9.

- **Application State**

  Displays the current state of the Acquisition System. The Acquisition System may be 'Arming System', 'Triggering', 'Reading Data', or in an 'Idling' state. This indicator is useful if the system seems to be acting 'sluggish' or 'hung-up'.

  Note: The first time the acquisition system is armed, the arming process may take up to 45 seconds to arm. This duration is normal and is instrument driven.

- **Oscilloscope 1 and 2 Graph**

  These display the raw scope data from the previous shot. Click on these to view an enlarged version of the plot.

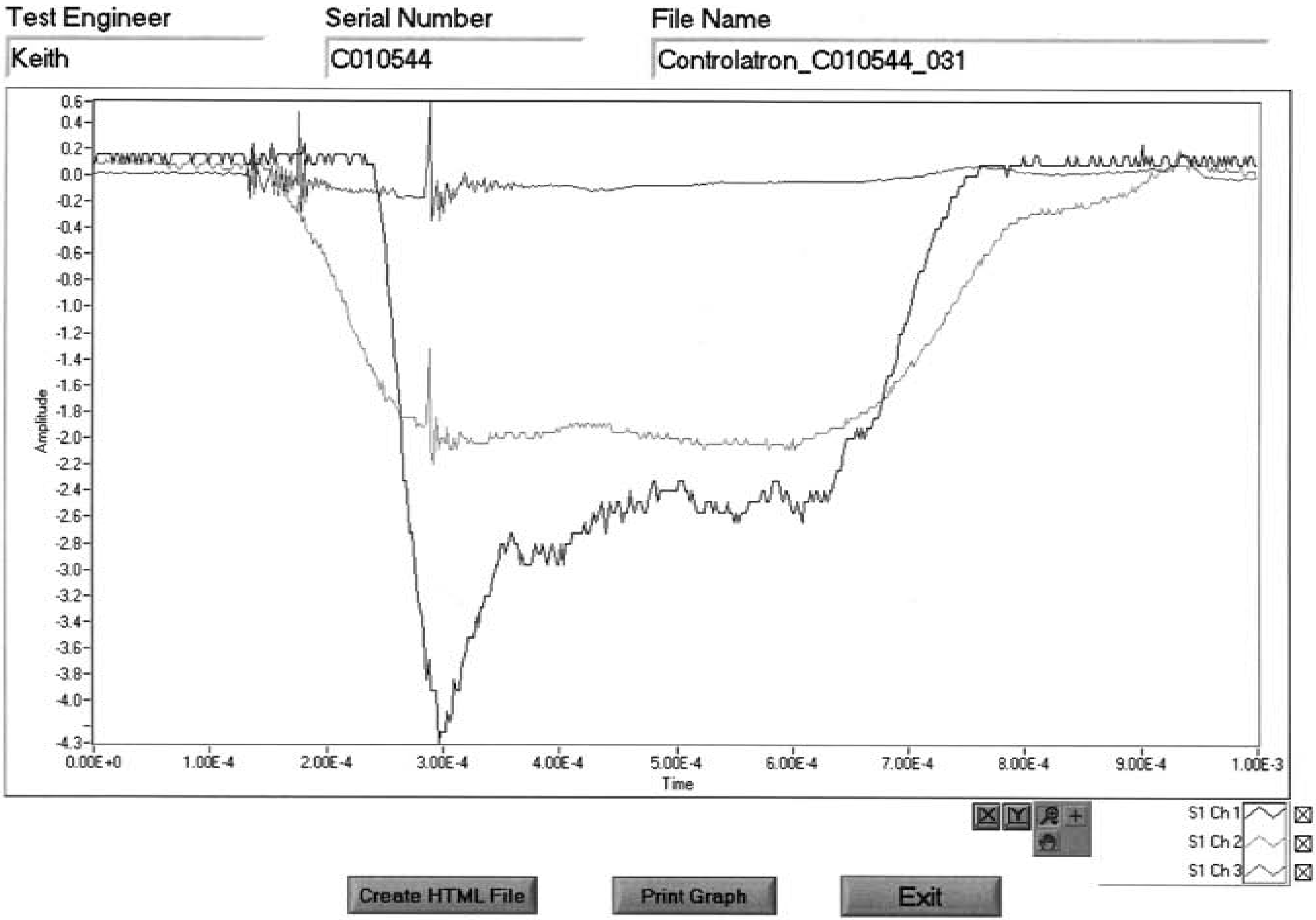


Figure 31. 'Enlarge Plot' Pop-up Panel

- **Discriminator\Counter Graph**

  Displays the counts over time for the previous shot. The vertical red lines illustrate the 'Active Window'. The 'Active Window' is the user-defined portion of the data that is used to calculate the 'Window Counts'.

- **Window Counts**

  Displays the total counts within the configured time window.

- **Window Count Plot**

  Small plot that accumulates the reported 'Window Counts' for all of the shots executed during the current test session. This plot dynamically grows and auto-scales to the values that are accumulated. The 'Average' and '% Variation' indicators below this plot are directly associated with it.

- **Average**

  Calculated average of 'Window Counts' collected during the current test session. This is the average value of values displayed on the 'Window Count Plot'.

- **% Variation**

  Calculated total variation of 'Window Counts' collected during the current test session relative to the 'Average' Window Counts.

- **System Armed**

  Displays if the Instruments are armed and ready to trigger.

2.4.4 Power System

Offers a quick status view of the current Power System State and the Interlock System as well as offering some functional control relating to the Power System (Emergency-Stop/E-stop).

2.4.4.1 Power System Section – Controls

- **Open Power Display Panel**

  Pressing this will pop open the control panel for the Power system.

- **Open Interlock Display Panel**

  Pressing this will pop open the control panel for the Interlock System.

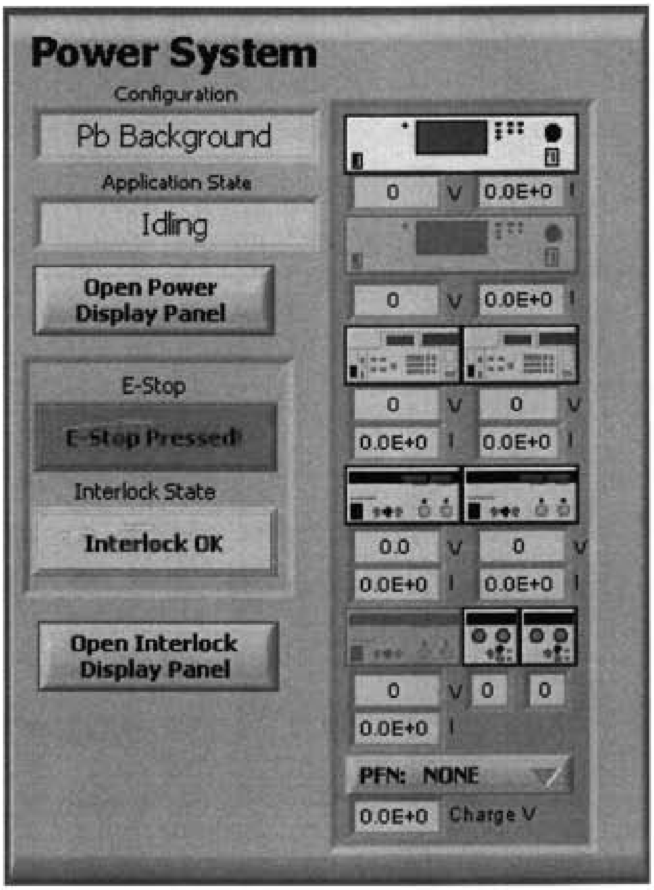


Figure 32. Power System Section of the Control Center Panel

- **Emergency Stop (E-Stop)**

  This button behaves both as a control and an indicator. If this button is pressed, the E-Stop will be asserted or de-asserted for the Power System.

  Note: The E-Stop can only be de-asserted if all interlocks are satisfied and a Power System Setup is selected.

2.4.4.2 Power System Section – Indicators

- **Configuration**

  Displays the name of the configuration that the Power System is currently in. Creating Power Configuration Setups are discussed in Section 2.10.

- **Application State**

  Displays the current state of the Power System. This indicator is useful if the system seems to be acting 'sluggish' or 'hung-up'.

  Note: When a power supply is turned on, there is a built in 10 (ten) second delay to permit the instrumentation to process internal initialization prior to programmatically attempting to initialize the device.

- **E-Stop**

  Displays the state of the E-Stop. This must be green to power on the supplies. The E-Stop cannot be de-asserted if any interlock is unsatisfied.

- **Interlock State**

  Displays the state of the interlocks. The Power system will shut down if any interlock fails.

- **Power Rack Display**

  Shows the voltage and current for each of the power supplies. Power supplies that are grayed out are turned off.

  Note: These indicators display the voltage and current levels being reported by the active power supplies, NOT the desired levels defined by the user on the Power System control panel.

2.4.5 Device Information

This section indicates device specific information as specified by the user on the 'Initialize Test Session' panel (Section 2.3). This section also permits the user to make comments about the previous shot that was taken.

2.4.5.1 Device Information Section – Controls

- **Common Comments**

  Pull down the list to dynamically add a common comment to the shot comments for the previous shot.

- **Shot Comments**

  Enter user comments here that will get stored along with the shot data. These get saved to a <serial number_shot number>_Comments.txt file in the shot number directory. This control is operated like a simple text editor. The user can manipulate the shot log to describe any events or anomalies that occurred during the previous shot.

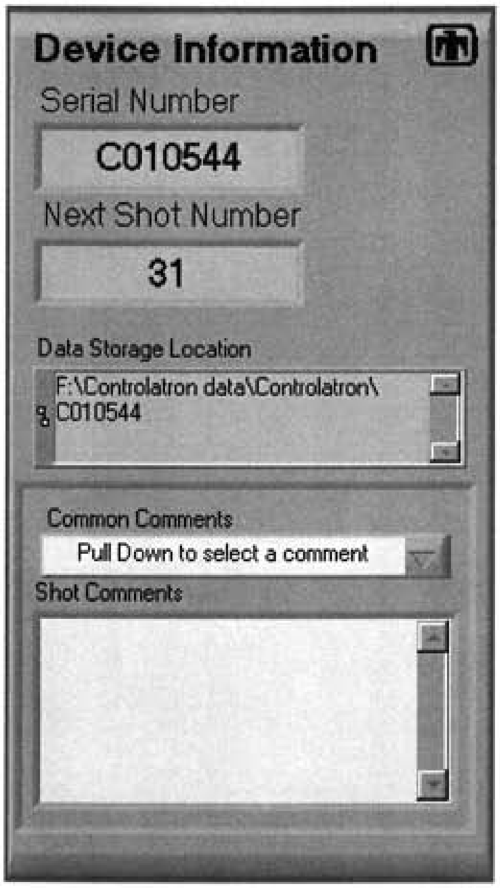


Figure 33. Device Information Section of the Control Center Panel

2.4.5.2 Device Information Section – Indicators

- **Serial Number**

  Displays the current serial number of the device under test. The user enters serial number information while the test session is being initialized (Section 2.3).

- **Next Shot Number**

  Displays what the next shot number will be when the countdown and trigger is complete.

- **Data Storage Location**

  This shows the top-level directory where data is stored.

2.5 Administrative Options

The following section pertains to sections of the Controlatron Test Suite that are only available to user's that have been assigned a 'Administrator' User Profile. These sections are safeguarded in this manner to protect the stability and continuity of this system.

2.5.1 User Manager

The User Manager is used to add users and administrators to the Controlatron Test Suite. If personnel attempt to logon to the system without proper authorization, access will not be granted and the system will be shut down.

Note: This safeguard only protects the system from dynamic testing, the Controlatron Test Suite application cannot safeguard from manual use of this test system.

Logging on as an administrator, accounts can be added or modified by selecting User>User Manager from the pull down menu on the Controlatron Control Center menu bar.

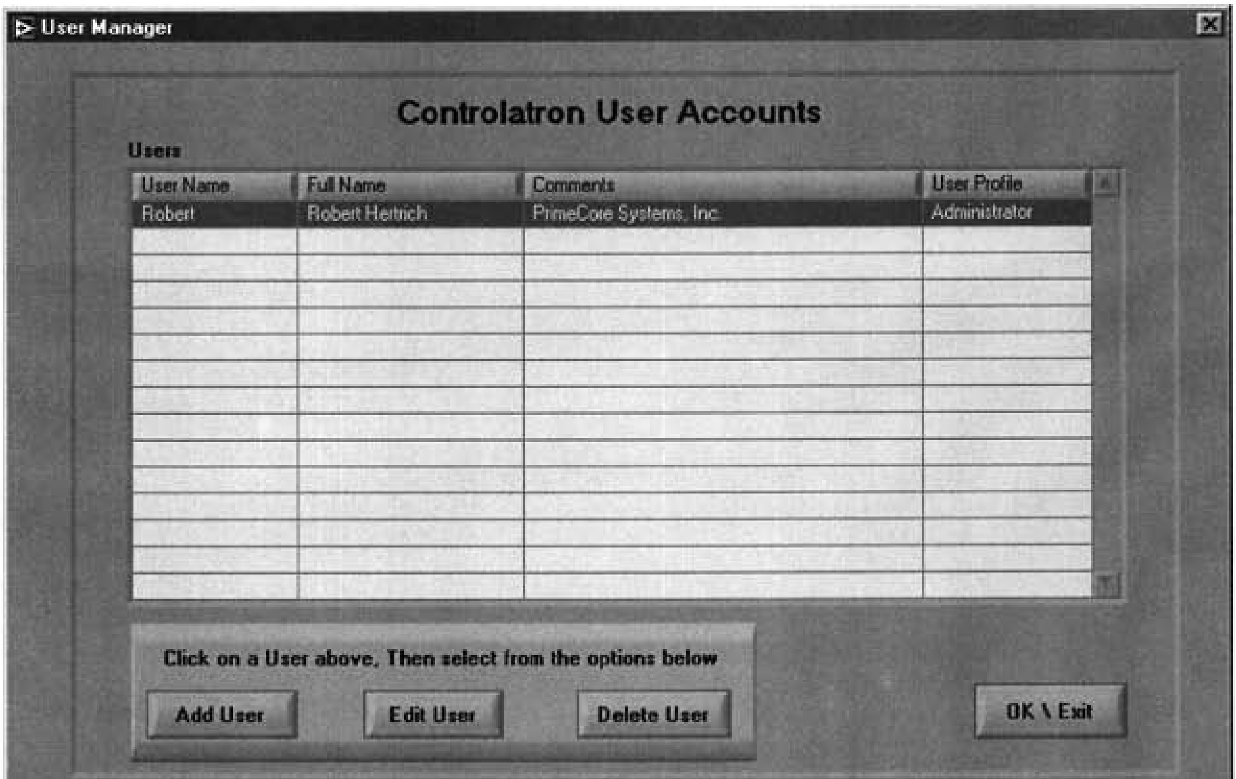


Figure 34. User Manager Pop-up Panel

2.5.1.1 User Manager – Controls

- Users

  The 'Users' table behaves as both a control and an indicator. As a control, the user can select an account to modify or delete by left clicking on the account of interest. This table shows account user names, full names of users, comments about users and the level of access granted to each account.

- **Add User/Edit User**

  Pops-up the 'Add New User'/'Edit User Profile' Panel. From this pop-up panel, modifications to a user account or a new user account can be made. All available inputs on the User Profile Information Pop-up panel must be filled in.

  Note: 'Password' and 'Confirm Password' must be identical including letter capitalization.

- **Delete User**

  When utilized, the currently selected user on the 'Users' table is deleted from the system.

- **OK\EXIT**

  Exits the user from this panel and returns to the Controlatron Control Center Panel. **Note:** When this panel is exited, the user will be prompted to replace the existing file User-File.PWD. This dialog will be evoked if changes have been made on the User Manager or not. If you do not wish to retain potential changes that were made, 'Cancel' this action. If you wish to retain all changes that have been made on the User Manager, press the 'Replace' button.

2.6 Selecting a Test to Perform

The following section illustrates the steps necessary to load a 'Test' into the test sequence. Prior to selecting a test to perform, the user must have available valid Tests constructed using the 'Test Editor' (Section 2.8).

2.6.1 Add Test Pop-up Panel

The 'Add Test' Pop-up Panel is invoked from the Controlatron Control Center Test>Add Test menu item.

The 'Add Test' panel lists all of the created tests that are available for execution. To select a 'Test' to include in the test sequence, simply left click on the desired test. Once a 'Test' is selected, this panel will close and the user will be returned to the Controlatron Control Center and

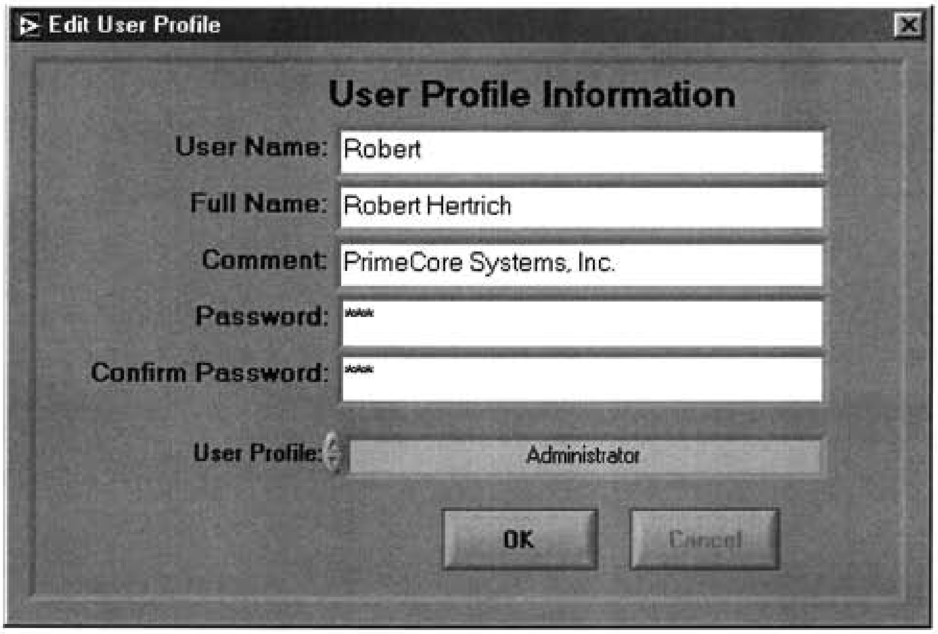


Figure 35. Edit User/Add New User Pop-up Panel

Figure 36. Replace UserFile.PWD Dialog Pop-up

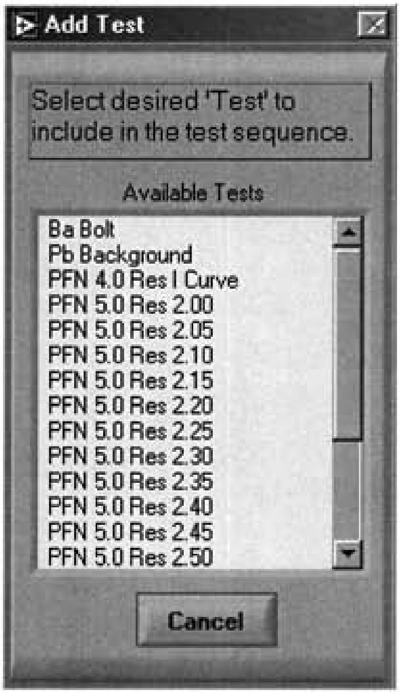


Figure 37. Add Test Pop-up Panel

the selected test will be added to the test sequence. If the 'Cancel' button is pressed, this panel will be exited and the test sequence will be left unaffected.

2.6.2 Test Sequence

This is a cutout view of the 'Testing Controls' portion of the Controlatron Control Center panel (Section 2.4.2) that relate specifically to the Test Sequence and the actions evoked by adding a test (Section 2.6.1).

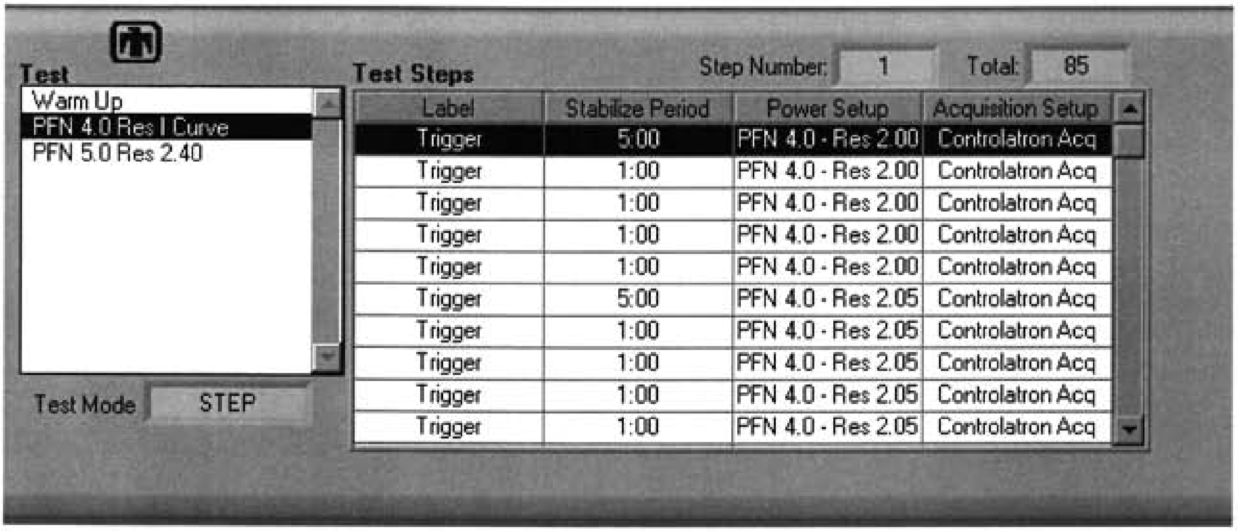


Figure 38. Test Sequence Indicators

The test sequence is controlled by menu items available under Test on the Controlatron Control Center pull down menu. From this menu, the user can add a Test to the test sequence, modify the selected Test, remove the selected Test from the test sequence, or create a new Test.

2.7 Running a Test

The following section describes the method of executing a test sequence.

2.7.1 Verifying Power System Status

Prior to executing a test, verify the state of the Interlocks and the E-Stop button. Note the type of test that has been loaded into the test sequence (Section 2.6). If the 'Test' to be performed requires the Power System to be energized, ensure that all interlocks are satisfied and the 'E-Stop' is deactivated (green).

Figure 38 shows the Power System portion of the Controlatron Control Center Panel when the E-Stop is not asserted and all interlocks are satisfied. The state of these systems can also be ascertained from the Power System Panel and the Interlock System Panel.

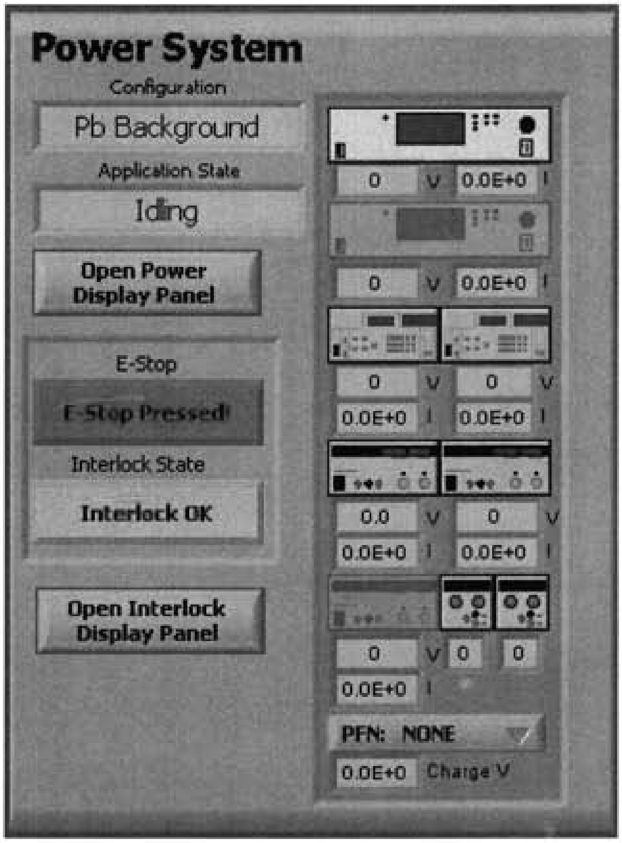


Figure 39. E-Stop and Interlock Verification

Note: Power is not required to trigger the acquisition system and collect data. Use 'Comments' to describe faulty data if the Power System was not in the desired state.

2.7.2 Starting the Test

The controls for running, pausing and stopping a test are located on the left of the screen. Whatever button is pressed on the left is the running state of the test.

