AUTOMATED CALIBRATION PROCEDURES UTILIZING ACTIVEX TECHNOLOGY

Speaker: Kenneth Simpson
Authors: Kenneth Simpson, Michael Cunavelis
Naval Surface Warfare Center, Corona Division
simpsonkm@corona.navy.mil
Phone: (909) 273-5251; FAX (909) 273-5446

Abstract

The United States Navy is developing a specialized calibration environment using ActiveX technology to create XML calibration procedures that have embedded automation components. The ActiveX automation components allow for automated control of the test instruments and calibration standards within an XML calibration document. The ActiveX components are populated through database tables containing information pertaining to the test instrument being calibrated, the calibration standards utilized, and the test actions to be performed. Automatic equipment detection, procedure performance options, and results data printing, storage, and viewing are also facilitated through the use of additional specialized ActiveX components. Calibration procedures can be performed manually using the normal text instructions or in an automated mode using the ActiveX components. Utilizing reusable modular test methods and sequences stored in a database, the new ActiveX driven calibration environment facilitates rapid development of automated procedures coexisting within the framework of the new XML based manual environment.

1.0 BACKGROUND

The U.S. Navy calibrates an extensive variety of test equipment at over one hundred calibration laboratories to support worldwide fleet deployments. The calibration workload that can be supported using an automated environment includes items from many different vendors and ranges from general purpose commercial off-the-shelf (COTS) test equipment to specialized test sets which include Radar and Swept Frequency Measurement test sets. Due to the diverse workload requirements of the U.S. Navy, a robust automated calibration environment is required to efficiently perform automated calibrations using a database-driven tabular structure. The U.S. Navy is pursuing a calibration procedure development environment which stores all calibration procedure data in a database, as shown in Figure 1.

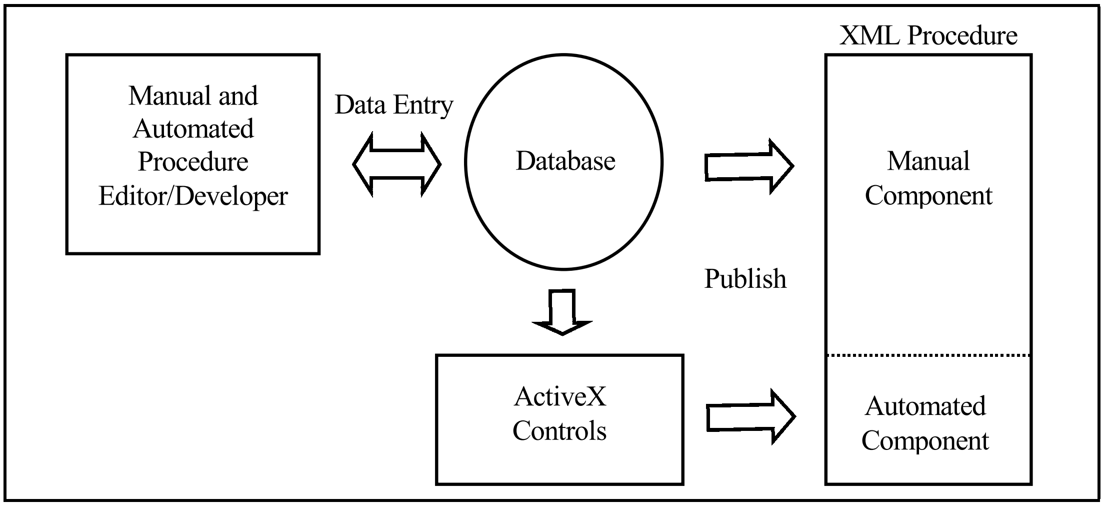


Figure 1. Calibration Development Environment

An automated XML (Extensible Markup Language) procedure with ActiveX controls can be constructed and published directly from this database, or alternately the XML procedure can be published without the ActiveX automation components and be executed manually. The database contains all the information necessary to publish automated and manual procedures including test instructions, calibration standards, calibration standards requirements, figures, and test tolerances. This concept of database procedure storage and retrieval will significantly reduce the development and maintenance cost of text-driven automated and manual calibration procedures.

In some circumstances the U.S. Navy still requires a manual procedure even when an approved automated procedure exists. Those circumstances include the requirement for troubleshooting and verifying an out-of-tolerance condition, providing back-up when automation does not support calibration standards substitution or when the automated system is inoperable.

ActiveX automation components were developed to handle all of the necessary tasks associated with performing automated calibrations. Equipment detection, procedure execution options, the automated testing itself, and the display of the results obtained are all handled through ActiveX controls embedded into an XML procedure.