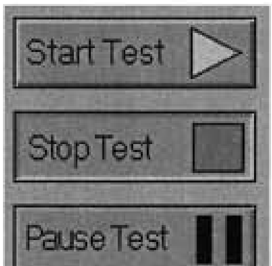


Figure 40. Test State Controls

- **Start Test**

  This begins the highlighted test and test step (Figure 39). If the selected 'Test' or 'Test Step' is not the first selection available, the application will assume that the user did not wish to perform the prior steps available in the test sequence and they will not be executed as a part of the currently active test sequence.

  This functionality promotes flexibility in the testing structure; however, the user must take care not to inadvertently omit critical test steps from the test sequence by not starting at the top of the list.

- **Stop Test**

  Stops the current test sequence. If the Acquisition System is armed, a 'Manual Trigger' button that appears in this section can trigger the system. To continue the test sequence, press the 'Start Test' button.

- **Pause Test**

  This pauses the test so that data and system settings can be reviewed prior to the next shot. Press 'Start Test' to take this out of Pause mode.

- **System Arming**

  During a test, the Acquisition System will automatically be armed if the 'Time to Trigger' countdown is less than 45 seconds and the system is not in an armed state. The acquisition system is configured during the arming state to the specified configuration setup.

- **System Triggering**

  This message covers the 'Time to Trigger' indicator when the Acquisition System has issued a trigger.

  Note: There is a time delay between triggering the system and acquiring data. This delay is hardware driven and is related to acquisition duration specified for the Stanford Research 430 Discriminator/Counter as well as Device communication and file writing functions within the application.

Figure 41. System Triggering Notification

2.7.3 Validating Returned Data

After the system has triggered and data has been read, verify the data validity to ensure that it seems to be within reasonable tolerances.

As discussed in Section 2.4.3, the Acquisition System portion of the Control Center offers a quick view of the data. Check the Oscilloscope graphs to make sure all desired traces were captured. If a scope channel was not made active on the Controlatron Acquisition Panel, data for that channel will not be returned. The Discriminator\Counter Graph should show the counts over time for the system. If data anomalies seem apparent in any of the plots, left click to evoke the 'Enlarge Plot Panel' (Section 2.4.3.1).

The Window Counts displays the total counts within the configured time window. At the bottom left hand corner is a graph showing the window counts of all the shots of the test over time, the average Window Count for this test session, and the % Variation of the collected Window Counts. This is useful for watching trends as the Controlatron is repeatedly fired.

There are several potential reasons for erroneous data to be returned. Items to be examined include; Interlock State, E-Stop State, Delay Generator Setup, Scope Configuration, Instrument power states and general misfires within the DUT itself. The test suite offers many system status indicators as well as video equipment to aid in resolving testing issues. The key to note is that if acquired data seems skewed, pause or stop the test procedure until reasons for erroneous data are resolved.

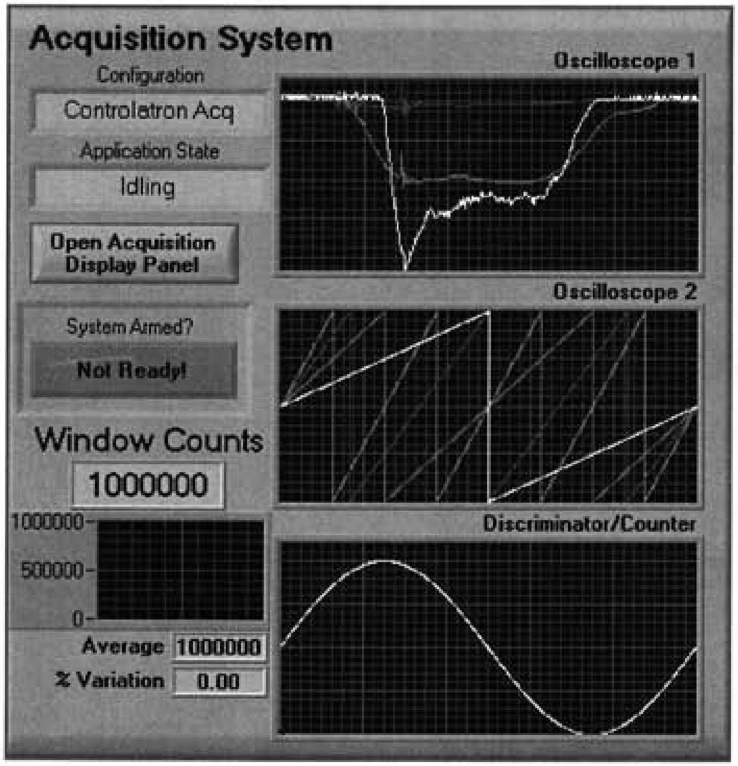


Figure 42. Validating Data

2.8 Creating a Test

This section elaborates the discussion of the Test Step Editor that was touched upon in Section 2.4.1.3. The specific topics of this section include the arrangement of controls and indicators on this panel and how to use them to create custom test sequences or modify existing test sequences.

The Test Step Editor is divided into 3 blocks for different data entry and display functions:

1. Step Data to Add
2. Test Creation Controls
3. Test Sequence

2.8.1 Step Data to Add

This information is used to define a Test Step within the Test.

- **Sub-Step Label**

  This defines the text that will appear in the 'Label' column of the 'Test Sequence' table. This label should be used to briefly describe the functionality of the Sub-Step. An Example is the label 'Warm-up period' found in the first step of the test sequence illustrated in Figure 42. This label tells the end user what to expect from this step, at least it strongly suggests the expected outcome of this step.

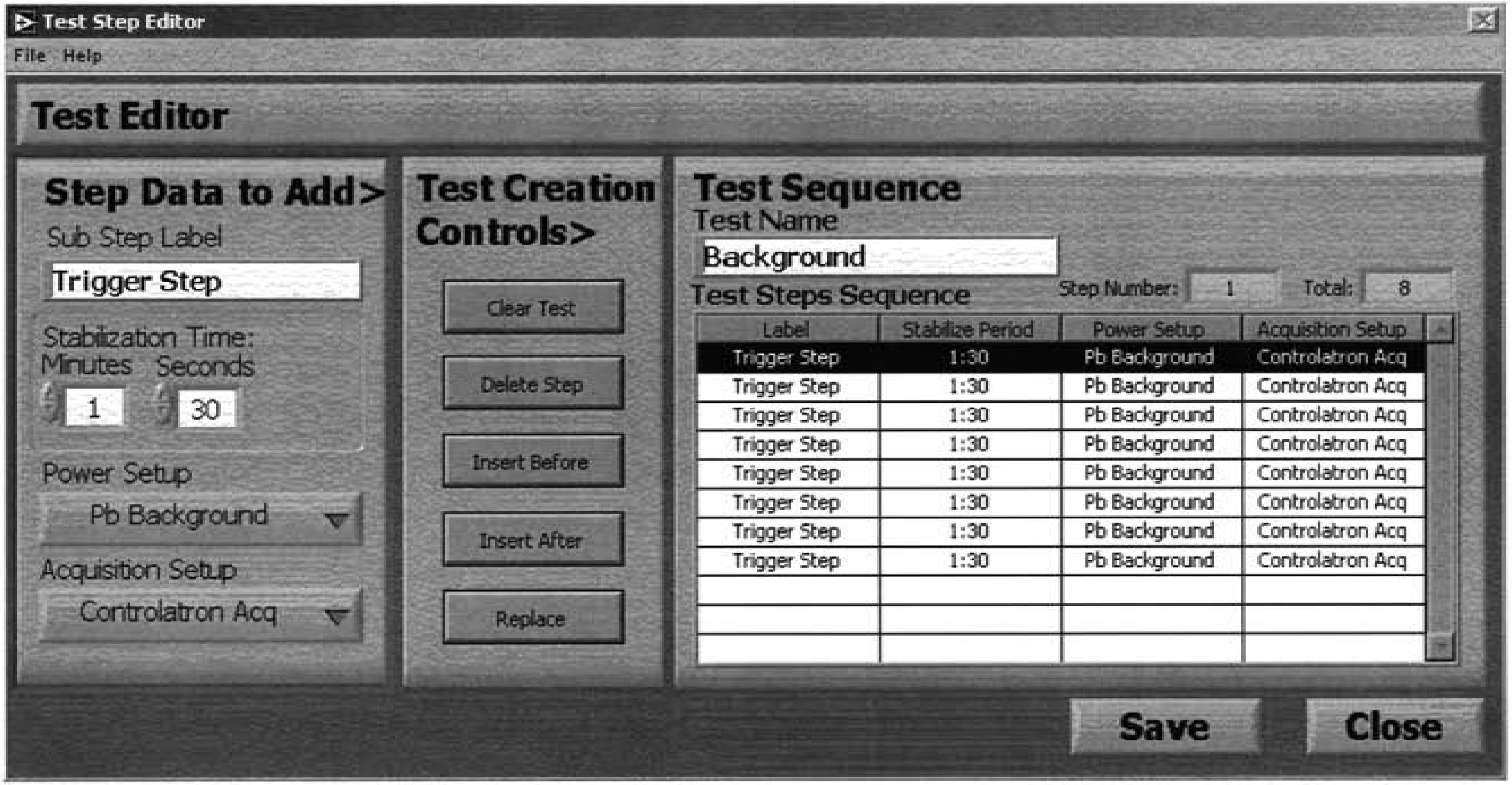


Figure 43. Test Step Editor

- **Stabilization Period**

  Collectively, the minutes and seconds controls are used to define the amount of time the system will be permitted to idle between setting the Power System and Acquisition System and triggering the shot.

  Note: The Acquisition System requires at least 45 seconds to arm the system the first time it is armed. After the system has been armed and triggered at least once, the minimum stabilization period can be reduced to approximately 15 seconds between shots.

- **Power Setup**

  This pull down menu contains all of the currently available Power System Setups (see Section 2.10). Select the Power Setup from the list that will be used when the test step that is being configured is called within a test sequence.

- **Acquisition Setup**

  This pull down menu contains all of the currently available Acquisition System setups (see Section 2.9). Select the Acquisition System Setup from the list that will be used when the test step that is being configured is called within a test sequence.

2.8.2 Test Creation Controls

The Test Creation Controls take actions to add, delete, or replace a test step to the test sequence, or to clear all the test steps from the test sequence.

- **Clear Test**

  Removes all the test steps from the Test Steps Sequence.

- **Delete Step**

  Removes the highlighted test step from the Test Steps Sequence.

- **Insert Before**

  Takes the Step Data To Add information and inserts it as a test step into the Test Steps Sequence right before the highlighted test.

- **Insert After**

  Takes the Step Data To Add information and inserts it as a test step into the Test Steps Sequence right after the highlighted test.

- **Replace**

  Takes the Step Data To Add information and replaces the highlighted test.

2.8.3 Test Sequence

The 'Test Steps Sequence' table and the test name illustrate what the test will look like when it is opened on the Controlatron Control Center Panel.

- **Test Name**

  Defines the label that will be used to reference this test sequence. The name given to the test sequence should briefly describe the type of test that is generated by the associated test sequence. An example of logical naming would be 'Background Average'. This name would strongly suggest that the associated test sequence would take several shots without power asserted to determine the background activities present near the test object.

- **Test Steps Sequence**

  This table summarizes the Test data that was added to create each step in the sequence. This is exactly the order and the nomenclature that will appear on the Control Center Panel when this test is loaded.

- **Save**

  Saves the Test Sequence by the defined Test Name. Pressing this button does not exit the user from this panel.

- **Close**

  Exits the user from this panel and returns to the Control Center Panel.

2.9 Configuring the Control Acquisition System

The Control Acquisition System controls the data acquisition for the Controlatron system. It controls a power switch for the acquisition rack, a VCR, a Video Switch, 2 Tektronics Scopes, 2 Stanford Delay generators, and a Stanford Research Counter\Timer. This screen is used to configure the instruments, save configurations, and to monitor the system when a test is running.

The display is divided into 4 blocks:

1. Selected Configuration

   This is a pull down control for selecting a previously created setup. When you select a setup, setup information is loaded into all the Instrument Configuration displays.

2. Arm State

   Displays whether the system is armed and ready to trigger or not.

3. Acquisition System

   This is a visual representation of the instruments included in the Acquisition System. Click on the instrument of interest within this section to configure its settings below.

4. Instrument Configuration

   This displays detailed setup, arming and configuration data for the selected instrument. The multiple displays available here are further described below.

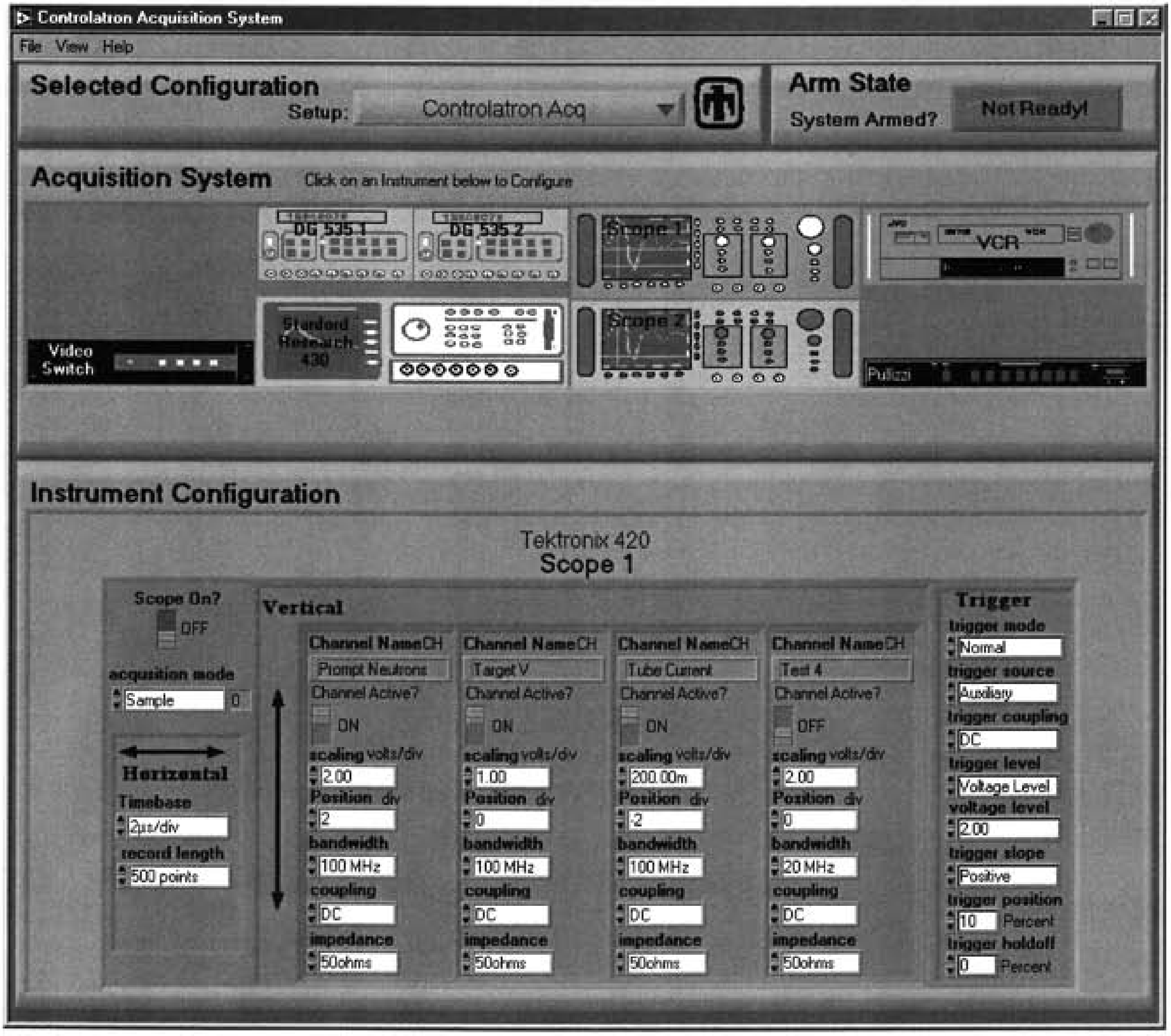


Figure 44. Controlatron Acquisition System Panel

2.9.1 Instrument Configuration Panels

This section offers greater detail about using the Controlatron Acquisition system to configure all of the devices included in the Acquisition System.

2.9.1.1 Tektronix Series 420 Oscilloscopes

The acquisition system contains two scopes that can be configured independently of one another. Select one of the scopes to configure and then utilize the following controls to setup the scope:

The variety of settings permits very flexible acquisition configurations. The notable sections of this setup panel include

- **Scope On**

  This determines if this scope will be armed and triggered during the acquisition cycle. If the scope in general is not turned on, it will be ignored by the system during the acquisition cycle and no data files will be written regarding this scope.

- **Horizontal**

  The horizontal settings are global meaning the Timebase and the record length will be the same for all acquisition channels.

- **Vertical**

  The Vertical settings refer to individual channels. The scaling, position, bandwidth, coupling and impedance can be set individually and independently of the other channels.

- **Trigger**

  Specifies the Trigger parameters for the scope. These parameters include the trigger mode, channel, coupling, voltage level, trigger slope, trigger position, and trigger hold off. Use caution in setting up the trigger channel. Make sure that the selected channel is cabled up and will receive a signal that at least causes the scope to trigger.

- **Channel Name**

  Individual names can be typed into the Channel Name control for each channel. The name specified in this control will be associated with the data that is also received on this channel. Use caution in cabling signals to the scope to ensure that signal routing and channel names match.

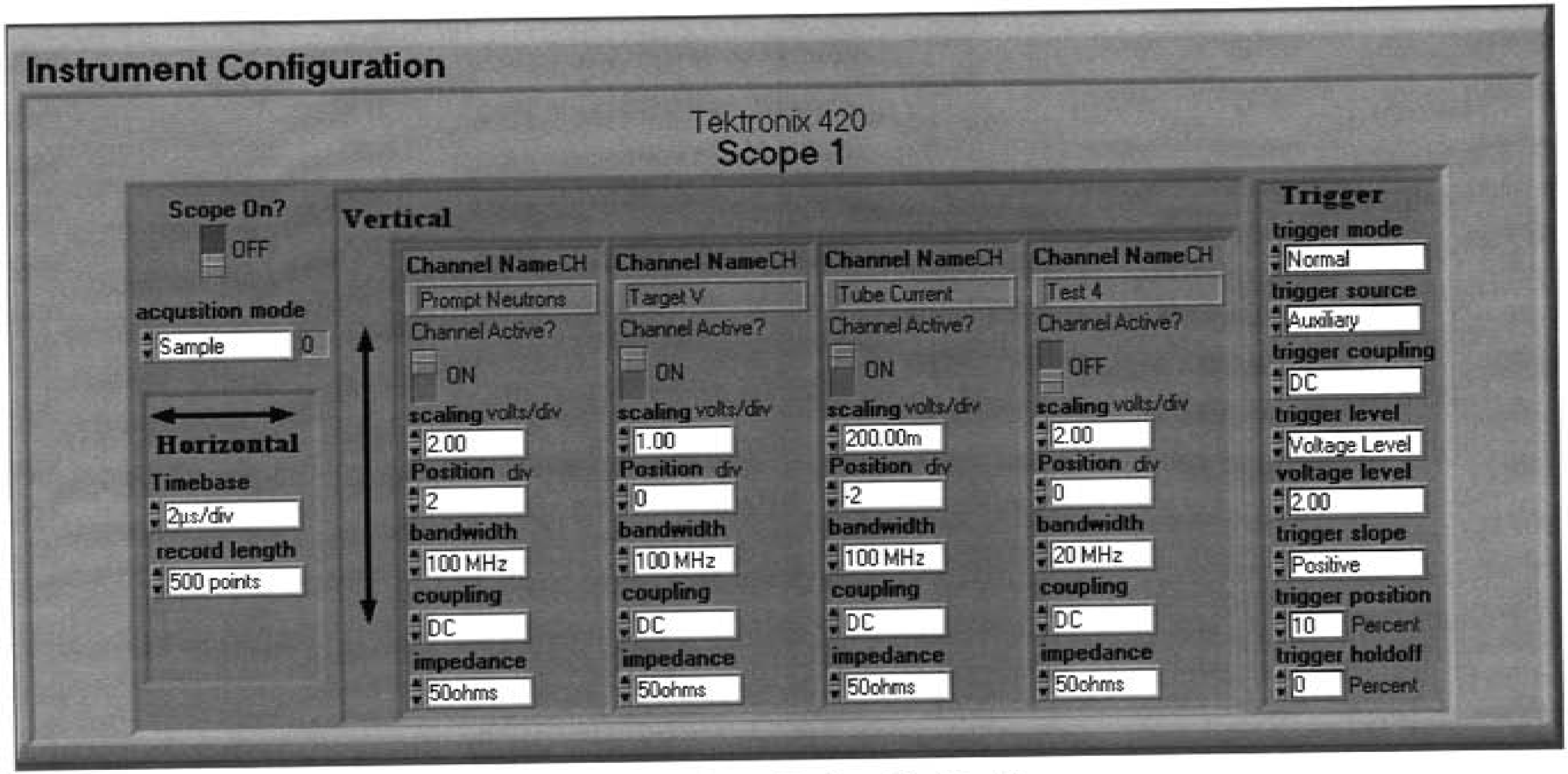


Figure 45. Scope Configuration Panel

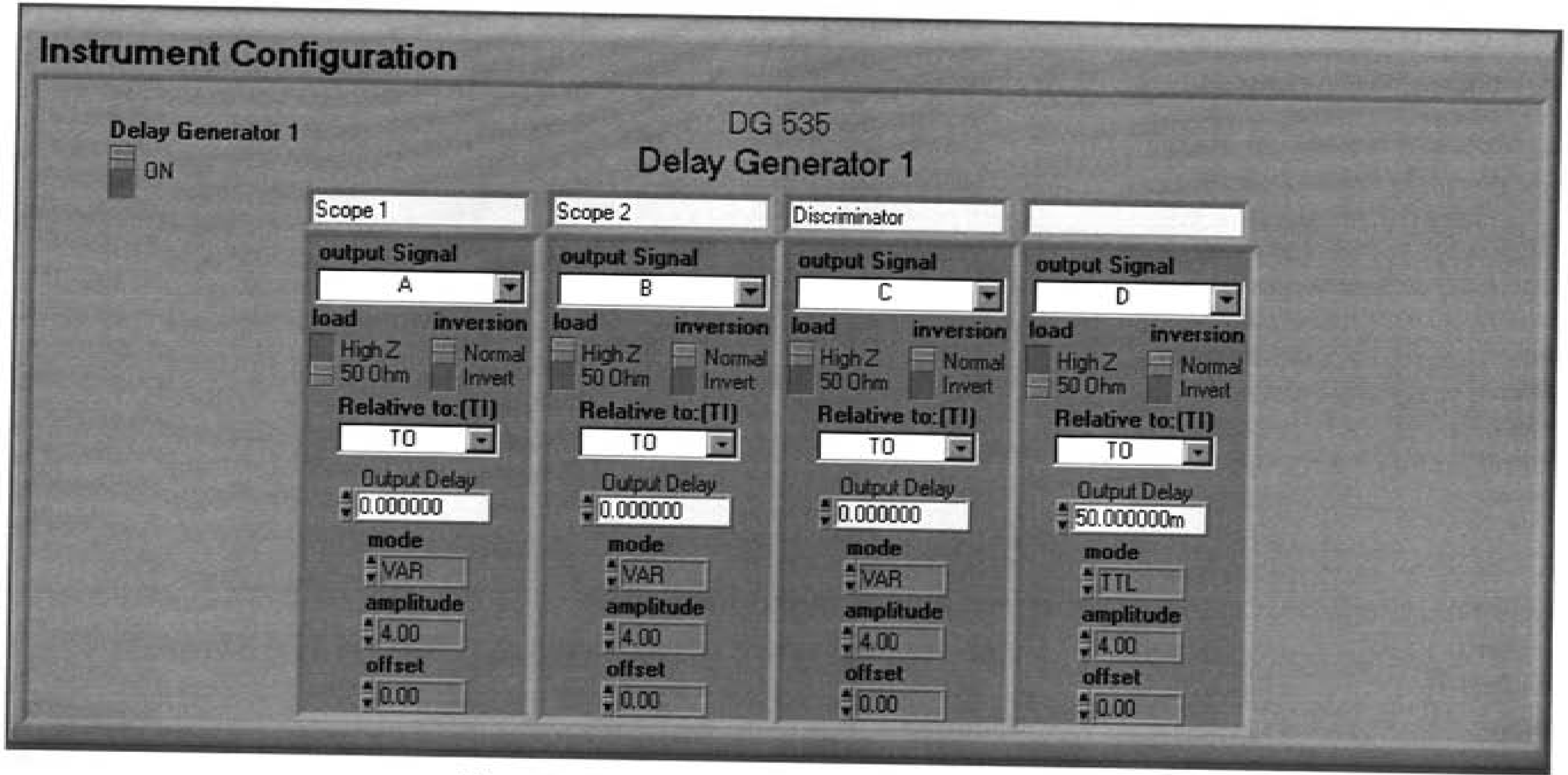


Figure 46. Digital Delay Generator Configuration Panel

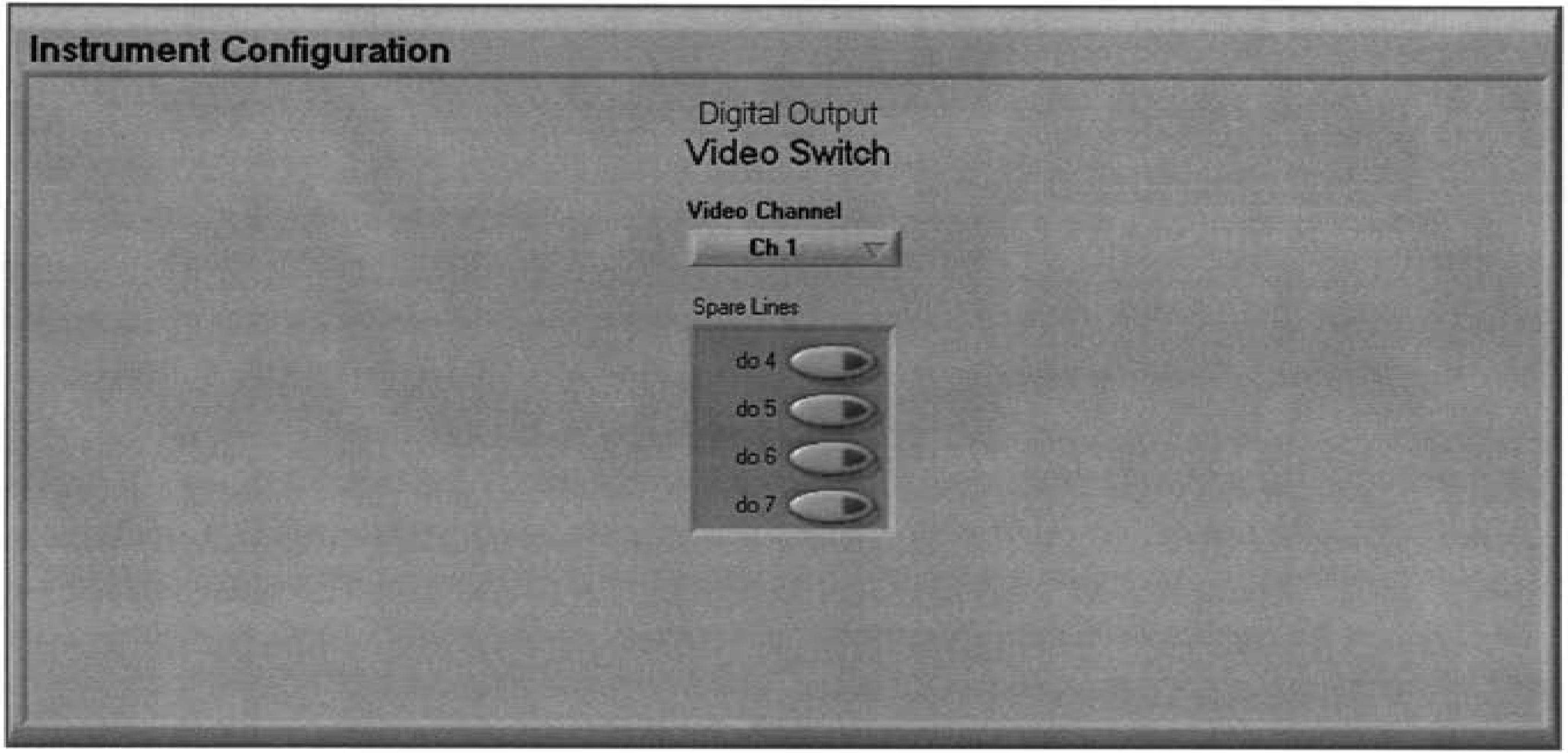


Figure 47. Video Switch Configuration Panel

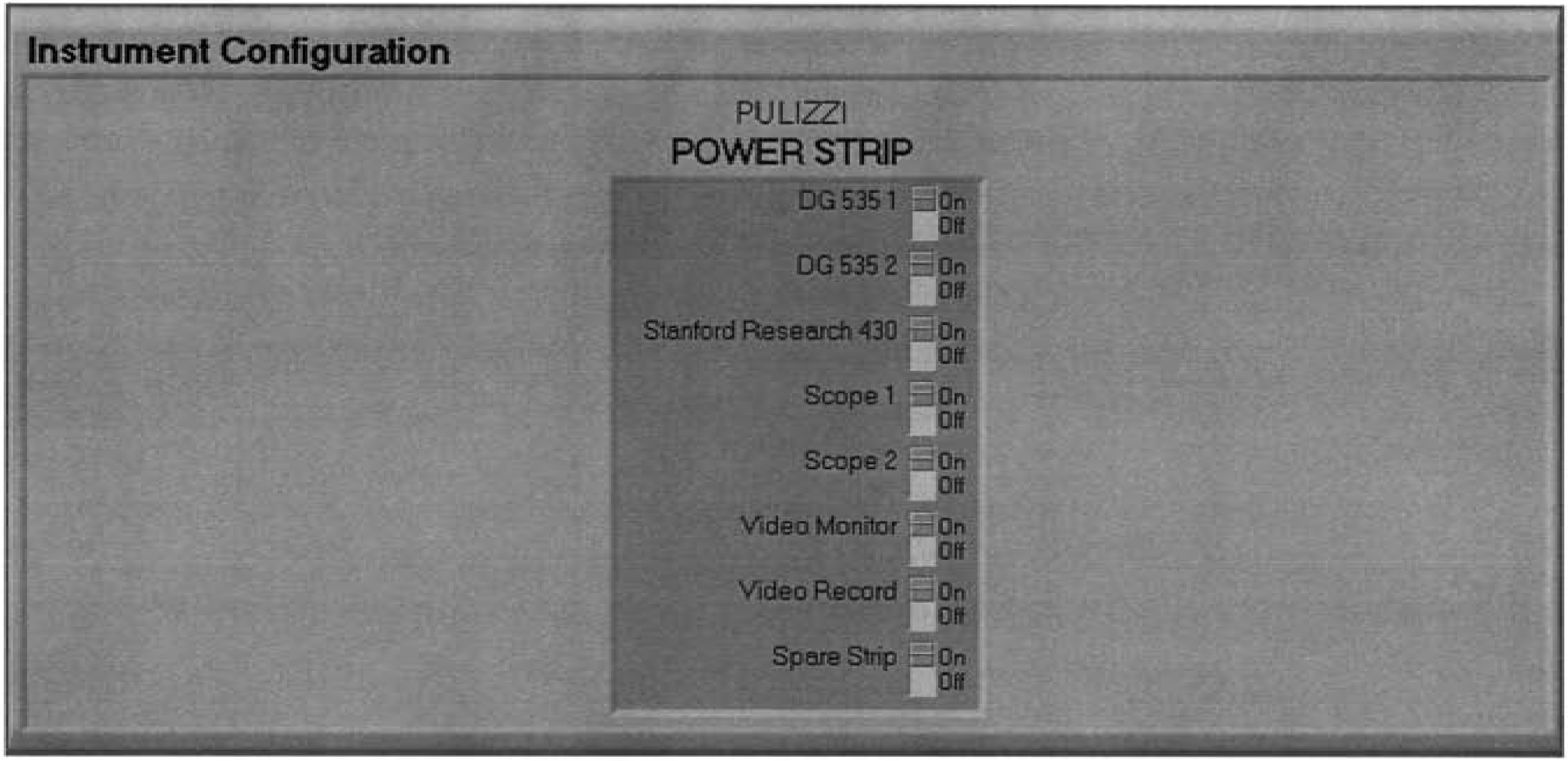


Figure 48. Pulizzi Power Strip Configuration Panel

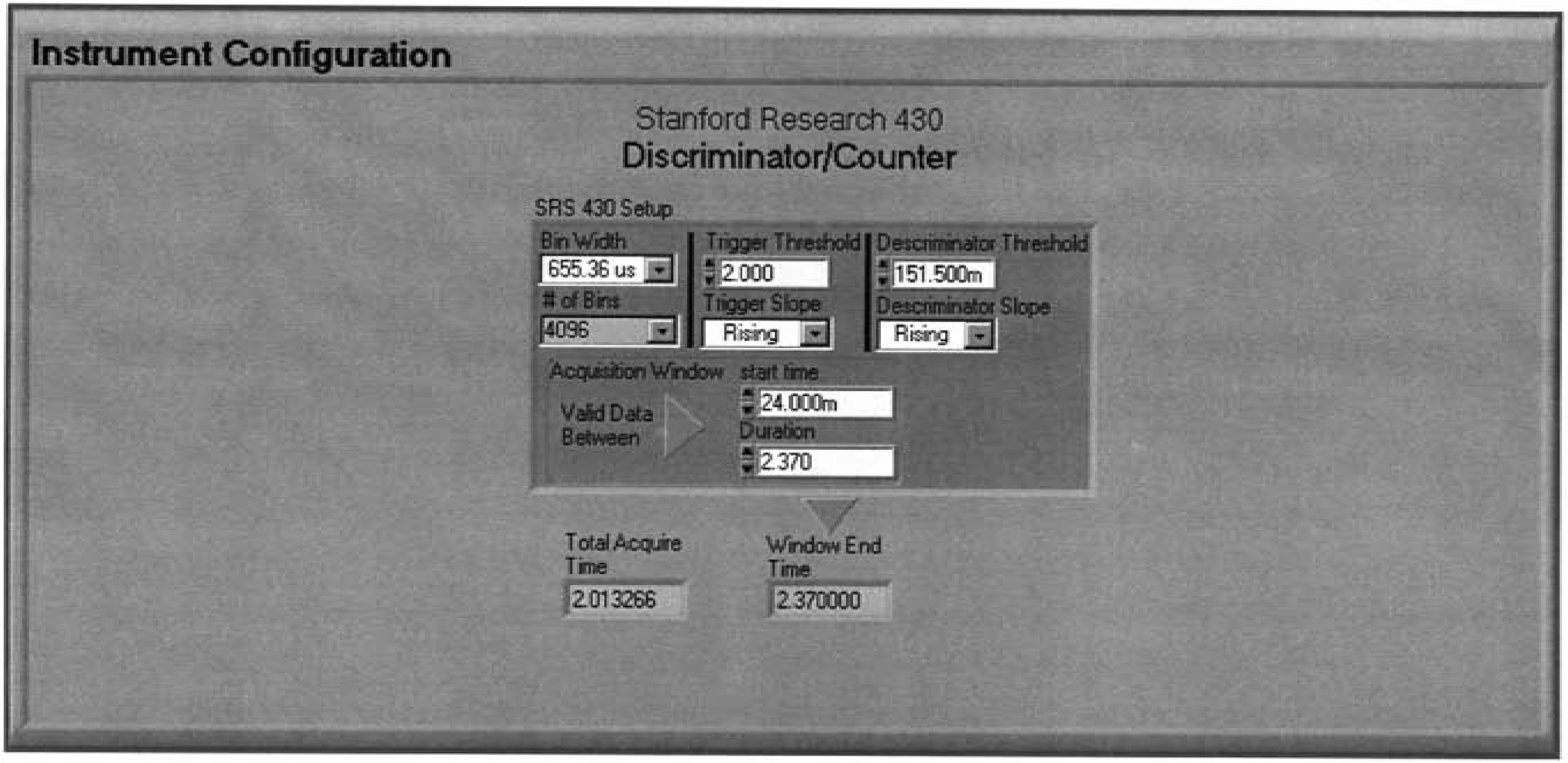


Figure 49. Discriminator/Counter Configuration Panel

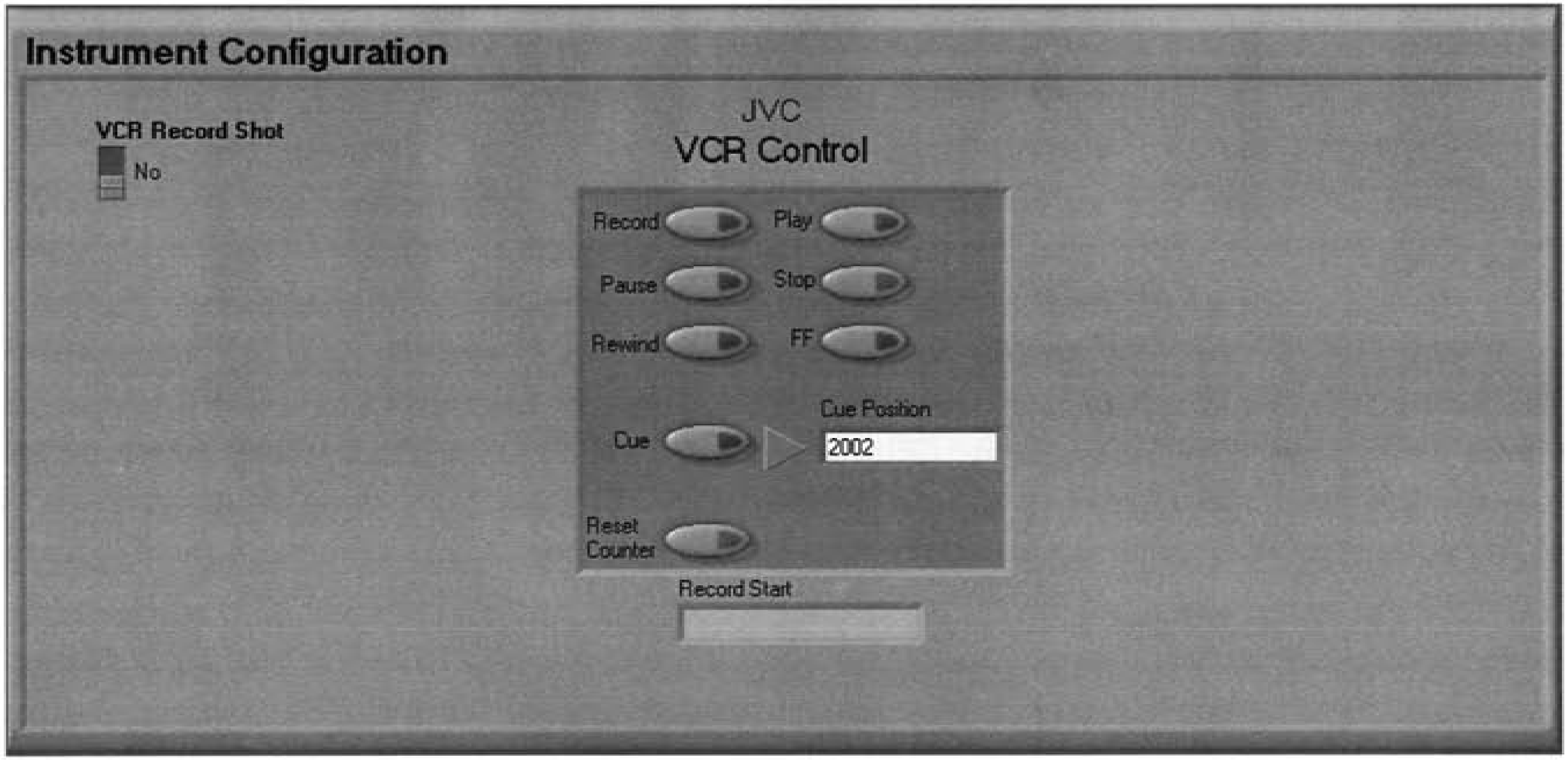


Figure 50. VCR Configuration Panel

- **Channel Active**

  Individual channels can be toggled on and off. If the individual channel is turned off, or if the scope as a whole is not turned on, data will not be collected. This is particularly important for scope one as it houses the default data channels.

2.9.1.2 Delay Generator Configuration

Two Stanford Research DG535 delay generators provide timing signals. These are configured using the display below.

Each channel on the Digital Delay Generator can be independently configured. The labels associated to each channel are used to help clarify where the signal is routed. Use the pull down menu to select which output signal will be described by the settings for each channel.

Note: If the Delay generator is turned off it will not create a pulse when the system is triggered. If it is on, all four channels will pulse at the time of the trigger.

Triggering the generators:

The first DG535 is triggered by software and the second (on the right) is triggered externally. This allows both generators to be triggered simultaneously in the software. Connect the external trigger of the 2^{nd} generator to the T0 signal output from the first delay generator.

2.9.1.3 Video Switch

This selects which video channel is selected in the configuration. This controls the 4-channel video switch. The spare lines are not yet used in the system but are for future expansion.

2.9.1.4 Pulizzi Power Strip Configuration

This allows the user to select what instruments are powered on in a configuration.

Note: It is not vital that unused components of the Acquisition System be powered down during an acquisition cycle. If an instrument has been powered off and then dynamically turned on, allow significant time for the instrument to complete its self-initialization phase (the scopes require about 35 seconds of warm-up time prior to being able to communicate with them). It is the user's responsibility to ensure that the components of the acquisition system are fully initialized prior to attempting to arm the system.

2.9.1.5 SR430 Discriminator\Counter Configuration

This configures the acquisition settings for the SR 430 counter\discriminator. Select the Bin width and number of bins to define how long the instrument will acquire data and with what horizontal resolution. The combination of these 2 fields produce a Total Acquire Time indicated below. Define the Acquisition window that is desired to actually count in using the start time and duration controls. The window end time is calculated and displayed in the Window End Time display. The Trigger Threshold and slope set up the trigger input which is normally TTL rising coming from the DG535 delay generators. The discriminator threshold and slope control the count signal level and slope.

2.9.1.6 VCR Control

This allows you to control the VCR's major functions and to enable or disable it in the configuration.

The Cue Position must be of the following format (non-drop frame) hr:mm:ss:ff to properly work. For example 01:25:00:10 will position it to 1 hr., 25 minutes, 0 sec, and 10 frames. Frames must be a number less than 30.

2.9.2 Saving a Configuration

After setting all the instruments to their desired settings for a Controlatron shot, save the configuration by selecting **File > Save Setup As** from the pull down menu at the top. Name the configuration and then it will be available in the Test Editor screen and the Setup pull down menu for configuring the instrumentation. When a test step is run, it will use this configuration (it loads it by name) to set up all the instruments.

2.9.2.1 Save Options From the Pull Down Menu

- **File > New Setup**

  Reinitializes all values to their default for creating a new setup configuration.

- **File > Save Setup**

  Saves the current settings to the currently selected setup.

- **File > Save Setup As**

  Saves the current settings to a new file name.

- **File > Remove Setup**

  Deletes the current setup from the system. Use caution when removing setups, any test that called the active setup will be made obsolete once the Acquisition setup is deleted.

2.10 Configuring the Power Control System

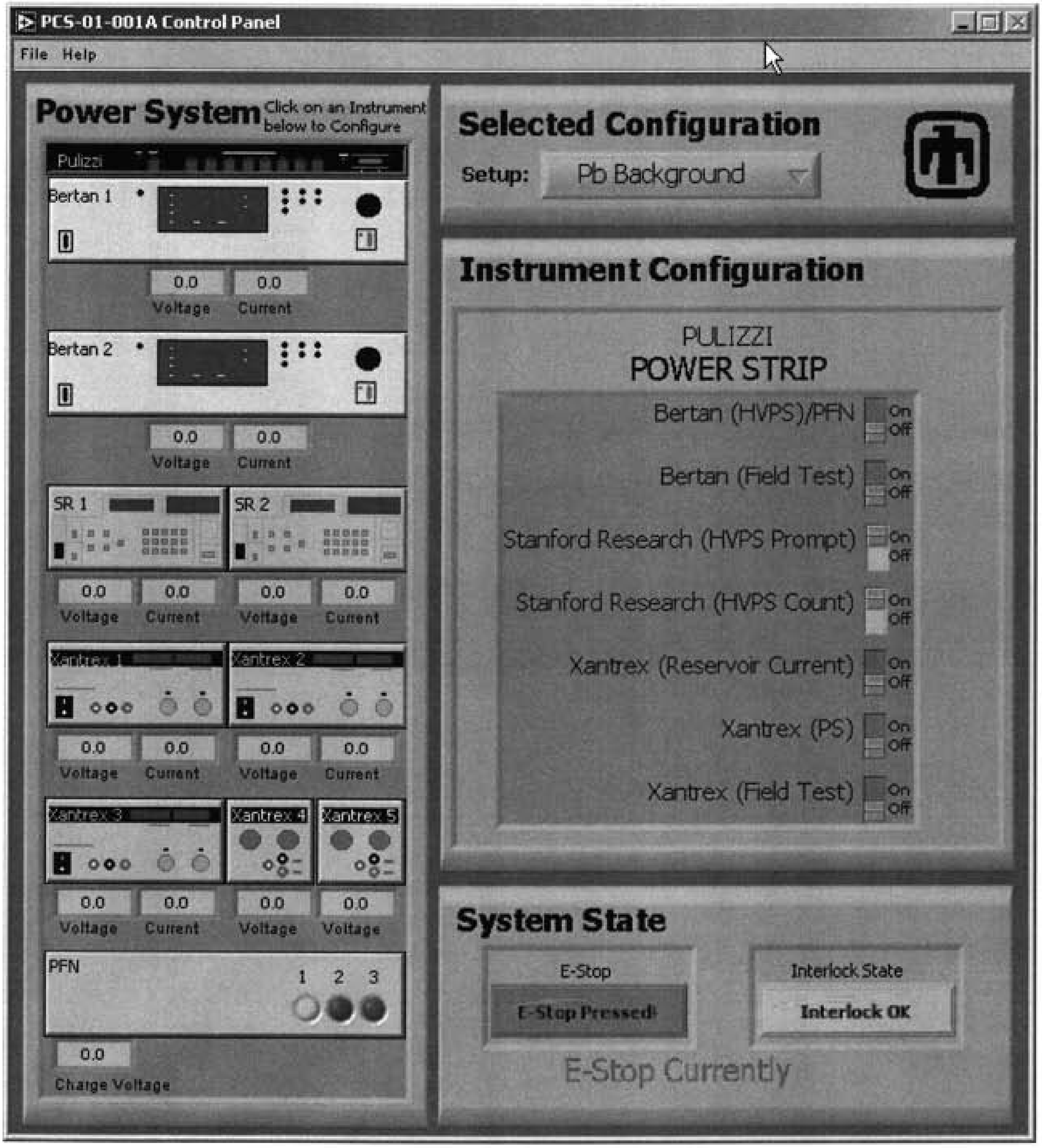


Figure 51. PCS-01-001A Control Panel

Nominal Settings for PCS

	Voltage	Current
Bertan 1	5000	8 mA
Bertan 2	3000	1 mA
SR 1	1550	5.25 mA
SR 2	1400	5.25 mA
Xantrex 1	50	2.15 A (reservoir current)
Xantrex 2	400	1.0 A (grid voltage)
Xantrex 3	0	0
Xantrex 4	12	100 mA
Xantrex 5	60	100 mA
PFN	#1 selected	

The panel of the Power Control System is divided into 4 major sections

1. **Selected Configuration**

   This is a pull down control for selecting a previously created setup. When you select a setup, it will load that setup information into all the Instrument Configuration displays and alter all of the systems in the PCS-01-001A Power System to conform to the settings in the file.

2. **System State**

   Displays the Interlock and E-Stop Status. The Power system will shut down if either of these indicators is red. The E-Stop can be asserted or de-asserted from this panel, however, the Interlocks can only be rectified on the Interlock System Panel (Section 2.11).

3. **Power System**

   This is a visual representation of the instruments the screen controls. Click on the power supply here to configure its settings below. This also indicates the current voltage and current readings for each supply. The power system polls all the supplies once every 2 seconds for these values.

4. **Instrument Configuration**

   This displays the voltage and current limit settings for the selected power supply. Use this to modify the settings for the selected supply. The Pulizzi indicator will allow you to turn off and on selected supplies for a configuration. The PFN allows you to select which PFN is used. These configurations set the desired power level for each device, the actual readings may differ from the setting.

2.10.1 Saving a Configuration

After setting all the power supplies to their desired settings for a Controlatron shot, save the configuration by selecting **File > Save Setup As** from the pull down menu at the top. Name the configuration and then it will be available in the Test Editor screen and the Setup pull down menu for configuring the power system.