Having an integrated automated and manual procedure execution environment is required to provide cost effective calibration support for the wide variety of U.S. Navy test and measurement equipment.

This integration of automation within the framework of a calibration procedure in the XML format using ActiveX technology is now referred to in the U.S. Navy as ActiveCal.

2.0 Calibration Procedure Overview

The new XML procedure design is structured to facilitate data storage and retrieval from a database by organizing all the procedure data within modules. Figure 2 provides an overview of the procedure and module structure.

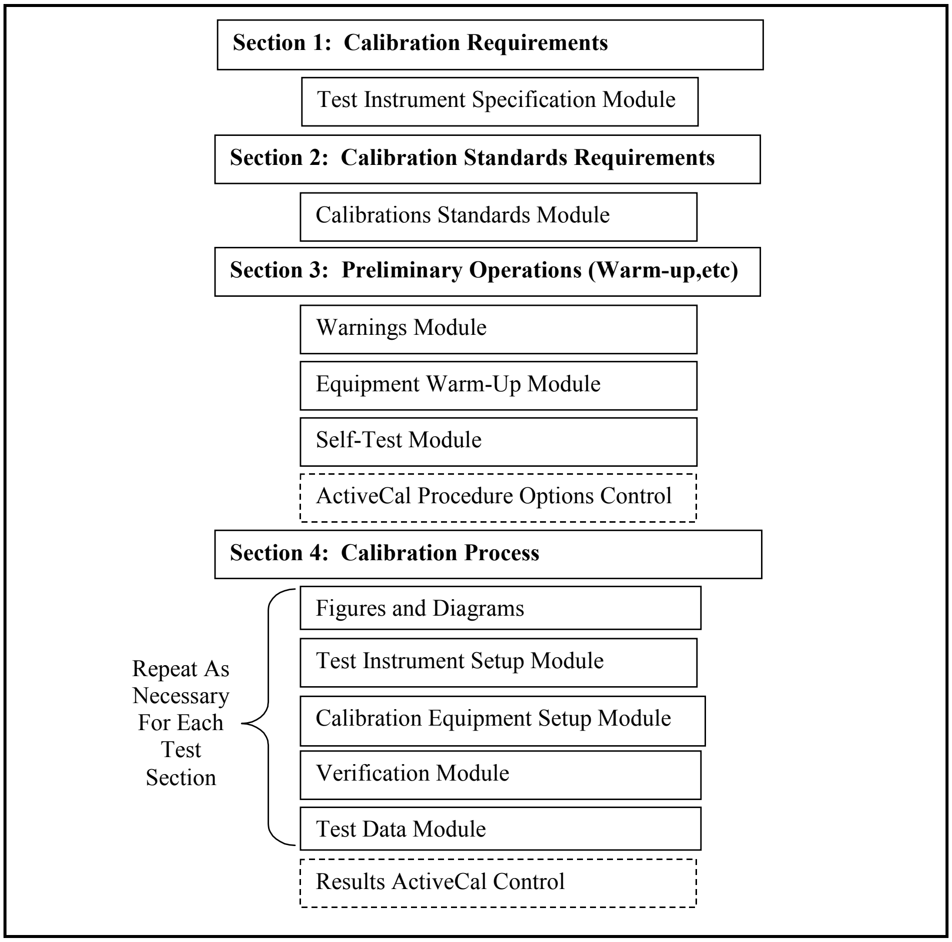


Figure 2. Calibration Procedure Format

The automated and manual procedures share all data except for the setup and verification modules, which are replaced by an ActiveX control in the automated procedure (Test/Results ActiveCal Control). An ActiveX control is used in Section 3 of the procedure for initial equipment detection and to set procedure execution options (ActiveCal Procedure Options Control). Another ActiveX control is used at the end of Section 4 to facilitate accessing the test results obtained (Results ActiveCal Control). Both the automated and manual procedures are XML documents viewed in a Microsoft Internet Explorer browser window.

2.1 Preliminary Operations Module

The Preliminary Operation module is designed to provide the initial information necessary to start the procedure. Initial warnings, cautions, and notes, along with initial turn-on information for the test instrument and the calibration standards. This includes any and all information pertinent to getting the calibration process up and running in a particular procedure. For an automated procedure, it was recognized that initial operations were necessary to detect the test instrument and the calibration standards to be able to ensure their presence and appropriate IEEE bus/address information. Additionally, execution options pertaining to an automated execution are also present to set the flow of the procedure.