Save Options From the Pull Down Menu:

- **File > New Setup:** Reinitializes all values to their default for creating a new setup configuration.
- **File > Save Setup:** Saves the current settings to the currently selected setup.
- **File > Save Setup As:** Saves the current settings to a new file name.
- **File > Remove Setup:** Deletes the current setup.

2.11 Configuring the Interlock System

The interlock system is configured on the Controlatron Interlock Control Center Screen. This screen allows the user to alter the configuration of the interlocks, view the status of each interlock and the overall interlock state.

The Interlock system uses the first 16 lines of the digital I\O card. You can select which channels you want to use with the Active button on the channel. Name the interlock channel and set the polarity of a passing interlock (whether high or low is a passing interlock). The Interlock State indicator shows if the interlock is OK or failed.

The overall interlock state is passing when all of the active interlocks are passing (green).

- The example above shows three active interlocks (door1, door2, and cabinet lid), which are passing when the line is less than 3 volts.
- The Digital I\O card will sense logic high at any voltage above 3V and can accept any voltage up to 28V.
- When you make changes, they are automatically saved and the interlock setup will be the same the next time the system is used.

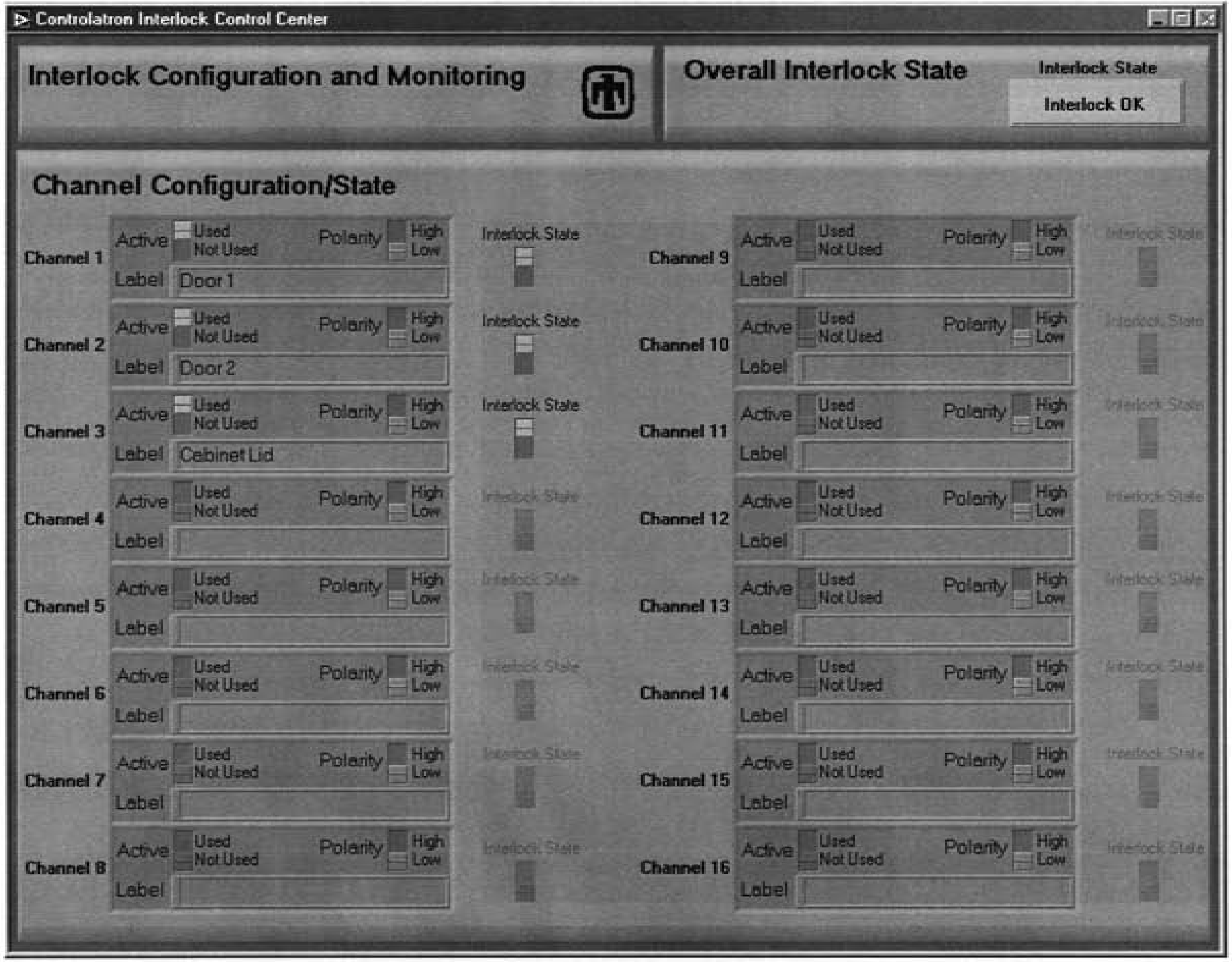


Figure 52. (Previous Page) Controlatron Interlock Control Center Panel

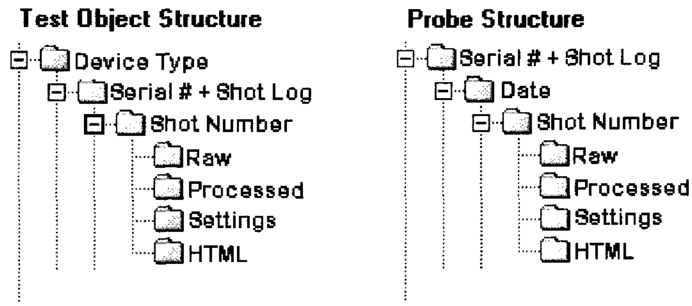


Figure 53. Data Storage Structures (Test Object and Probe)

2.12 Data Output

2.12.1 Output File Structure

For every device tested a directory is created for its test data. This structure is pictured below. This is how to find the data for a particular shot. The structure is different for a probe since it is a calibration based on date. The directory is based on the serial number and then goes down to the shot number (a new directory is added per shot).

2.12.2 Data Output Files

Several data files are dynamically created and place into the above-mentioned Data Storage Structure each time the system is triggered.

2.12.2.1 Shot Log

The Shot log contains data for all the shots performed on a device. It is a comma delimited ASCII file with extension .csv.

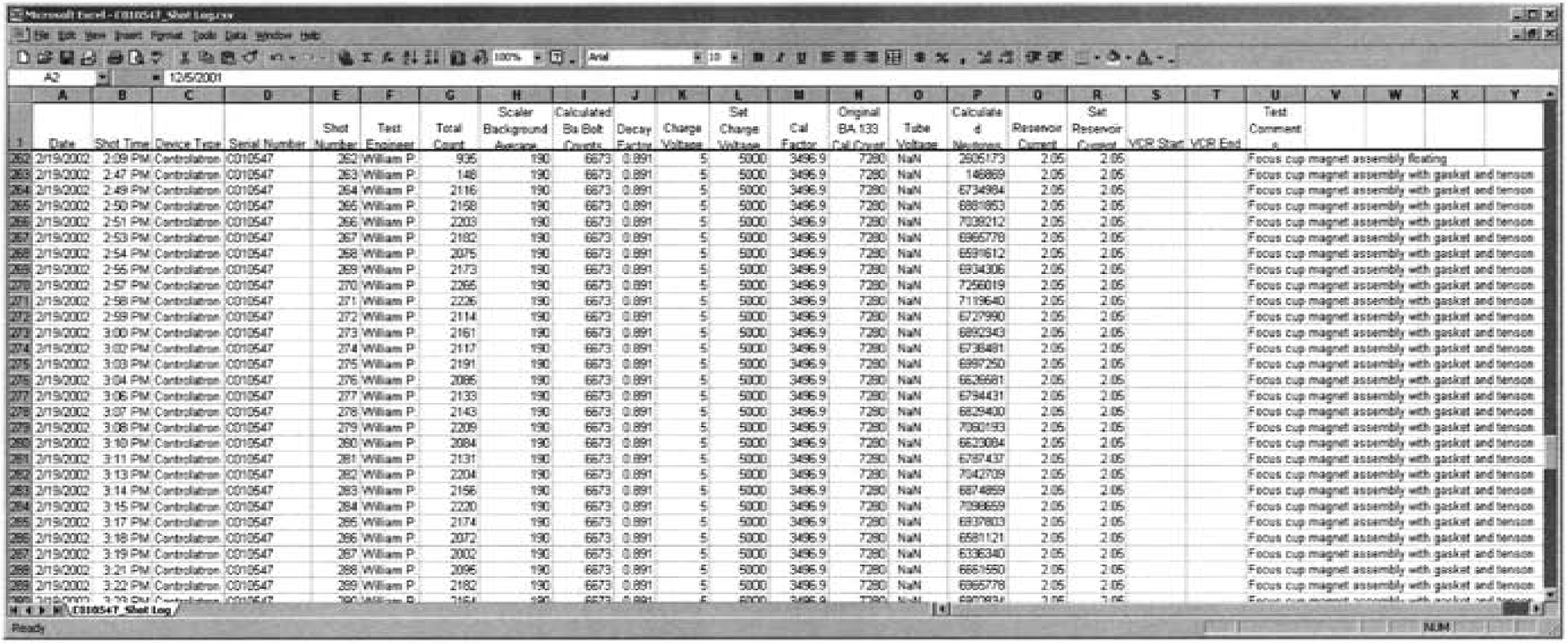
2.12.2.2 HTML Directory

- **Shot Report**

  This is an HTML document located in the HTML directory that displays the header information and all of the scope traces turned on and the discriminator data.

- **Other HTML Files**

  The data viewer allows creation of custom HTML files that are stored here.



1	Date	Shot Time	Device Type	Serial Number	Shot Number	Test Engineer	Total Count	Scaler Background Average	Calculated Ba Bolt Counts	Decay Factor	Charge Voltage	Set Charge Voltage	Cal Factor	Original BA 133 Cal Count	Tube Voltage	Calculated Neutrons	Reservoir Current	Set Reservoir Current	VCR Start	VCR End	Test Comments
262	2/19/2002	2:09 PM	Controlatron	C010547	262	William P.	935	190	6673	0.891	5	5000	3496.9	7280	NaN	2606173	2.05	2.06			Focus cup magnet assembly floating
263	2/19/2002	2:47 PM	Controlatron	C010547	263	William P.	148	190	6673	0.891	5	5000	3496.9	7280	NaN	146869	2.05	2.06			Focus cup magnet assembly with gasket and tension
264	2/19/2002	2:49 PM	Controlatron	C010547	264	William P.	2116	190	6673	0.891	5	5000	3496.9	7280	NaN	6734984	2.05	2.06			Focus cup magnet assembly with gasket and tension
265	2/19/2002	2:50 PM	Controlatron	C010547	265	William P.	2158	190	6673	0.891	5	5000	3496.9	7280	NaN	6881853	2.05	2.06			Focus cup magnet assembly with gasket and tension
266	2/19/2002	2:51 PM	Controlatron	C010547	266	William P.	2203	190	6673	0.891	5	5000	3496.9	7280	NaN	7038212	2.05	2.06			Focus cup magnet assembly with gasket and tension
267	2/19/2002	2:53 PM	Controlatron	C010547	267	William P.	2182	190	6673	0.891	5	5000	3496.9	7280	NaN	6965778	2.05	2.06			Focus cup magnet assembly with gasket and tension
268	2/19/2002	2:54 PM	Controlatron	C010547	268	William P.	2075	190	6673	0.891	5	5000	3496.9	7280	NaN	6591612	2.05	2.06			Focus cup magnet assembly with gasket and tension
269	2/19/2002	2:56 PM	Controlatron	C010547	269	William P.	2173	190	6673	0.891	5	5000	3496.9	7280	NaN	6934306	2.05	2.06			Focus cup magnet assembly with gasket and tension
270	2/19/2002	2:57 PM	Controlatron	C010547	270	William P.	2265	190	6673	0.891	5	5000	3496.9	7280	NaN	7256019	2.05	2.06			Focus cup magnet assembly with gasket and tension
271	2/19/2002	2:58 PM	Controlatron	C010547	271	William P.	2226	190	6673	0.891	5	5000	3496.9	7280	NaN	7119640	2.05	2.06			Focus cup magnet assembly with gasket and tension
272	2/19/2002	2:59 PM	Controlatron	C010547	272	William P.	2114	190	6673	0.891	5	5000	3496.9	7280	NaN	6727990	2.05	2.06			Focus cup magnet assembly with gasket and tension
273	2/19/2002	3:00 PM	Controlatron	C010547	273	William P.	2161	190	6673	0.891	5	5000	3496.9	7280	NaN	6892343	2.05	2.06			Focus cup magnet assembly with gasket and tension
274	2/19/2002	3:02 PM	Controlatron	C010547	274	William P.	2117	190	6673	0.891	5	5000	3496.9	7280	NaN	6738481	2.05	2.06			Focus cup magnet assembly with gasket and tension
275	2/19/2002	3:03 PM	Controlatron	C010547	275	William P.	2191	190	6673	0.891	5	5000	3496.9	7280	NaN	6997250	2.05	2.06			Focus cup magnet assembly with gasket and tension
276	2/19/2002	3:04 PM	Controlatron	C010547	276	William P.	2086	190	6673	0.891	5	5000	3496.9	7280	NaN	6626681	2.05	2.06			Focus cup magnet assembly with gasket and tension
277	2/19/2002	3:06 PM	Controlatron	C010547	277	William P.	2133	190	6673	0.891	5	5000	3496.9	7280	NaN	6794431	2.05	2.06			Focus cup magnet assembly with gasket and tension
278	2/19/2002	3:07 PM	Controlatron	C010547	278	William P.	2143	190	6673	0.891	5	5000	3496.9	7280	NaN	6829400	2.05	2.06			Focus cup magnet assembly with gasket and tension
279	2/19/2002	3:08 PM	Controlatron	C010547	279	William P.	2209	190	6673	0.891	5	5000	3496.9	7280	NaN	7060193	2.05	2.06			Focus cup magnet assembly with gasket and tension
280	2/19/2002	3:10 PM	Controlatron	C010547	280	William P.	2084	190	6673	0.891	5	5000	3496.9	7280	NaN	6623084	2.05	2.06			Focus cup magnet assembly with gasket and tension
281	2/19/2002	3:11 PM	Controlatron	C010547	281	William P.	2131	190	6673	0.891	5	5000	3496.9	7280	NaN	6787437	2.05	2.06			Focus cup magnet assembly with gasket and tension
282	2/19/2002	3:13 PM	Controlatron	C010547	282	William P.	2204	190	6673	0.891	5	5000	3496.9	7280	NaN	7042709	2.05	2.06			Focus cup magnet assembly with gasket and tension
283	2/19/2002	3:14 PM	Controlatron	C010547	283	William P.	2156	190	6673	0.891	5	5000	3496.9	7280	NaN	6874859	2.05	2.06			Focus cup magnet assembly with gasket and tension
284	2/19/2002	3:15 PM	Controlatron	C010547	284	William P.	2220	190	6673	0.891	5	5000	3496.9	7280	NaN	7098659	2.05	2.06			Focus cup magnet assembly with gasket and tension
285	2/19/2002	3:17 PM	Controlatron	C010547	285	William P.	2174	190	6673	0.891	5	5000	3496.9	7280	NaN	6937803	2.05	2.06			Focus cup magnet assembly with gasket and tension
286	2/19/2002	3:18 PM	Controlatron	C010547	286	William P.	2072	190	6673	0.891	5	5000	3496.9	7280	NaN	6581121	2.05	2.06			Focus cup magnet assembly with gasket and tension
287	2/19/2002	3:19 PM	Controlatron	C010547	287	William P.	2002	190	6673	0.891	5	5000	3496.9	7280	NaN	6336340	2.05	2.06			Focus cup magnet assembly with gasket and tension
288	2/19/2002	3:21 PM	Controlatron	C010547	288	William P.	2096	190	6673	0.891	5	5000	3496.9	7280	NaN	6661550	2.05	2.06			Focus cup magnet assembly with gasket and tension
289	2/19/2002	3:22 PM	Controlatron	C010547	289	William P.	2182	190	6673	0.891	5	5000	3496.9	7280	NaN	6965778	2.05	2.06			Focus cup magnet assembly with gasket and tension

Figure 54. Example Shot Log in Excel

2.12.2.3 Raw Directory

- **Discriminator Raw.csv**

  Contains the raw discriminator counts as returned from the instrument without processing. It is a comma delimited ASCII file with extension .csv.

- **Scope Raw.csv**

  Contains the raw Scope Data as returned from the instrument without processing. It is a comma delimited ASCII file with extension .csv.

2.12.2.4 Processed Directory

- **Discriminator Processed.csv**

  Contains the discriminator counts as returned from the instrument with processing. It is a comma delimited ASCII file with extension .csv.

- **Scope Processed.csv**

  Contains the Scope Data as returned from the instrument with processing. It is a comma delimited ASCII file with extension .csv.

2.12.2.5 Settings Directory

- **Instrumentation Setup.csv**

  Contains all the setup information for the Acquisition and Power systems.

2.13 Installing or Reinstalling the Software

Overview:

This section describes how to reinstall the system onto a new computer, load the necessary drivers and software.

- **Step 1:** Prepare the computer. Load Windows 2000 operating system on it if it isn't on yet. Do not install any of the GPIB or other cards into the system yet.

- **Step 2:** Install National Instruments GPIB software. This is also available free of charge from www.ni.com. This is required to run the GPIB card rev2.0 or later

C010544_014_Scope 1_Processed.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Date	Shot Time	Device Type	Serial Number	Shot Number	Test Engineer	Total Count	Scaler Background Average	Ba Bolt Counts	Decay Factor	Charge Voltage	Set Charge Voltage	Cal Factor	BA 133 Calibration Count	Tube Voltage	Calculated Neutrons	Reservoir Current	Set Reservoir Current
2																		
3	9/18/01	3:15 PM	Controlatror	C010544	14	William P. Noel	2093	6790	14167	1	0	5000	2	8420	-1.5	-11487	0	2.15
4	t0 = -0.000002																	
5	delta t = 0.000000																	
6	0.032	-0.28014	0.03598															
7	0.024	-0.42013	-0.132074															
8	0.032	-0.34013	0.031979															
9	0.016	-0.48013	-0.128072															
10	0.016	-0.44013	0.508131															
11	0.096	-0.40013	-0.448175															
12	0.12	-0.48013	0.079994															
13	0.024	-0.54012	-0.304129															
14	0.072	-0.56012	0.027978															
15	-0.024	-0.60012	-0.096062															
16	-0.032	-0.68011	-0.112067															
17	0.024	-0.64012	-0.068053															
18	0.032	-0.72011	-0.112067															

Figure 55. Example Processed Data File in Excel

C010544_014_Instrumentation Setup.csv

	A	B	C	D	E	F	G	H	I	J
1	PCS 01 001A Power System Settings									
2	Instrument	Power State	Set Voltage	Actual Voltage	Set Current	Actual Current				
3	Bertan 1	OFF	5000	0	0	0				
4	Bertan 2	ON	3000	0	3000	0				
5	Stanford Reseach 1	ON	-1550	0.01	0	-1550				
6	Stanford Research 2	ON	-1225	0.01	0	0				
7	Xantrex 1	ON	50	2.15	50.01	0				
8	Xantrex 2	ON	400	1	400.08	0				
9	Xantrex 3	OFF	0	0	0	0				
10										
11	PFN	Selected PFN	Charge Voltage							
12		None Selected	0							
13										
14	Controlatron Acquisition System Settings									
15	Stanford research 430									
16	Bin Width	# of Bins	Trigger Threshold	Trigger Slope	Descriminator Threshold	Descriminator Slope	Start Time	Duration		
17	15	4096	2	Rising	0.205	Rising	2.40E-02	2.37E+00		
18										
19	DG 535 Pulse Generator 1	On								
20	Output Signal	Output Delay	mode	load	inversion	amplitude	offset	Relative to:(TI)		
21	A	0.00E+00	TTL	50 Ohm	Normal	4	0	TO		
22	B	0.00E+00	TTL	50 Ohm	Normal	4	0	TO		
23	C	0.00E+00	TTL	50 Ohm	Normal	4	0	TO		
24	D	5.00E-02	TTL	50 Ohm	Normal	4	0	TO		
25										
26	DG 535 Pulse Generator 1	On								
27	Output Signal	Output Delay	mode	load	inversion	amplitude	offset	Relative to:(TI)		
28	A	0.00E+00	TTL	50 Ohm	Normal	4	0	TO		
29	B	0.00E+00	TTL	50 Ohm	Normal	4	0	TO		
30	C	0.00E+00	TTL	50 Ohm	Normal	4	0	TO		
31	D	0.00E+00	TTL	50 Ohm	Normal	4	0	TO		
32										

C010544_014_Instrumentation Set

Figure 56. Example Settings File (part 1) in Excel

C010544_014_Instrumentation Setup.csv

	A	B	C	D	E	F	G	H	I	J
34	acqusition mode	Timebase	record length	trigger mode	trigger source	trigger coupling	trigger slope	voltage level	trigger position	trigger holdoff
35	Sample	2µs/div	500 points	Normal	Channel 1	DC	Negative	-0.3	10	0
36										
37	Channel	Ch Active	scaling	offset	coupling	impedance	bandwidth			
38	Prompt Neutrons	ON	0.2	0	DC	50ohms	100 MHz			
39	Target V	ON	0.5	0	DC	50ohms	100 MHz			
40	Tube Current	ON	0.1	0	DC	50ohms	100 MHz			
41	Test 4	Channel OFF	1	0	DC	50ohms	20 MHz			
42										
43	Scope 2 Settings									
44	acqusition mode	Timebase	record length	trigger mode	trigger source	trigger coupling	trigger slope	voltage level	trigger position	trigger holdoff
45	Sample	500µs/div	5000 points	Normal	Auxiliary	DC	Positive	1	10	0
46										
47	Channel	Ch Active	scaling	offset	coupling	impedance	bandwidth			
48	Test 1	ON	1	0	DC	50ohms	100 MHz			
49	Test 2	Channel OFF	1	0	DC	50ohms	20 MHz			
50	Test 3	Channel OFF	1	0	DC	50ohms	20 MHz			
51	Test 4	Channel OFF	1	0	DC	50ohms	20 MHz			

Figure 57. Example Settings File (part 2) in Excel

- **Step 3:** Install National Instruments Serial Software rev1.7 or later.

- **Step 4:** Install LabVIEW Full or Professional Development system if you need to edit source code. Select the default installation settings.

- **Step 5:** If you did not perform step 4, install National Instruments NIDAQ rev 6.9 for the DIO card.

- **Step 6:** Shut down the Computer

- **Step 7:** Add the cards to the system. There should be 2 NI PCI GPIB cards (or equivalent part # 778032-01). You should install 1 NI 6527 DIO Card for Interlock Control and video Switch Control. Install a NI 8 port (or more) RS232 serial card (part # 777642-08).

- **Step 8:** Turn on the system. The system should find the cards added.

- **Step 9:** Verify the GPIB Installation by going to "Start - Programs - National Instruments – NI-488.2 –Getting Started Wizard". Click on verify your hardware and software configuration. This will test the GPIB cards. If this passes, then you should be able to use the GPIB instruments through software. Connect in the GPIB and serial cables as described in the hardware manual.

- **Step 10:** Verify the DAQ card and serial port operation by launching the measurement and automation explorer located at: "Start-Programs-National Instruments-Measurement and Automation Explorer." Click on the devices tab and device number 1 should show up as a 6527 card. Run the test panel to verify proper installation. Verify the serial ports by going to the Devices and Interfaces tab and selecting ports. At least 10 ports should be present if the serial board and software are properly loaded.

- **Step 11:** Load the Controlatron Software. Insert the CD and double click on Setup. Accept the default settings and it should install the Controlatron Software. Proceed with configuring the system below.

2.14 Configuring the System

Setting the serial port settings

The COM ports need to be configured so that the COM port numbers for the NI serial card are 11-18. Go to the control panel, and go to the NI-Ports utility and set #1 to port 11, 2 to port 12, etc.

2.15 Manually Testing and Troubleshooting the Instruments

If the instruments are not responding properly in the Controlatron program, you can verify operation of the instruments with the low level drivers.

The general procedure for testing an instrument using its driver example is to go to the Controlatron Physical Drivers directory and go to the directory associated with the instrument. Open the VI Tree for that instrument and go to the block diagram for the VI tree. Double click on an example VI for the instrument to open its panel. In the Instrument Descriptor, change the GPIB address to match what the instrument is set for, then set the other controls to set the instrument as desired. Click on the run arrow at the top left and the sample program should configure the instrument to the settings or return an error if the system is not working properly.

- Change the Instrument descriptor to match the GPIB address that the instrument is set for.
- **DG535:** Open the VI marked dg535 Sample set the GPIB address and run it. The Delay generator should output pulses according to the settings.
- **SR430:** Open the VI labeled Example SR430 Application set the GPIB address and run it. Alter the settings as desired and press the "arm" button. The 430 should configure itself. Pass a trigger to the trigger input on the 430, and then press the read button to read the counts. If you have any kind of appropriate signal, it should return the counts on the graph.
- **SR350 Power Supplies:** Open the VI marked SR PS3xx Getting Started, set the GPIB address, the desired voltage and run it. The Power supply should get set to the appropriate voltage.
- **Xantrex Power Supplies:** Open the VI marked XanXFR Init and Set, set the GPIB address, the desired voltage and run it. The Power supply should get set to the appropriate voltage.
- **Bertan Power Supplies:** Open the VI marked Init and Set_Bertan, set the GPIB address, the desired voltage and run it. The Power supply should get set to the appropriate voltage.

3. Programmers Guide

3.1 Overview of Programming Architecture

The Controlatron Software is programmed in LabVIEW 6i. You must have the LabVIEW 6i Full or Pro Development package from National Instruments to edit the source code. It is recommended to upgrade to the 6.02 bug patch (available free from www.ni.com).

The basic programming architecture style is based on the "Message Management" system described in Section 3.4. This breaks the code into several major modules that each can operate independently to control a sub-system of the Controlatron tester. The main program is called the "Controlatron Control Center Top Level.vi". This VI is responsible for controlling the main user interface and running tests. This module launches, controls, and utilizes modules for the power system, acquisition system, and interlock system. These modules in turn call libraries of high-level instrument drivers that perform the software tasks for controlling the hardware equipment, instrumentation, and interfaces. The figure on the next page illustrates the hierarchy of the software.

In the following sections, the modules and their functions are described. Following that is a description of the hardware drivers and tutorials for the Message Manager and the Queued State Machine architectures.

3.2 System Modules

The code for each system module is contained in it's own directory. Modules are state machine programs that serve as user interfaces, or programs to control a system. These are developed as state machines by the programmer that utilize the message manager. These are the basically all the high level components of a LabVIEW application created using the message manager architecture. Refer to Section 3.4 on the message manager to learn what a module is and how to use messaging with the modules. The VI with the name "Top Level" in it is the heart of each module and the place to begin at to debug or add functionality.

3.2.1 Controlatron Control Center (CIC Top Level.vi)

3.2.1.1 Overview

This module is the main user interface, the first and top-level program called, and the test logic controller. This controls the running the test, test creation, and basic data viewing. This is a state machine based program that uses the message manager (see the tutorial below).

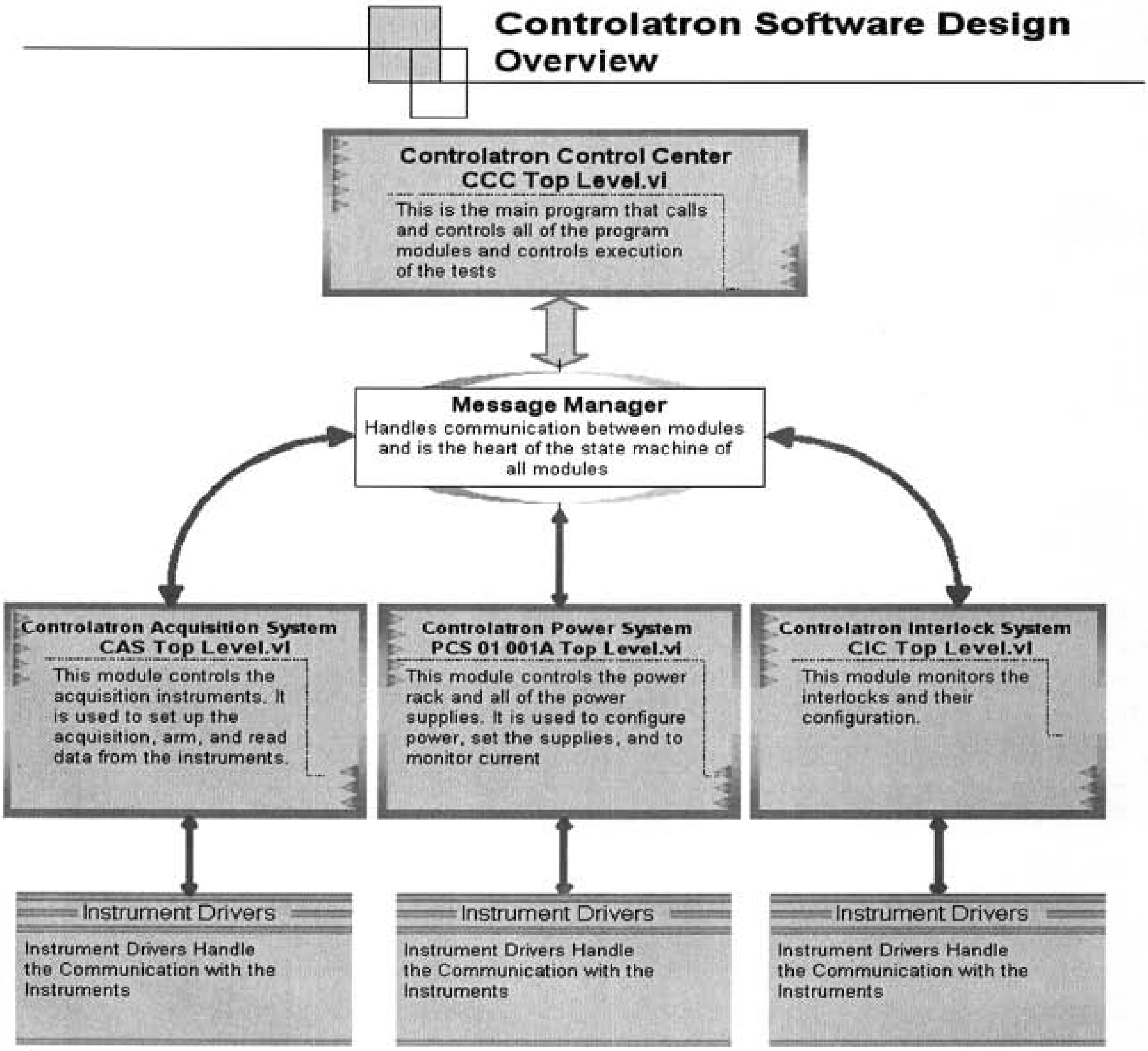


Figure 58. Hierarchies and General Architecture of Controlatron Software

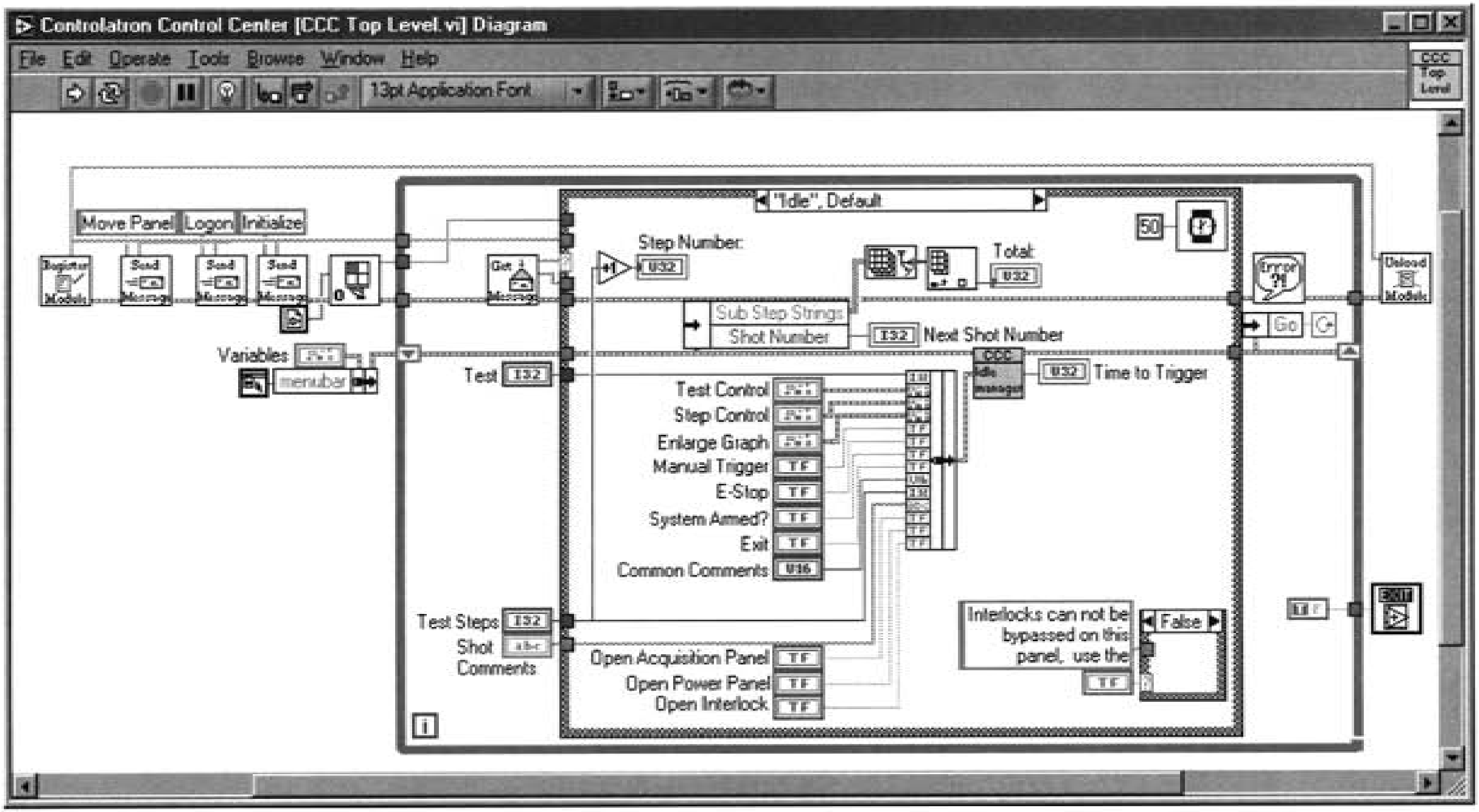


Figure 59. Controlatron Control Center Block Diagram

3.2.1.2 Control Center Operation

This module is the primary application for the program. The Controlatron Control Center "idle" case is responsible for checking the user buttons and determines what actions the module needs to perform. The CCC Idle Case Manager.vi performs most of the logic in determining what actions are to be taken and sends messages accordingly.

Other Key cases include:

- **Initialize**: This case loads the other 3 modules and performs other initialization routines when this program is first run.
- **Change Test Object**: this calls the configuration screens to change the Controlatron and probes.
- **Test Control:** This performs the stop\start\pause logic for the system.
- **Step Control**: Performs the logic for fast-forward and rewind functions.
- **Change selected sub step:** this sends the configurations to the power and acquisition system after a new step is called.
- **Configure Acquisition\Power\Interlocks:** These cases call up the front panel of the other modules.
- **Arm System:** Sends a command to arm the acquisition module.
- **Prepare for Trigger:** This case gets the acquisition and power setups, and then triggers the system.
- **Receive Plot Data:** The acquisition module calls this case. It sends the full data to the control center to this case, which creates all the data files and updates the displays.
- **Enlarge Plot**: This case brings up a magnified plot display for the selected graph.
- **User Manager**: this case pops open the user manager to add new users to the system.

3.2.1.3 Control Center Hardware and Instruments

The Control Center itself does not call any hardware. Instead it draws upon the other modules for instrument communication.

3.2.2 Controlatron Acquisition System (CAS Top Level.vi)

3.2.2.1 Overview

This module controls the instrumentation for the Controlatron data acquisition. It is also the user interface for manually controlling the instruments and defining their test configurations. This includes the DG535 delay generators, TK TDS420 scopes, SR430 Photon Counter\Discriminator, a Pulizzi Relay controller, the VCR, and a Video switch.

3.2.2.2 Acquisition System Operation

The Controlatron Acquisition System "idle" case is responsible for checking the user buttons and determines what actions the module needs to perform. The CAS Idle Case Manager.vi performs most of the logic in determining what actions are to be taken and sends messages accordingly. The "State Checker for VCR.VI" handles the VCR control (it is separated in case a separate application needs to control the VCR).

The PCS Manage Power State Change.vi calls the power supply drivers and sets them. It is located in the "Power State Change" case. All the different power supply drivers have a high-level identical driver interface used in this sub-VI. The "read values" case reads the voltage and current settings of the supplies.

The acquisition system is responsible for saving and reading it's own setup configurations. They are stored in the CAS Setups directory and have a .cas extension.

Key cases include:

- **Idle:** Checks if the system needs to take action
- **Initialize:** Initializes the system
- **Power State Change:** called when an instrument is turned off or on in the Pulizzi.
- **New Setup:** Loads default values into the configuration
- **Load Setup:** Loads the selected configuration into the module.
- **Save Setup:** saves the current setup.
- **Arm:** Configures and Arms all the Instruments
- **Trigger:** Sends a trigger to the delay generator then tells the system to go to the read data case.
- **Read Data:** Waits for the end of acquisition then collects the data and sends it to the CCC module.

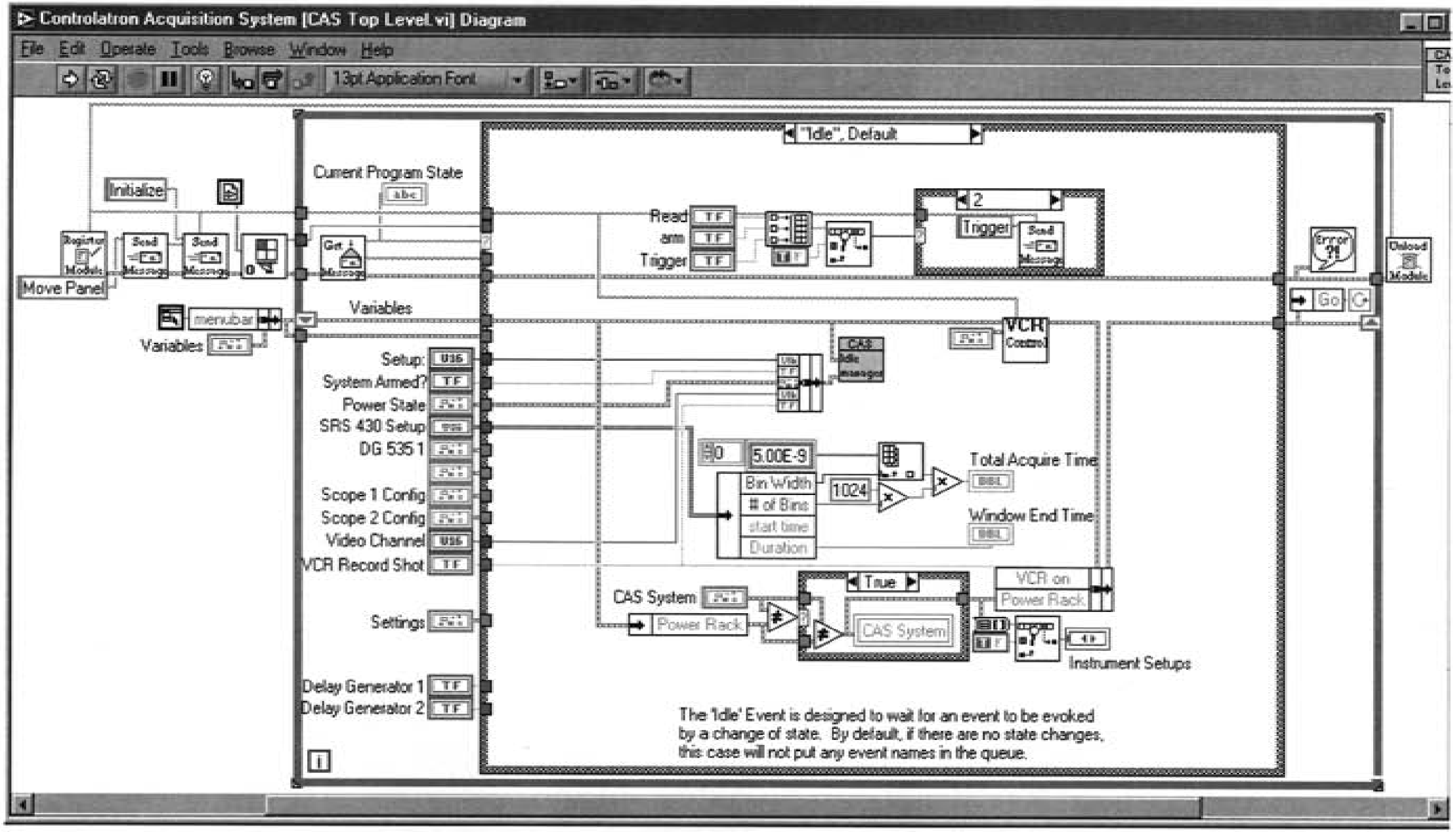


Figure 60. Controlatron Acquisition System Block Diagram

3.2.2.3 Hardware and Instruments for the Acquisition Module

- **Pulizzi 3302 FS**

  This is an 8-channel programmable power relay module. It is controlled via RS232 serial. For the Power Rack connect it to port 2 from the NI fan out cable on the RS232 card. It requires a DB9 Female to DB9 Female direct (not null) serial cable. It powers up the Racks Power supplies. The Drivers are in the physical drivers\Pulizzi directory.

  Connect the instruments to the Pulizzi as follows:

  - Ch1: DG535 1 (master)
  - Ch2: DG535 2 (slave)
  - Ch3: SR430 Counter\Discriminator
  - Ch4: Scope 1
  - Ch5: Scope 2
  - Ch6: Video Monitor System
  - Ch7: Video Record System
  - Ch8: Spare Power Strip

- **2 DG535 Delay Generators:**

  The acquisition system commands up to two Stanford Research Delay Generators. Drivers are in the physical drivers\dg535 directory. They are GPIB-Controlled and should be daisy chained to the GPIB Bus on the instrumentation rack.

- **2 Tektronics TK TDS 420 Scopes**

  The acquisition system commands up to two Tek 420 Scopes. Drivers are in the physical drivers\TDK420 directory. They are GPIB-Controlled and should be daisy chained to the GPIB Bus on the instrumentation rack.

- **SR430 Counter\Discriminator**

  The acquisition system commands a Stanford Research 430 Counter\Discriminator. Drivers are in the physical drivers\SR430 directory. They are GPIB-Controlled and should be daisy chained to the GPIB Bus on the instrumentation rack.

3.2.3 Controlatron Power System

3.2.3.1 Overview

This module controls everything with the power rack. It is also the user interface for manually controlling the power rack and defining its test configurations. This includes the Bertan Power

Supplies, Xantrex Low Voltage Power supplies, Stanford Research Supplies, and a Pulizzi Relay control.

3.2.3.2 Power System Operation

The Controlatron Power System "idle" case is responsible for checking the user buttons and determines what actions the module needs to perform. The PCS01 Idle Case Manager.vi performs most of the logic in determining what actions are to be taken and sends messages accordingly.

The PCS Manage Power State Change.vi calls the power supply drivers and sets them. It is located in the "Power State Change" case. All the different power supply drivers have a high-level identical driver interface used in this sub-VI. The "read values" case reads the voltage and current settings of the supplies.

The Power system is responsible for saving and reading it's own setup configurations. They are stored in the PCS 01 001A Setups directory and have a .pcs extension.

Key Cases include:

- **Idle:** Checks if the system needs to take action
- **Initialize:** Initializes the system
- **Power State Change:** Called when an instrument is turned off or on in the Pulizzi
- **New Setup:** Loads default values into the configuration
- **Load Setup:** Loads the selected configuration into the module
- **Save Setup:** Saves the current setup
- **Power Setting Change:** Sends the power settings to the supplies
- **Read Values:** This is periodically called to read the voltage and current
- **Get Power Setup String:** This is called by the CCC main program and sends the configuration state of the power system for the CCC's system data files
- **Change PFN:** Sets the PFN to use
- **E-stop:** Handles the logic when the E-stop state changes

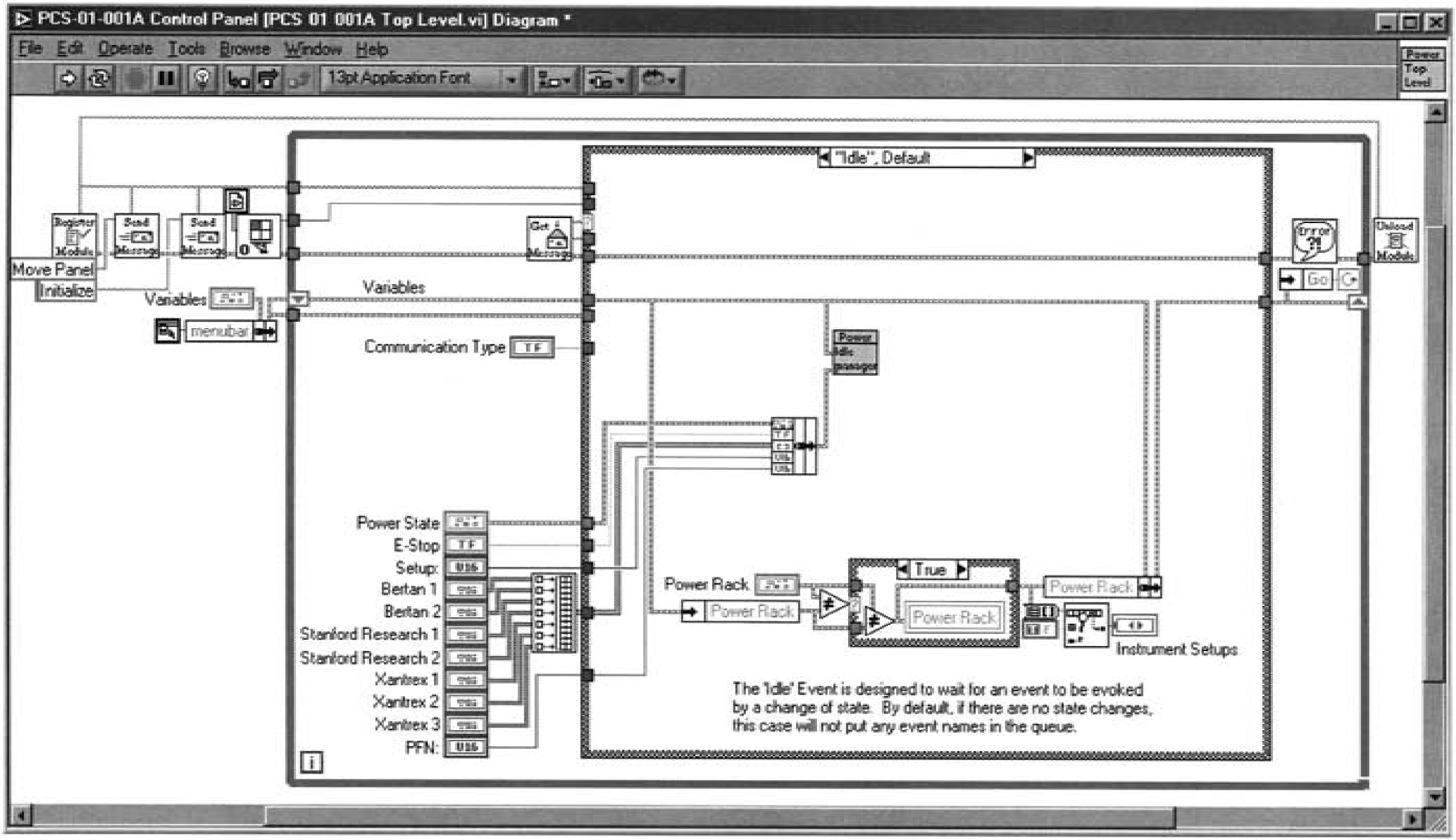


Figure 61. Power System Block Diagram

3.2.3.3 Hardware and Instruments for the Power Module

The Power Module is designed to control and monitor the PCS-01-001A Optically Isolated, Multi-Output Power Supply System. This system includes:

- **Pulizzi 3302 FS**

  This is an 8-channel programmable power relay module. It is controlled via RS232 serial. For the Power Rack connect it to port 1 from the NI fan out cable on the RS232 card. It requires a DB9 Female to DB9 Female direct (not null) serial cable. It powers up the Racks Power supplies. The Drivers are in the physical drivers\Piluzzi directory.