2.1.1 ActiveCal Procedure Options Control

The ActiveCal Procedure Options Control, displayed in Figure 3, shows the available options the user has available to both scan for equipment present on the IEEE bus and the run-time execution options of the procedure.

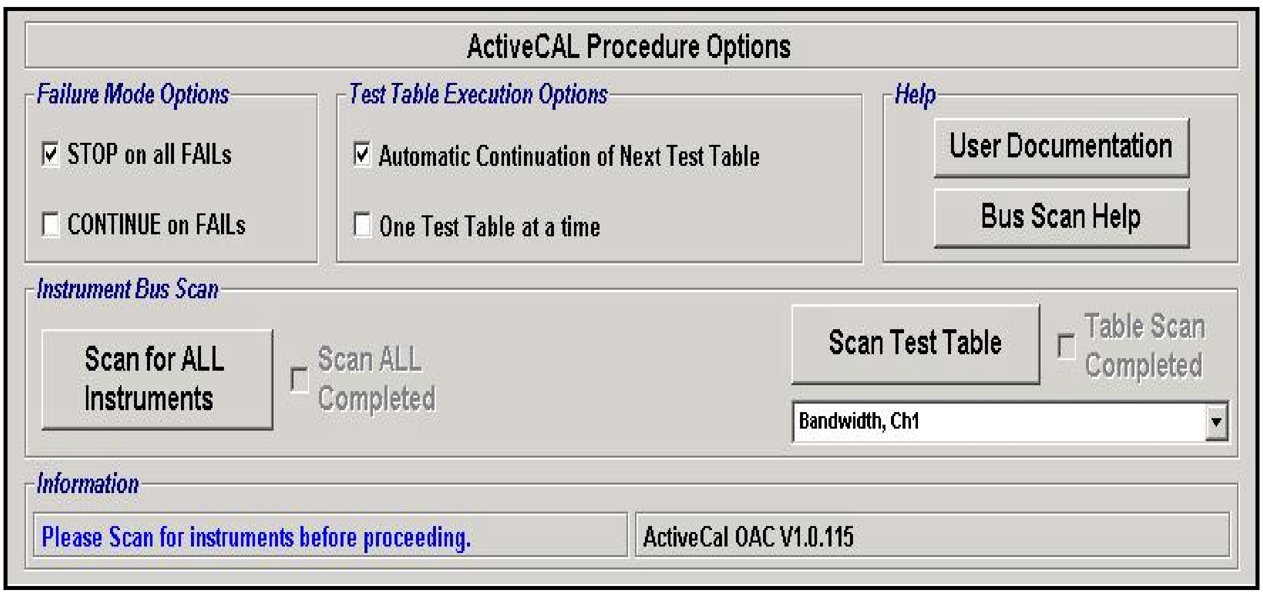


Figure 3. ActiveCal Procedure Options Control

There are two Failure Mode options available to the user. By default, the test execution will stop on any fail condition detected and await user input to proceed, re-run the test that failed, or to abort the test. Alternatively, an option is available to continue on any failures detected.

Also, there are two Test Table Execution options available. By default, the procedure will try to execute in as automated a fashion as possible. If it is possible to continue from one test table to the next, the execution flow will continue until the end of the procedure or until user interaction is required. Alternatively, for troubleshooting purposes one test table can be executed to allow for testing of any particular test table in the procedure. Any options set are saved to a temporary database for the Test/Results ActiveCal Controls in the main body of the procedure to access and take the appropriate action.

To ensure proper communication with the equipment and to ensure the bus cable connections are valid, an Instrument Bus Scan feature is available. The Bus Scan feature will detect the calibration standards and test instrument available on the IEEE bus(es). Also, the current IEEE Bus and Address of the equipment is detected and stored for use in communicating with the equipment during the automated calibration process.

Figure 4, Address Scanner – Required Devices Display shows the equipment required to be present on the IEEE bus for detection. After initiating a scan, Figure 5, Address Scanner – Identification Display shows the results from the scan: the port (Test Instrument or Standard) where the equipment was found, and the IEEE address of the detected equipment. This information about the port and the address of each item is stored in a temporary database for use by the Test/Results ActiveCal Controls in the main body of the procedure for communication with the equipment.