  Connect the power supplies to the Pulizzi as follows:

  - Ch1: Bertan1 (HVPS), + PFN
  - Ch2: Bertan2 (Field Test)
  - Ch3: Stanford Research Power Supply 5K PS350 #1 (HVPS Prompt) + Xantrex 4
  - Ch4: Stanford Research Power Supply 5K PS350 #2 (HVPS Count) + Xantrex 5
  - Ch5: Xantrex 1 (Reservoir Current)
  - Ch6: Xantrex 2 (PS)
  - Ch7: Xantrex 3 (Field Test)
  - Ch8: Fan for cooling the Rack

- **Three Xantrex XHR Power Supplies:**

  The system controls up to 5 Xantrex XHR series power supplies. Drivers are in the physical drivers\Xantrex directory. They are GPIB-Controlled and should be daisy chained to the GPIB Bus.

- **Two Stanford Research Power PS350 Supplies:**

  The system controls up to two SR PS350 power supplies. Drivers are in the physical drivers\SR PS3xx directory. They are GPIB-Controlled, and should be daisy chained to the GPIB Bus.

- **Two Bertan 225-20R Power Supplies:**

  The system controls up to two Bertan 225 series power supplies. Drivers are in the physical drivers\Bertan directory. They are GPIB-Controlled, and should be daisy chained to the GPIB Bus.

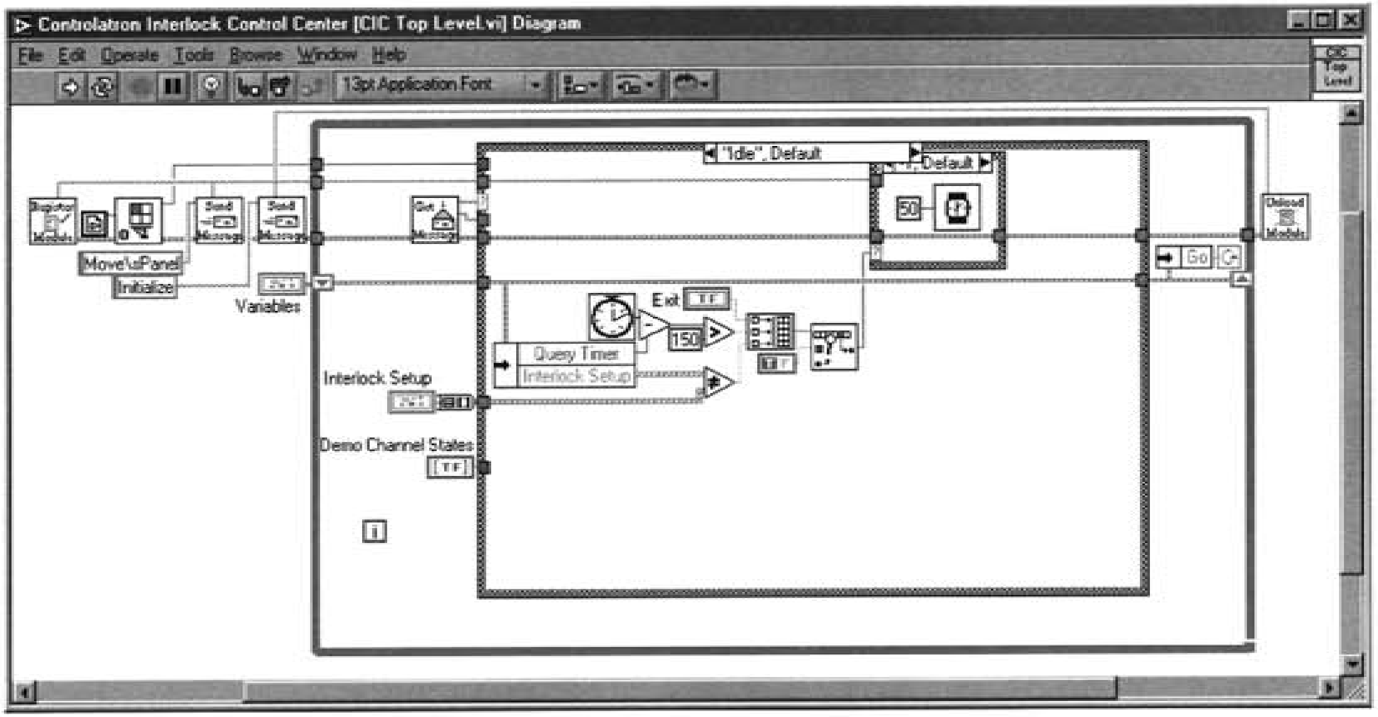


Figure 62. Controlatron Interlock Control Module Idle Case

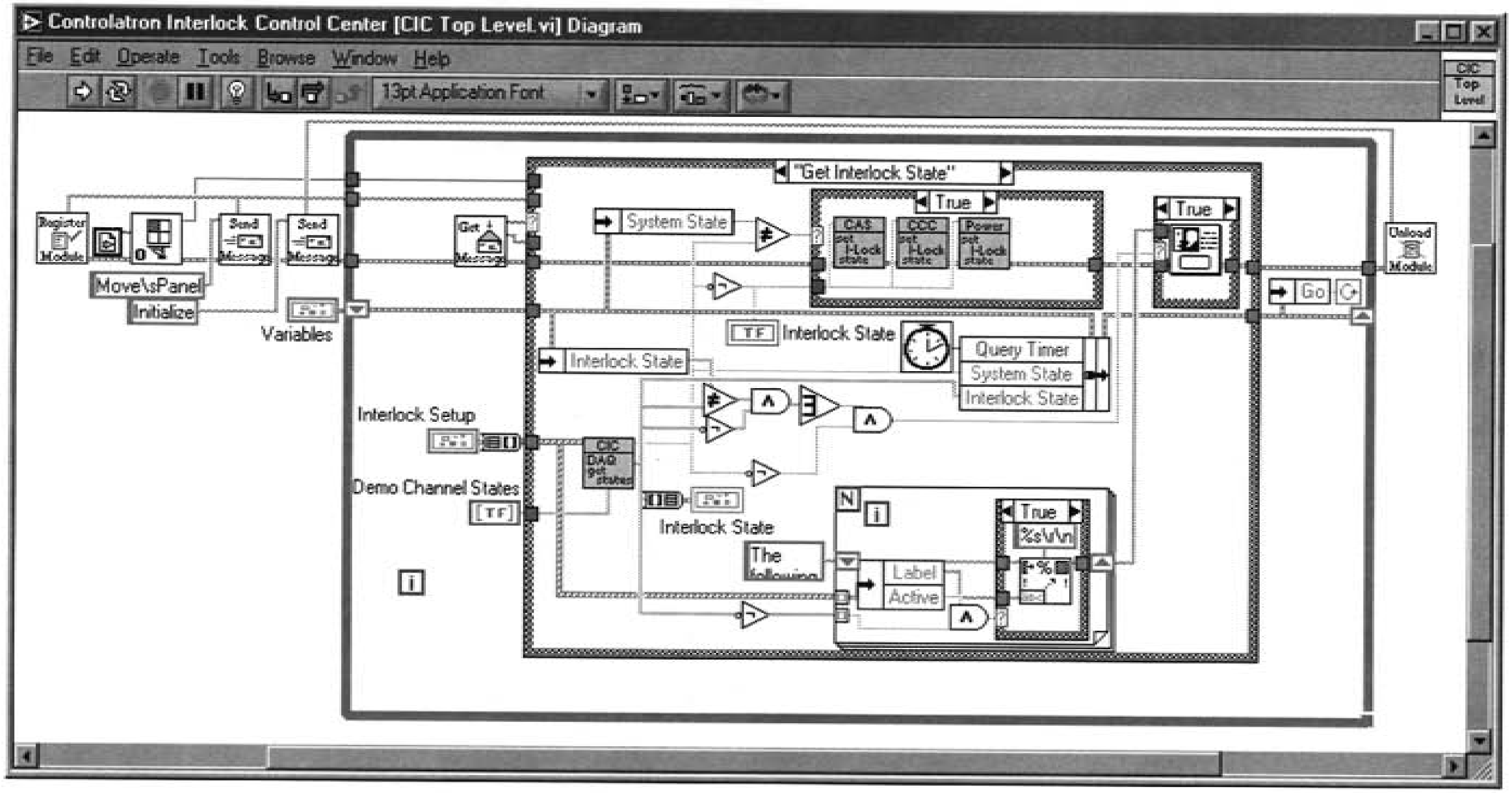


Figure 63. Controlatron Interlock Control Module 'Get Interlock State' Case

3.2.4 Controlatron Interlock System (CIC Top Level.VI)

3.2.4.1 Overview

This module monitors the interlocks and informs the other modules of the interlock state so they can shut down in the event of an interlock fault.

3.2.4.2 Operation

The Idle case checks to see if the user changes the configuration and if so it replaces the interlock setup variable data.

The CIC DAQ get interlock status VI, calls the DIO card and checks the first 16 input channels of the DIO card. If the Interlock states change, it sends the message "set interlock state" to all other modules telling of its condition.

3.2.4.3 Hardware and Instruments

- **DIO 6527**

  The Interlock Module utilizes the National Instruments DIO 6527 card. This card is an optically isolated DIO card that can accept 24 channels of 0–28 V Input (anything above 2V is a logic high) and can switch 24 channels of digital outputs 0–60 V up to 0.120 amps.

The Interlock Module only calls the DIO card drivers. These are basic DAQ Read Digital Port calls standard to LabVIEW that control any series of DIO card (so long as the hardware has enough lines).

3.3 System Instrument Drivers

3.3.1 Overview

The code for the Controlatron is highly dependent on drivers, which are sub-programs that handle communications with a particular device or instrument. These drivers allow the high level program to operate independently of knowing the nuts and bolts of how to communicate with a device. High-level drivers were written so that it is very simple to implement the major functionality of the system. The drivers are located in the "Physical Drivers" sub-directory of the source code. Each instrument has it's own directory where it's driver is located.

The instrument drivers were all written to conform to VXI PnP standards for LabVIEW drivers. This dictates a common set of features, connector panes, and style. The driver sets contain functions listed below, examples, and a VI tree that illustrates all of the drivers for the device and sorts them by function.

Driver Function Classes:

- **Initialize:** This is the first VI to be called. It opens a communication channel with the instrument and where applicable performs a reset and ID query of the instrument.
- **Setup\Config:** These calls perform a setup or configuration command to an instrument.
- **Action Status**: These calls command an instrument to perform an action or trigger.
- **Data:** These calls collect data from an instrument.
- **Utility:** These calls perform a miscellaneous function or calibration on an instrument.
- **Close:** This is the last VI to be called. It shuts down the communication channel with the instrument and frees up its resources.
- **Example:** These are high-level examples that show you how to utilize the drivers to perform a high level function.
- **Getting Started:** This is a high level call designed to quickly check an instrument.

3.3.2 Bertan Drivers

The Bertan high voltage power supply is controlled by GPIB. PrimeCore Systems using LabVIEW and VISA Calls wrote Bertan drivers. All of the power supply drivers have 4 high level drivers (at the top of the VI Tree) that are used in the Power System top level software that have identical functionality and interfaces for all the power supplies. This includes an Initialize and Set, Set Voltage and Current, Turn all Off, and Read Voltage and Current. These high level drivers in turn call lower level drivers as necessary to perform those basic tasks.

3.3.3 DG535 Pulse Generator Drivers

The DG535 is a GPIB controlled instrument. The drivers were written by PrimeCore using LabVIEW and VISA Calls. There is a high level example VI that shows how to use the drivers and can test the instrument independently of any other software. The drivers were designed to accept the typedef control illustrated below to configure the instrument. Use the DG535 Setup.ctl when writing software using these drivers.

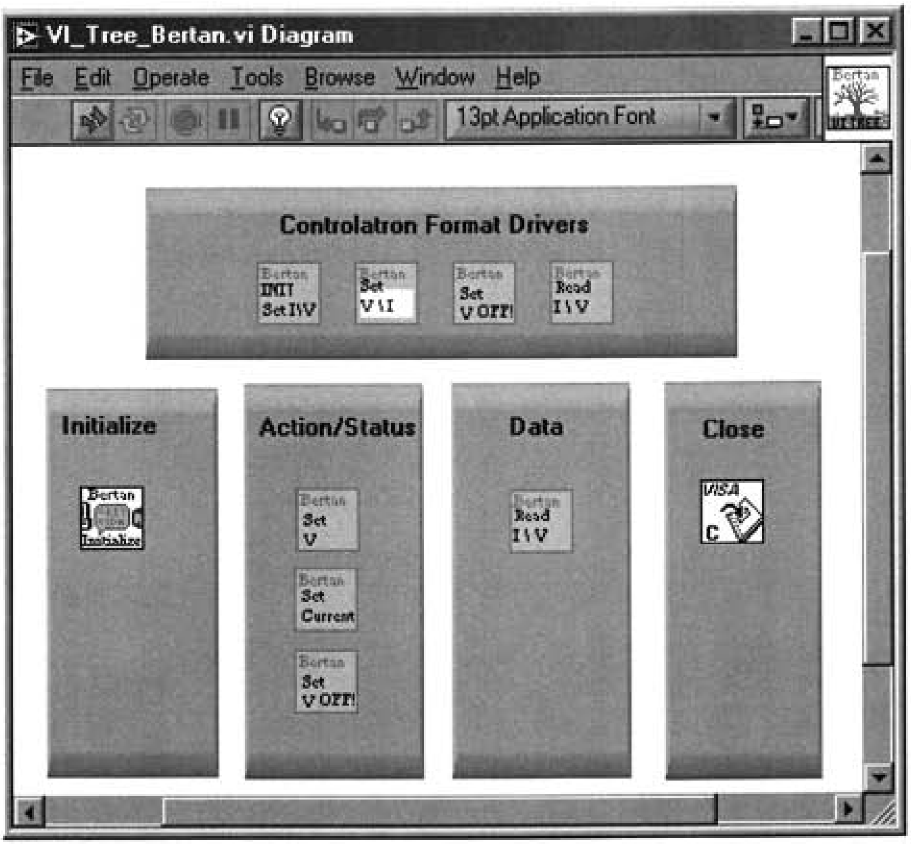


Figure 64. VI Tree for the Bertan High Voltage Power Supply Drivers.

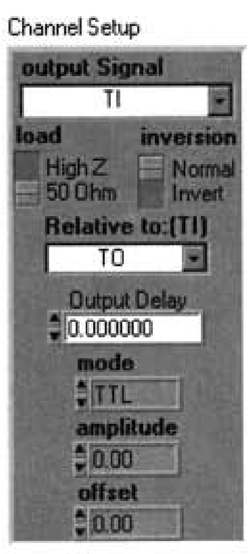


Figure 65. Channel Setup Custom Control associated with DG 535 Drivers

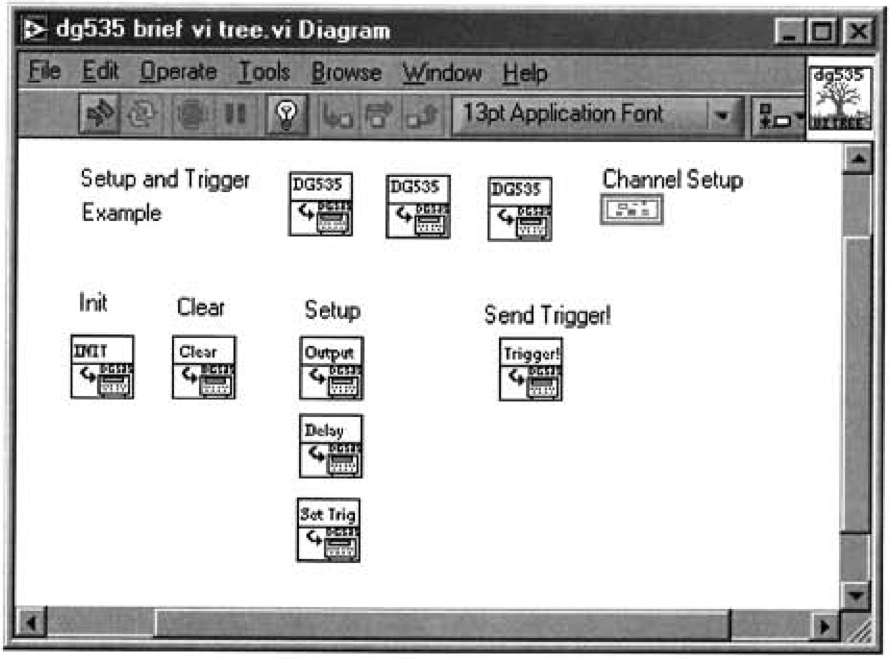


Figure 66. DG535 VI Tree

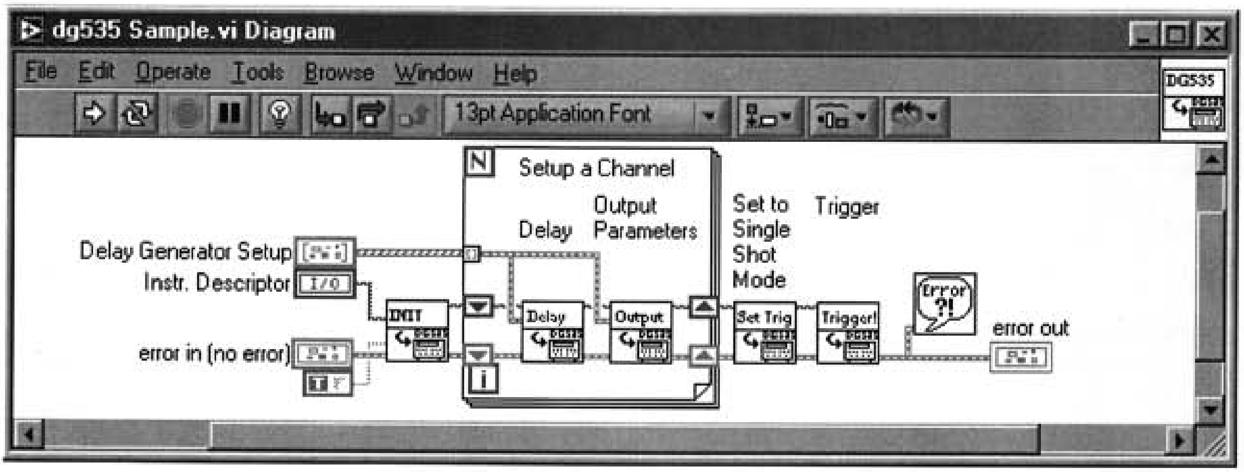


Figure 67. DG535 Programming Example

This shows the block diagram of the example that initializes, configures, and triggers the delay generator.

3.3.4 Pulizzi Power Relay Switch Drivers

The Pulizzi relay modules are controlled by RS232 serial. PrimeCore using LabVIEW and VISA Calls wrote the drivers. It is a very basic set of drivers that only supports the necessary functionality for the Controlatron. Typically it is only necessary to call the Init and log on, then the power up selected to control the device.

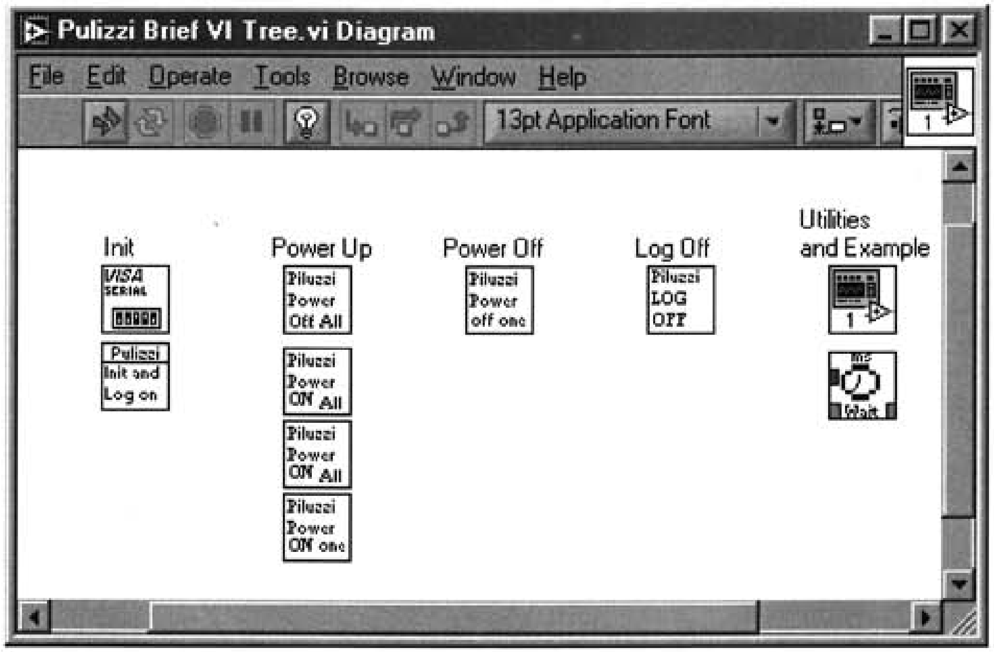


Figure 68. Pulizzi Driver Tree

3.3.5 SR430 Counter\Discriminator Drivers

The SR430 is a GPIB controlled instrument. PrimeCore using LabVIEW and VISA Calls wrote the drivers. There is a high level example VI that shows how to use the drivers and can test the instrument independently of any other software. The setup and arm VI and the poll and read VI can be used to configure and arm the device, then retrieve the data in two steps. The drivers were designed to accept the type-def control illustrated below to configure the instrument. Use the SR430 Setup.ctl when writing software using these drivers.

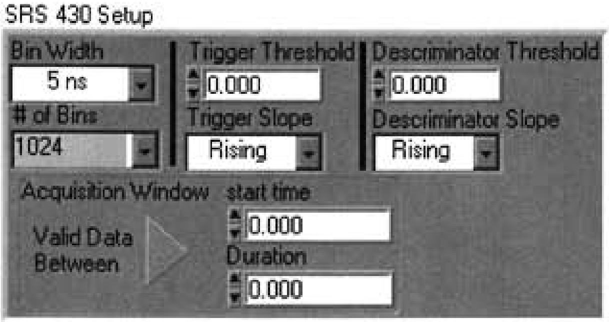


Figure 69. Custom Control used with SR430 drivers

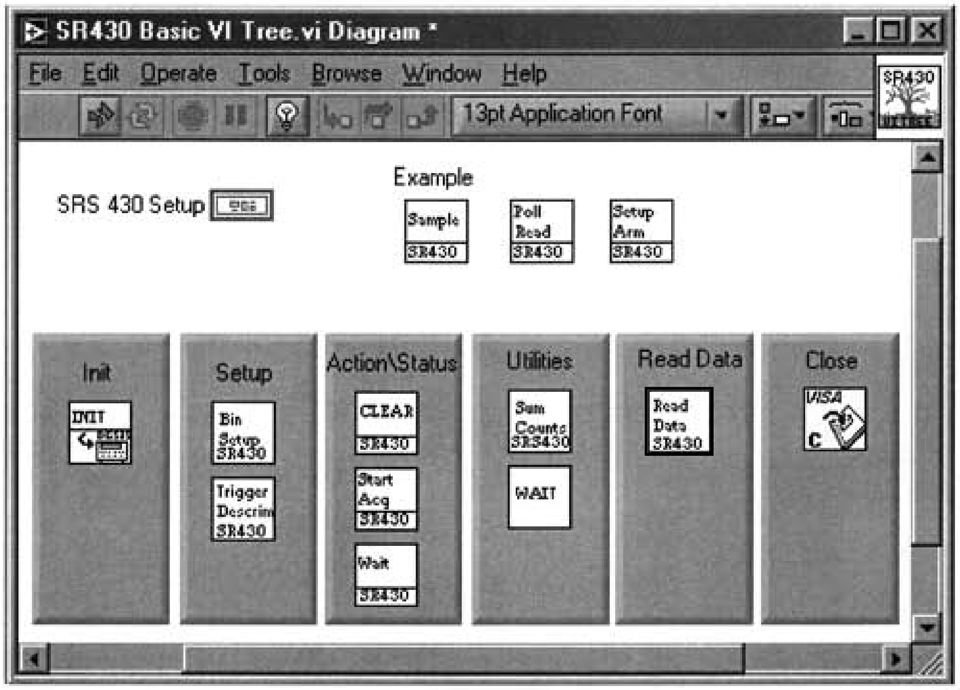


Figure 70. SR430 Function Tree

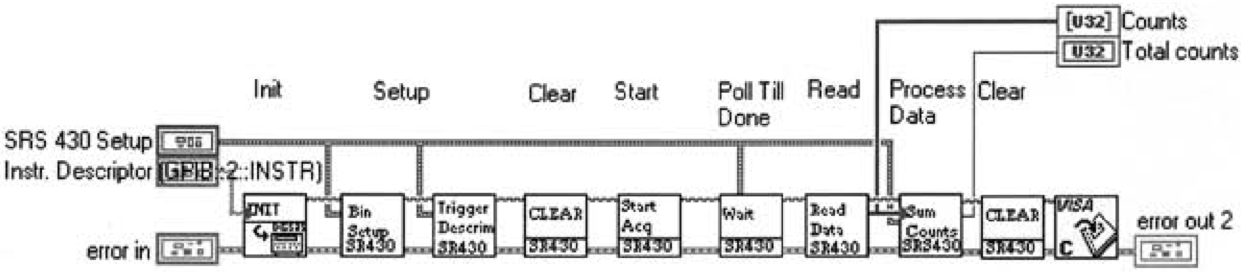


Figure 71. Sample SR430 Driver

3.3.6 SR PS3xx Power Supply Drivers

The Stanford Research power supplies are controlled by GPIB. Stanford Research using LabVIEW developed the drivers. All of the power supply drivers have 4 high level drivers (at the top of the VI Tree) that are used in the Power System top level software that have identical functionality and interfaces for all the power supplies. This includes an Initialize and Set, Set Voltage and current, Turn all Off, and Read Voltage and Current. These high level drivers in turn call lower level drivers as necessary to perform those basic tasks. This driver set also includes a getting started and a high-level application example.

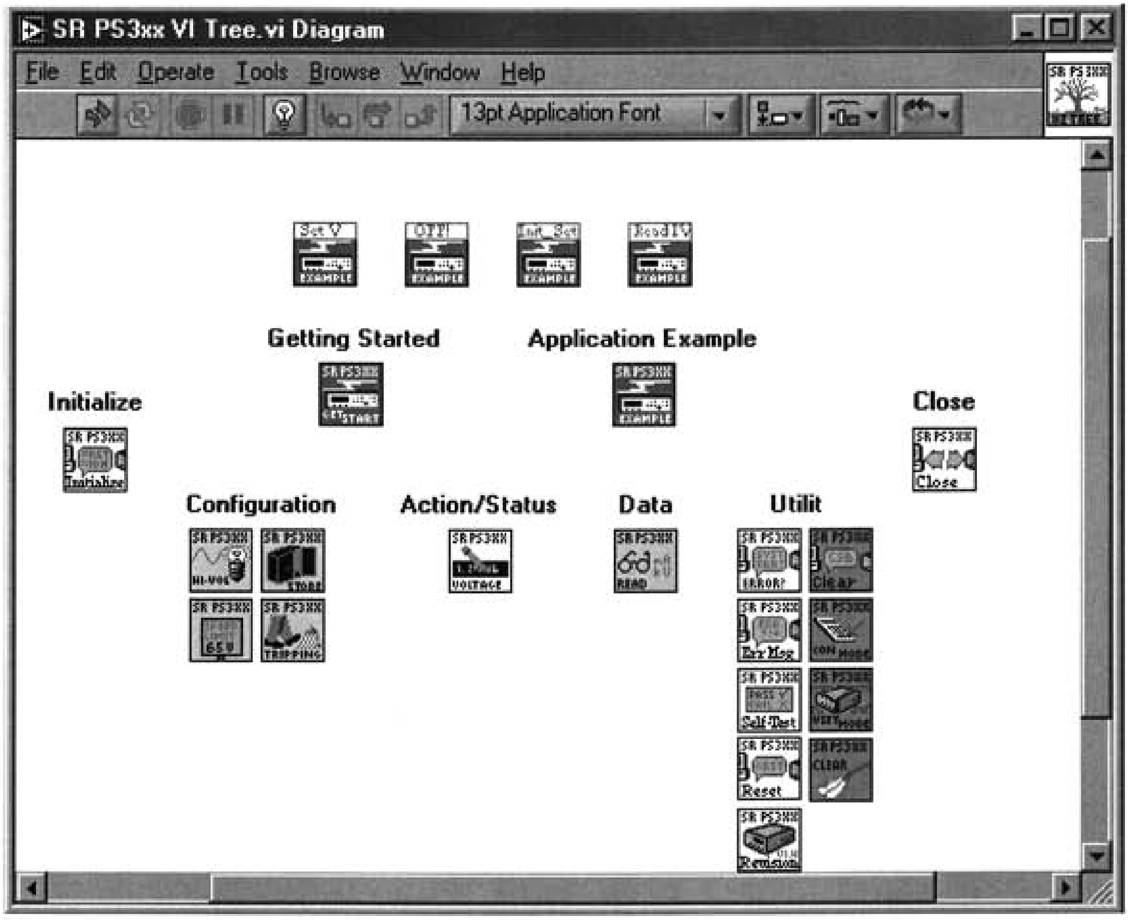


Figure 72. SR PS3xx Function Tree

3.3.7 TKTDS4xx Drivers

The TKTDS420 420 is controlled by GPIB. Tektronics using LabVIEW developed the drivers. The drivers are separated into 2 libraries, one for their TDS series in general, and another for the 4xx series. Additionally, several high level VIs were added for the Controlatron application.

3.3.7.1 High-Level Controlatron Drivers

Drivers found in Figure 70 perform high-level configuration and data retrieval. They use the typedef below as the data type to configure the scope.

3.3.7.2 TDS Setup.ctl

This is a type-def used by the high level drivers to configure a scope. Note: These drivers can be used to set up most of the common scope setting and triggering parameters.

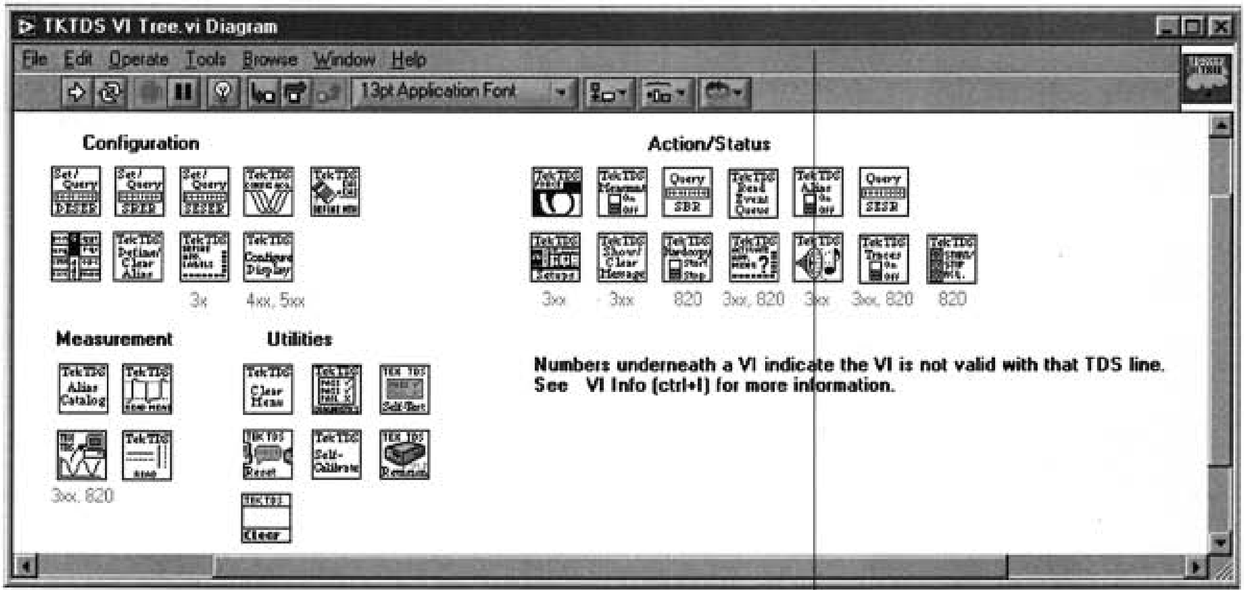


Figure 73. TKTDS General Command VI Tree

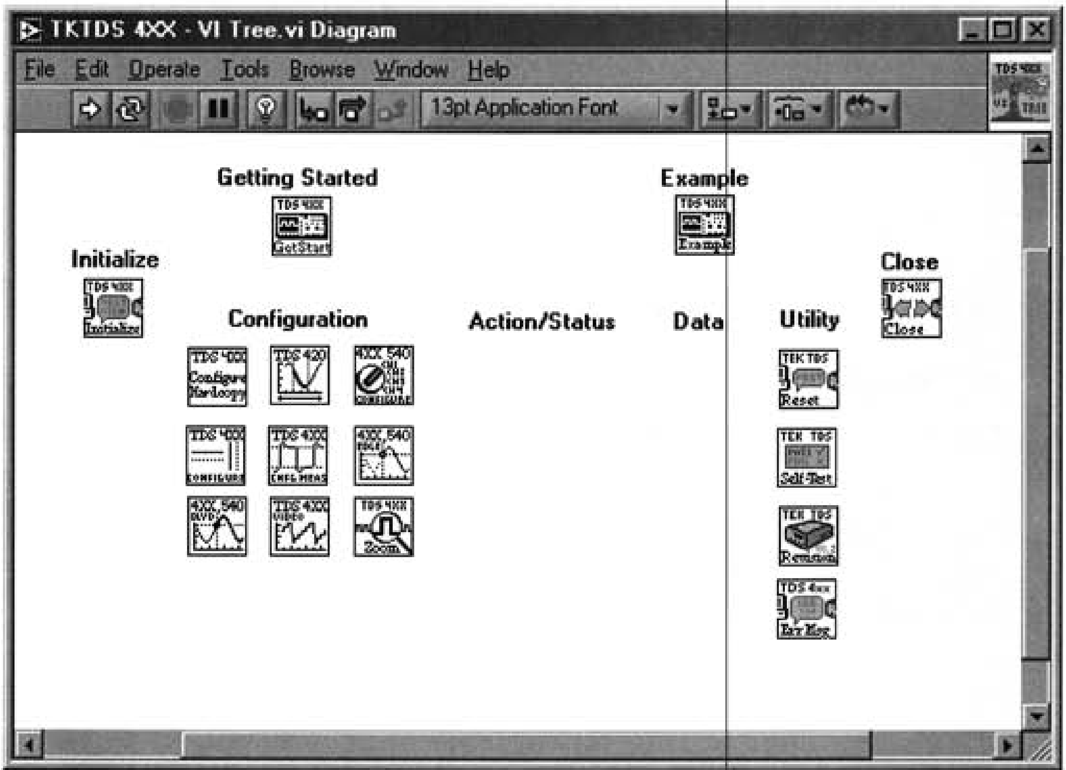


Figure 74. TKTDS 400 Series Specific Drivers

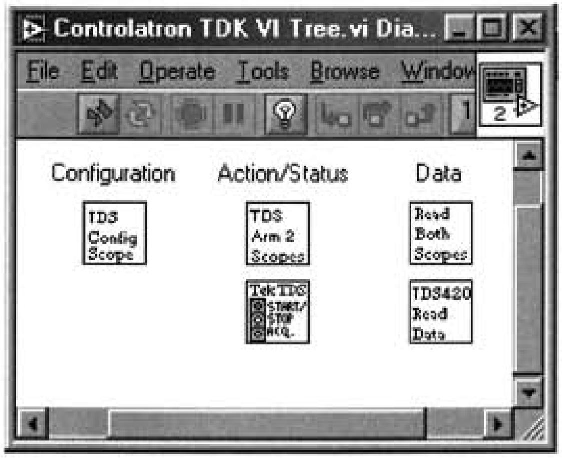


Figure 75. TKTDS 420 Controlatron High Level Drivers

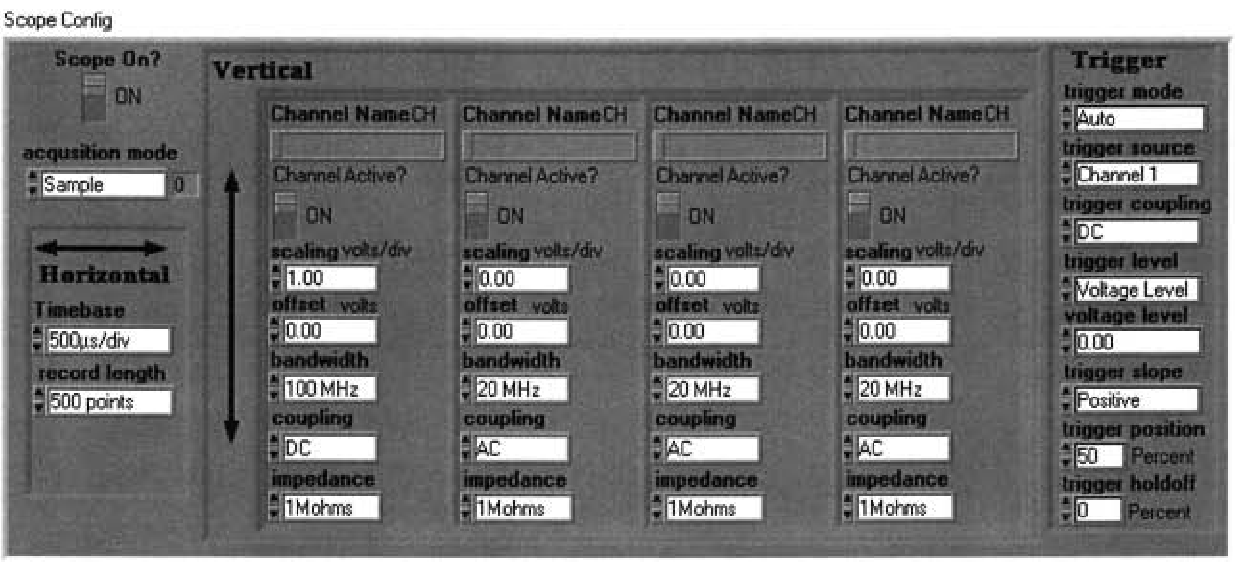


Figure 76. Scope Configuration Type Def Control

3.3.8 Xantrex Power Supply Drivers

The Xantrex power supplies are controlled by GPIB. Xantrex using LabVIEW developed the drivers. All of the power supply drivers have 4 high level drivers (at the top of the VI Tree) that are used in the Power System top level software that have identical functionality and interfaces for all the power supplies. This includes an Initialize and Set, Set Voltage and current, Turn all Off, and Read Voltage and Current. These high level drivers in turn call lower level drivers as necessary to perform those basic tasks. This driver set also includes a getting started and a high-level application example.

Controlatron Format Drivers

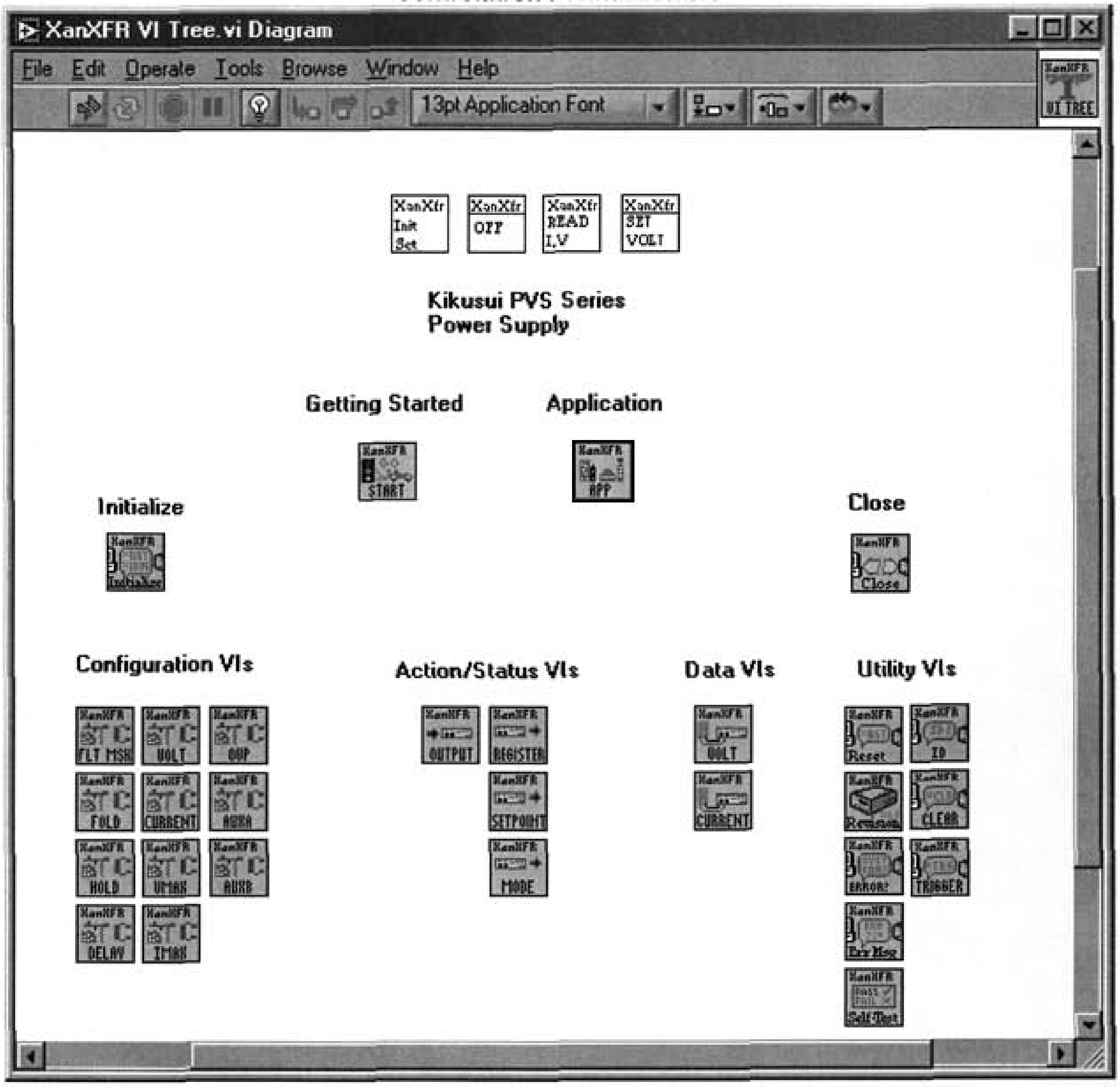


Figure 77. Xantrex Power Supply VI Tree

3.4 Message Manager Architecture Description and Tutorial

PrimeCore Systems utilizes one of 2 basic architectures for implementing GUIs and complex sub-programs: the queued state machine, and the message manager system. They are both similar in concept and very flexible. The major difference is that the queued state is primarily designed to be a single stand-alone program and the message manager is designed for running multiple programs that can all communicate with each other through a standardized interface. The Queued state is generally better for novice programmers and simple systems, while the message manager is targeted toward large complex systems and distributed systems. This document is intended as a reference guide to learn to troubleshoot and implement the Message Manager based programs in general terms. Open the VI "Sample User Interface.vi" which is used as the example for this tutorial.

3.4.1 What is a state machine?

In it's simplest terms a state machine is just a case structure inside a while loop. It is a handy way to piece together an application. There are often a pretty large number of cases, each responsible for taking a particular action. Additionally, there is always at least one case that runs when nothing is happening and checks to see if the program should do something. So the basic theory is that when nothing is pressed or nothing is going on, all the program does is checks to see if it should be doing something. If it needs to do something, the program figures out what it should do and calls up the case or cases responsible for doing that task and those cases then perform whatever needs to be done. In the example below, when the user presses the left button for instance, the program goes to the left case and moves the piece to the left if it can. This creates a simple yet very flexible architecture for programming virtually any user interface in LabVIEW.

3.4.2 What is a 'State'?

Each Case in the case structure encased in the while loop is considered a 'State'. Each State can be called to perform an action based upon a user request, managed in the 'Idle' State, or programmatically by any other available state. Each state has a unique name and should perform one logical group of functions. An example of a reasonable state is a 'Write Data File' state. This example state would manage all of the necessary functions to create a file, write data to the new file, and then close the systems resources used in the file creation. In the application, this one state could manage the entire data file writing requirements. An advantage to the developer is the inherent modularity in the architecture.

3.4.3 What is the Message Manager?

There are many ways to implement a state machine. The message manager serves as the message queue and communications portal for a state machine or a collection of state machines running in concert. There is a set of utility VIs called the message manager that handles all of this communication. On top of these utility VIs, the message manager is a philosophy and method of programming architecture that is centered around multiple state machine VIs running concurrently, passing data and actions to each other through the message manager utilities. Each separate state machine VI is also called a "module" since if you design the code well, then each major function or sub-system will have it's own "module" that can be reused and debugged by itself, independent of other modules.

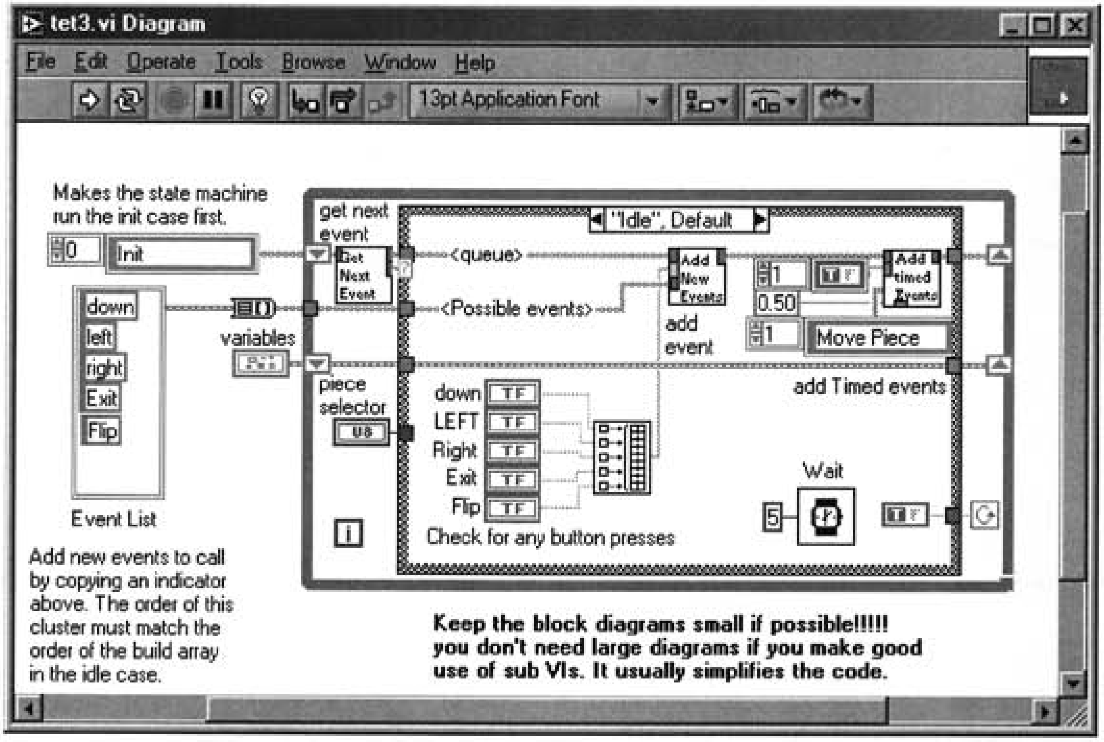


Figure 78. Example Queued State Machine 'Idle State'

3.4.3.1 The Key Elements to the Message Manager are:

- ***Modules:*** These are state machine programs that serve as user interfaces, drivers, or programs to control a system. These are developed as state machines by the programmer that utilize the message manager. These are the basically all the high level components of a LabVIEW application created using the message manager architecture.

  Figure 79 is the Idle Event of the CIC Top Level.vi found in the Controlatron Application. Notice the differences between this diagram and the Queued State Machine Diagram illustrated in Figure 78. There are events prior to executing the main 'While Loop' that are required to register this VI as a Module.

- ***Module Manager:*** This is a behind the scenes program that is the heart of the message manager. It contains queues for modules, initializes, or destroys queues.

- ***Register Module:*** This VI serves to initialize the message manager and to initialize the module that this VI is placed in. Normally, this is the first VI called in a module. This tells the module manager to allocate a queue for this module.

- ***Get Message.VI:*** Every module should use the Get Message.VI to run the state machine. It checks the queue for messages for that module and feeds a set of strings (the message) to the case structure of the state machine. It basically runs every state machine.

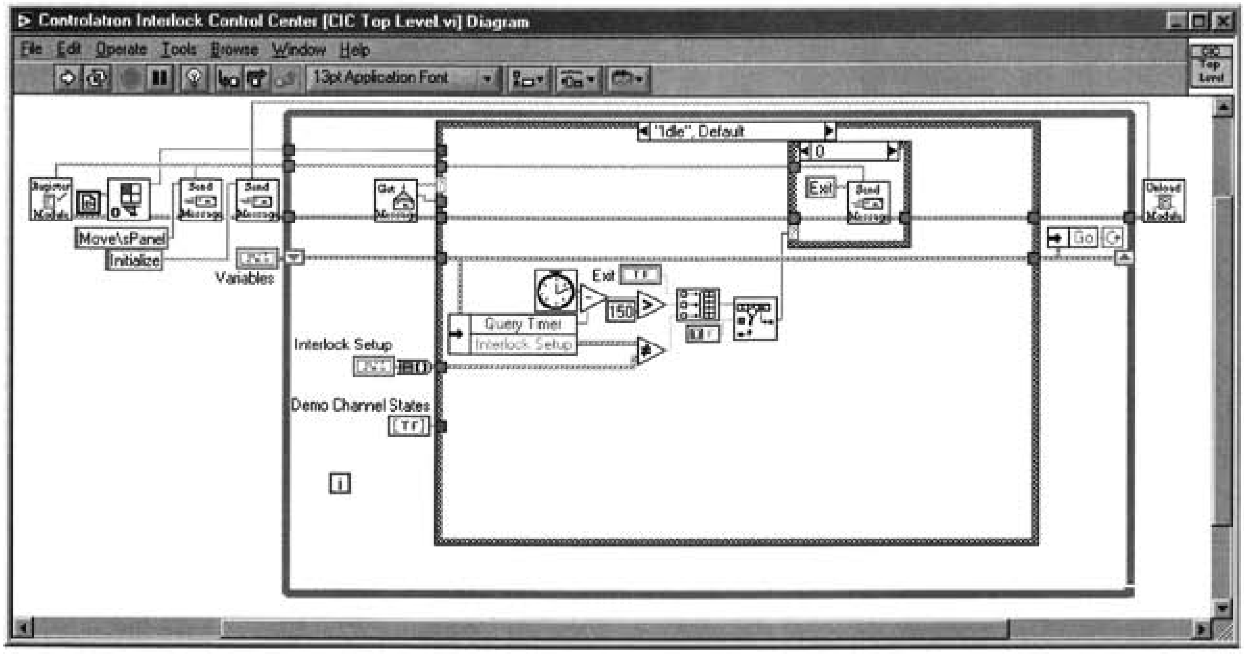


Figure 79. CIC Top Level Diagram (Message Manager Idle Event)

- ***Send Message.VI:*** This VI is used to send a message to a module that is retrieved later by the Get Message.VI. To use it, specify the module you want to send the message to (by name), the event (the case you want to run), and the event buffer (any data you want to send with the message).

3.4.3.2 What is a Message?

A message is a set of data that tells a module what state to go to and provides data for that state to run if necessary. In simple terms, the message Manager VIs use 3 strings to make up a message: the event, the event buffer, and the source. This elaborates on most state machines in that a string defines the state, and it also contains data to run with the state. This often proves to be more useful and flexible than just using a queue of states and transferring data and variables separately.

- ***Event:*** This is a string that tells a module what case (event) to run. It is output by the "get message.vi" and should be wired in to the case selector of the module. The event should match the name of a case in the module

- ***Event Buffer:*** This is a string that can contain data to be used in conjunction with the event. You can represent any type of data as a string, that is why that was selected. This allows you to transfer data between different modules. This is optional, you don't always need data, sometimes it is enough just to tell the module what event to go to.

- ***Source:*** This tells what module the message came from. This is useful for troubleshooting when you have multiple modules communicating with each other.

3.4.3.3 How do I add events to the message queue?

Since all top-level programs running in a message manager based system use the "get message.vi" to run the state machine, you must use the message manager VIs to control the actions of your program.

To send a message to run a case from inside the same module you can use the "send message.vi" or the "deliver message.vi". The key difference between these two is that the send messages puts the message at the end of the stack, and deliver message bumps the message to the top of the stack (forces it to be the next event run). If you are sending a message to the module you are in, then you don't need to wire in the destination module input.

3.4.3.4 Packaging and Un-packaging Data for a Message

Using the message manager, one of the most time consuming and difficult tasks is to convert the data to string for the event buffer data. Using the following strategies can help packaging the data and remain flexible and change tolerant.

For Simple data such as a Boolean, array, or numeric it is easiest to use the type cast function to pack data to a string for sending through the message manager.

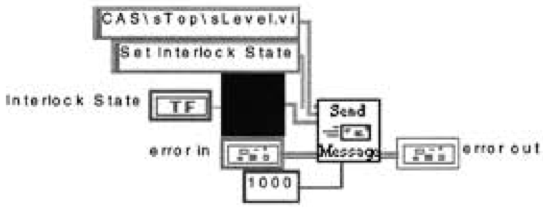


Figure 80. Packing Data For Message Management

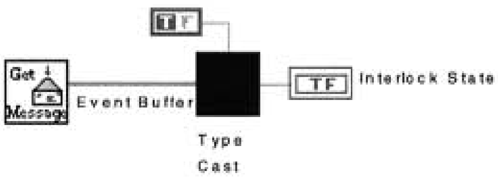


Figure 81. Un-Packing Data Received in a Message

For sending more complex data structures, it is usually easiest to package the data using the Flatten and Un-flatten from string functions. These will convert complex clusters or other data types to a string. The best way to do this is to use a cluster saved as a type-def control. Use the type-def control as the key to flatten and un-flatten the data as shown below. Using a type-def means that if you change the type-def data type, it will automatically update all the packaging and un-packaging functions you create for you and make the code more robust and tolerant to change.

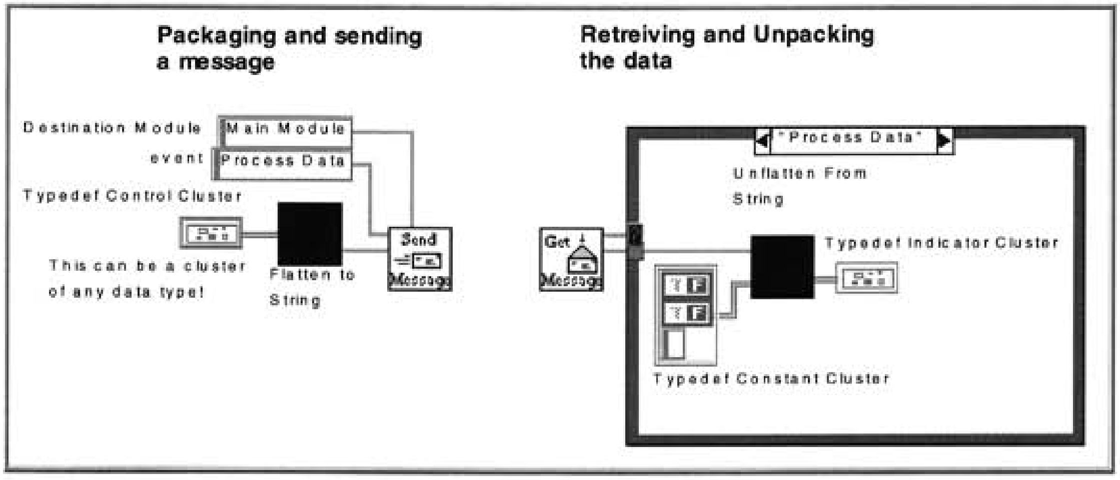


Figure 82. Packing and Un-Packing Data Summary

3.5 Overall Architecture of a Message Manager Based Program

The basic concept is to break up a large application into several modules that handle a particular task or sub-systems. That way you may be able to reuse modules and also have different programmers working on different modules so you can break up a project into smaller pieces easier. All the modules communicate to each other through the message manager.

Through the message manager, any module can communicate with any other module, but in general it is usually a good idea to have one or two high level modules that generally control the application and call a series of modules that perform the low level tasks of the system. If the system isn't very complex, then it is simplest to use a single high-level application to control the logic of the control system, and many modules to perform the sub-tasks and control sub-systems. If it is a large application, you may wish to create a more complex hierarchy and object-orient the design of the code. See the discussion on object orientation at the end of the tutorial.

3.6 Remote Message Manager

This basic architecture lends itself well to distributed applications running on different machines. The remote message manager just adds a layer to calls going to another computer. The remote message manager.vi is the engine that manages the remote connections. Once you establish a connection to another machine, then you can use the "send remote message.vi" instead of the send message.vi to communicate with a remote machine. Architecturally, it allows multiple programs running on different machines to communicate through a common interface. In future versions of the message manager it is planned to implement this more transparently without separate "remote" calls.

3.7 Advantages of a State Machine in General

- **Size:** It allows you to make a large program with a small block diagram.

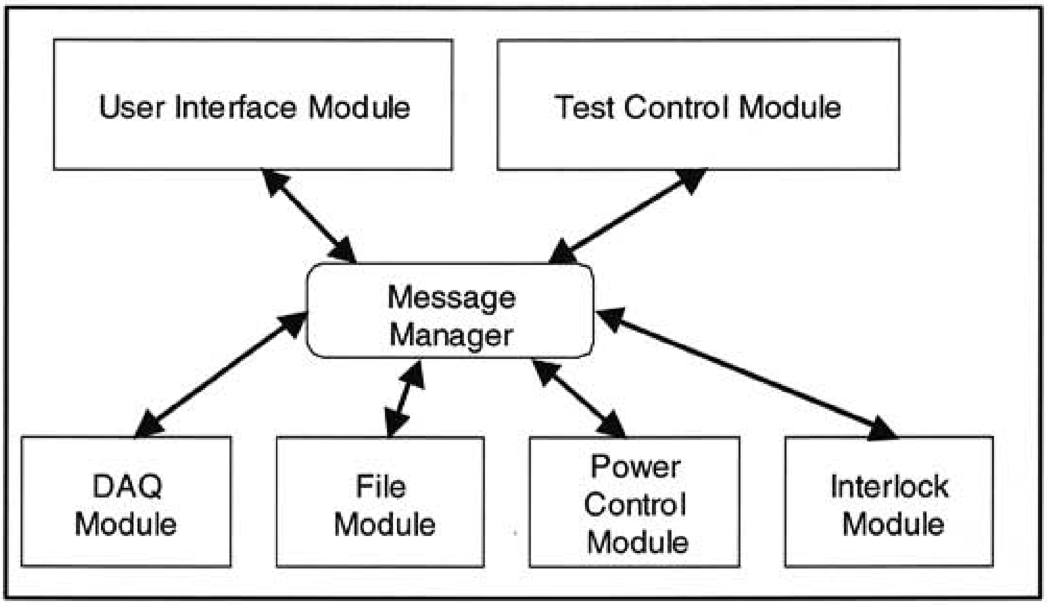


Figure 83. General Message Manager Architecture

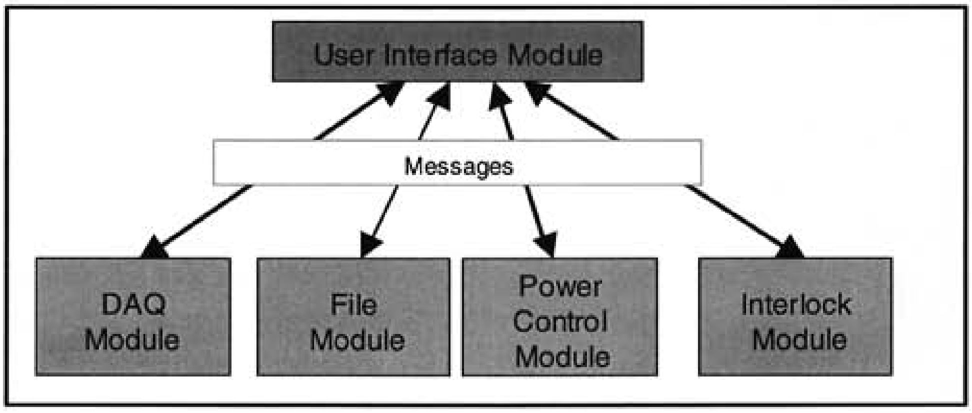


Figure 84. Module Responsibility Layout

- **Efficiency:** The state machine tends to efficient if implemented properly.

- **Troubleshooting:** Since every action takes place pretty much in it's own event (or case) it tends to be far easier to figure out what is going on than other architecture styles. Since the diagrams are smaller and more segmented, the light bulb and single step features work better than with large diagrams. They also tend to produce less "race conditions" by nature since everything is divided into small cases.

- **Standardization:** If every developer in a group uses it, it is much easier to debug and augment someone else's code.

- **Flexibility:** You can create any sequence of actions you desire with a state machine.

3.8 Advantages of the Message Manager

- **Standardization:** The best feature is that it is a flexible architecture that can be adapted to just about any program. It is a more intricate philosophy than a queued state machine.

- **Parallel Operation:** This allows you to run many sub systems together, all running in parallel with their own loop update rates.

- **Scalability and Reusability:** If you are careful in your top-level design, you can create modules that can be used in other programs with a minimum of rewriting code. Also, if you design your modules well, you can have stand-alone sub programs for sub-systems independent of your top level application, or you can use access functions to very quickly build new programs.

- **Common Data Portal:** All data that passes between different sub-programs uses the same interface. This reduces complexity and makes the code easier to understand.

3.9 Disadvantages of the Message Manager

- ***Speed of Object Oriented Code:*** Object oriented code requires more overhead in accessing data and the functionality of the system. It generally makes better code, but there are more layers to access data and function, thereby slowing down the code. The message manager is pretty well optimized, but if you are transferring large amounts of data, it can slow the system down.

- ***More difficult to troubleshoot than a simple state machine:*** It is harder to figure out what is going on with multiple modules running together than a single state machine. Design how the modules interact before integrating the modules together! Also, be sure to document how the system modules interact, otherwise it is very difficult to figure out what is going on a year down the road.

- ***More Up-Front coding time:*** It takes longer to write and get it working, especially if you use access functions and object-orient it. However, if you design it well, it should be easier down the road to troubleshoot and add functionality. Also, it should speed up projects with multiple developers since it is by nature modular. I do not recommend it for a simple DAQ application, but I highly recommend it for complex systems involving multiple instruments and\or functionality.

3.10 Object Orientation Using the Message Manager

The message manager architecture is well suited to making LabVIEW code more Object Oriented. You can't truly make LabVIEW code completely object oriented, but using the strategies below; you can accomplish most of the key properties of object-oriented code.

With the message manager architecture, the key to accomplishing object orientation is to first create modules for each major sub-system. The modules will be the "objects". The module should contain all of the functionality for the sub-system or user interface. If you are really gung ho about object orientation, you can even adapt the modules to create copies of themselves and refer to them by reference for using multiple objects that are the same. That is beyond the scope of this discussion, but certainly possible.

Additionally, all of the data involved with the sub-system should reside in the module (no globals or shortcuts) and should only be accessed through function calls to the module. If you need the data from a module in the user interface or another module, you should create an access function that reads it from the module itself and displays it through the user interface (or wherever). That way your data is protected and can only be accessed through proper channels. Also, write access functions for all commands, functions, setups, or whatever you need to command the module to do. Like the data encapsulation, it forces the programmer to go through proper channels to command the module to perform an action.

The sample uses both simple methods and some of the object-style access functions to call the modules so you can see the benefits of writing access functions for your modules.

So an object should consist of a set of VIs and type-def controls as follows.

- ***The Module VI:*** This is the message manager based state machine that contains all of the functionality code of the sub-system. It serves as the "object". It also contains all the data associated with that module, preferably in a shift register.

- ***Access Functions:*** These functions call the module through the message manager and perform a function or to return data. These functions encapsulate the module; so write access functions for ***everything*** you need to do with the module object. These are how to utilize a module if you want it object oriented. In other words, don't use the send message or package or un-package the data for another module in the diagram of a module, use access functions for everything.

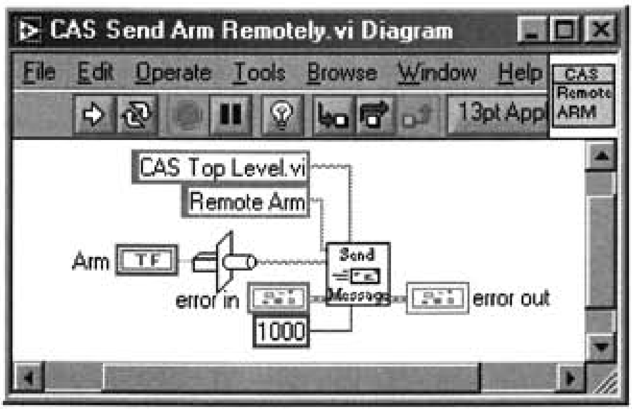


Figure 85. Example 'Access Function'

Figure 84 is a block diagram for a simple access function. It packages the data, tells it what case and module to go to so all the programmer needs to do is to drop this access function down and wire in the Arm Boolean. This helps encapsulate the functionality of the module.

- ***Type-Def Controls:*** Use type definitions or strict type-def controls for all complex data structures. That way your modules and access functions change together automatically if you have to change your data types. Use clusters and use unbundled and bundle by name wherever possible so that the code does not break when you add a variable. Try not to rename or delete variables unless necessary.

3.11 Queued State Machine Architecture Description and Tutorial

PrimeCore Systems utilizes one of 2 basic architectures for implementing GUIs and complex sub-programs: the queued state machine, and the message manager system. The more simple architecture used is the Queued State Machine. The Queued State Machine is very powerful in that it aids

3.11.1 First of all what is a state machine?

In it's simplest terms a state machine is just a case structure inside a while loop. There are often a pretty large number of cases, each responsible for taking a particular action. Additionally, there is always at least one case that runs when nothing is happening and checks to see if the program should do something. So the basic theory is that when nothing is pressed or nothing is going on, all the program does is checks to see if it should be doing something. If it needs to do something, the program figures out what it should do and calls up the case or cases responsible for doing that task and those cases then perform whatever needs to be done. In the example below, when the user presses the left button for instance, the program goes to the left case and moves the piece to the left if it can. This creates a simple yet very flexible architecture for programming virtually any user interface in LabVIEW.

3.11.2 What is a queued state machine?

There are many ways to implement a state machine; the queued state method utilizes a queue that is an array of strings to control the state machine. When an action is evoked, the state machine adds a string element to the queue array. The next time the loop cycles the string element is fed into the case structure using the Get Next Event VI. Using an array of strings allows for a whole sequence of events to be loaded up in the queue and run one after another for cascaded events.

3.11.3 How do I add events to the queue?

Add New Events and Add Periodic Events VIs can be used to add events to the queue. The input elements are a list of events (as an array of strings) that are potentially available and a Boolean array that tells the sub-VI weather each event should run or not (a true causes that event to be added to the queue). The array of strings just needs to match up element for element with the array of Booleans.

Other ways of taking action can include comparing a value of an indicator to its previous value. If they are not equal, the Boolean is true and an action should occur.

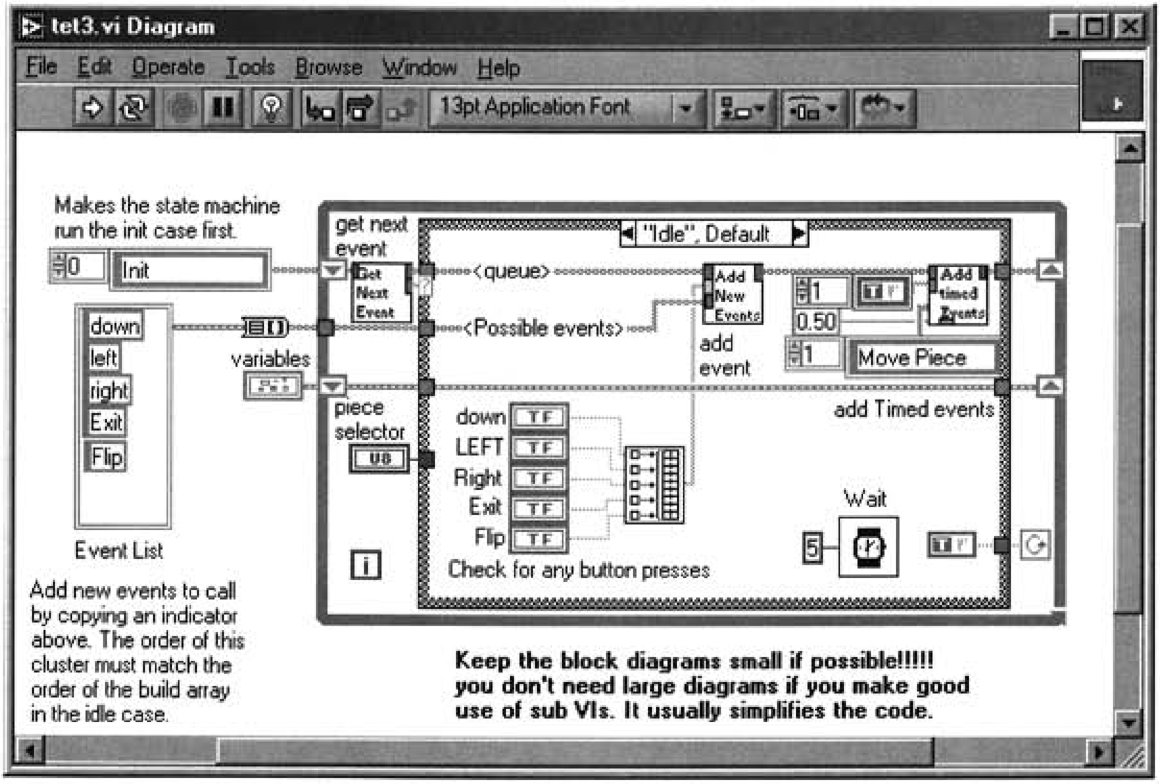


Figure 86. Adding Events to the 'Queue' of a Queued State Machine

Figure 85 shows an example of using a cluster and a shift register to contain variables. In the idle case it checks the value of the piece variable and if it changes it will feed a true to the add new events case and cause an action to happen. That way you can use the state machine to control events based on changing values or Booleans.

Also, if you want to call one case right after another you can use the build array function to call another event. That is useful for cascaded actions or for sequential operations.

The Build array in the upper right side causes the update piece case to be called right after the move piece case is called.

3.11.4 Timed Events

Using the Add Periodic Events VI, you can cause an action to occur every time interval. In this example the move piece case gets called every half a second to drop the piece. It works just like the Add New Events except that it only adds events when the timer goes off.

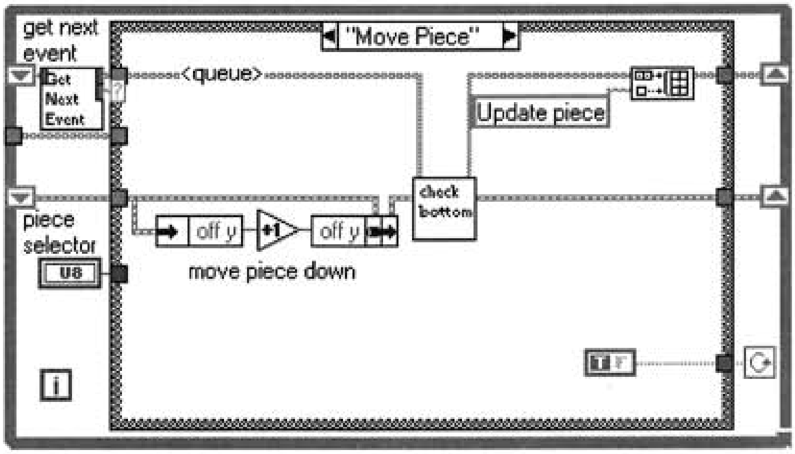


Figure 87. Programmatically adding events to the Queue

3.11.5 What can I do in a "State"

Everything that happens should pretty much have it's own case or state. You can update the display, send a command to an instrument, or read in information in a case.

3.11.6 Advantages of a State Machine

- **Size:** It allows you to make a large program with a small block diagram.
- **Efficiency:** The state machine tends to be efficient if implemented properly.
- **Troubleshooting:** Since every action takes place pretty much in it's own event (or case) it tends to be far easier to figure out what is going on than other architecture styles. Since the diagrams are smaller and more segmented, the light bulb and single step features work better than with large diagrams.
- **Standardization:** If every developer in a group uses it, it is much easier to debug and augment someone else's code.
- **Flexibility:** You can create any sequence of actions you desire with a state machine.

Distribution

6			Primecore Systems Inc. Atten: Robert Hertrich (2) Keith Barrett (2) Archives (2) 8421 Osuna Rd. Suite C-1 Albuquerque, New Mexico 87111
1	MS	0521	F.M. Bacon, 2502
1		0516	Terry Crow, 2564
1		0515	R.W. Howe, 2561
1		0516	R.J. Koss, 2564
1		0515	M.L. Martinez, 2561
10		0516	W.P. Noel, 2564
1		0516	D.L. Wallace, 2564
1		9018	Central Technical Files, 8945-1
2		0899	Technical Library, 9616
1		0612	Review and Approval Desk, 9612 For DOE/OSTI