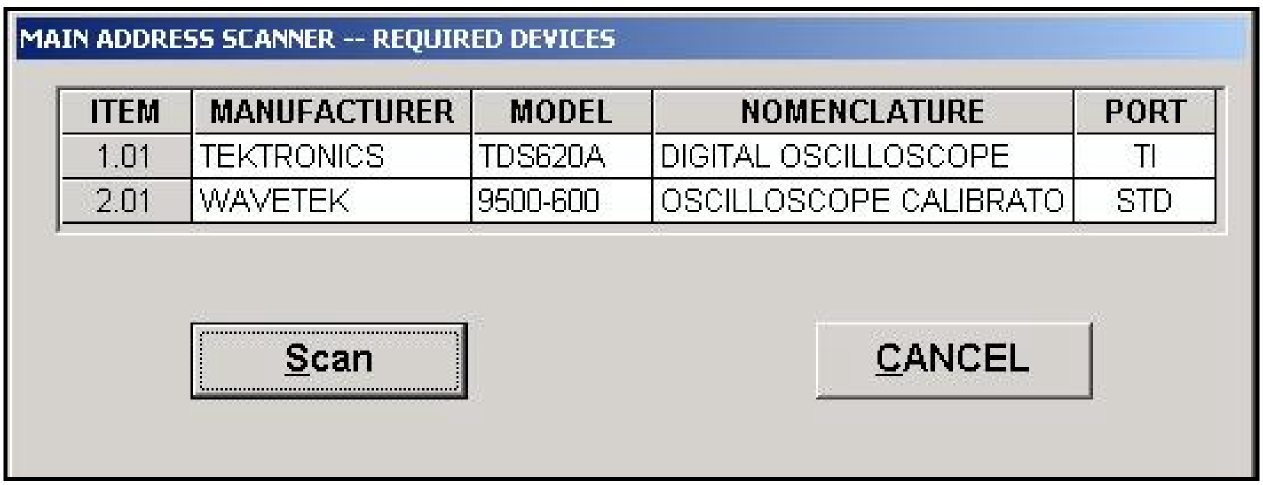


Figure 4. Address Scanner – Required Devices Display

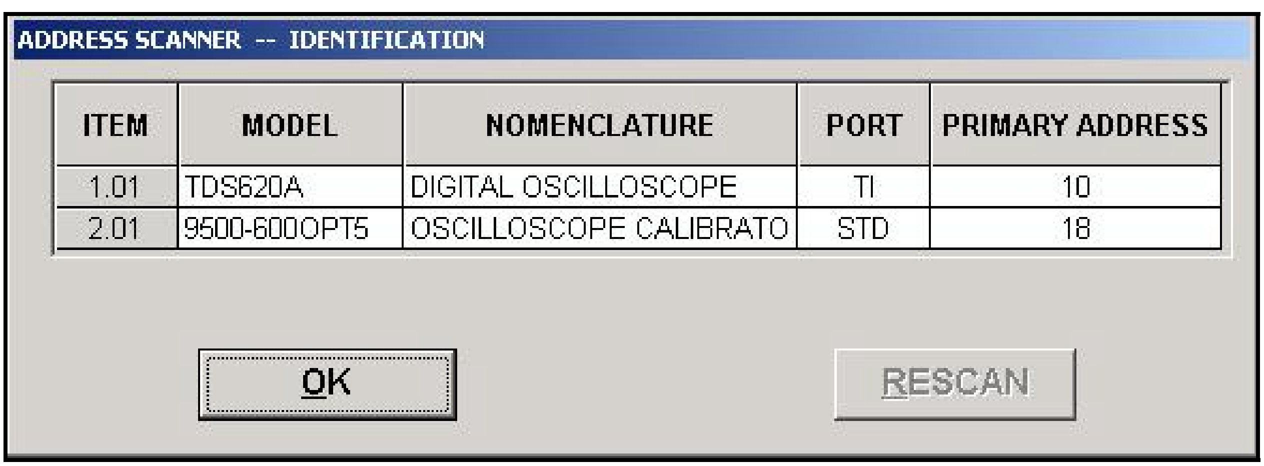


Figure 5. Address Scanner – Identification Display

Additionally, help is available for the ActiveCal automation components with links to the user documentation on the ActiveCal technology and specific help information is available related to Bus Scan problems.

2.2 Automated Procedure Setup and Verification Modules

Figure 6 shows the manual and automated procedure structures side by side to illustrate the common procedure data and structure of the two components. The Test/Results ActiveCal Control is essentially a self-contained test sequence consisting of all the bus-commands, test-tolerances, and user-interaction required to perform an individual test, i.e. vertical amplitude accuracy. These automated sequences in the automated procedure follow the setup and verification steps that would be performed in the manual procedure. The automated sequences processed by the Test/Results ActiveCal Control are developed and stored in a database using an automated procedure editor.

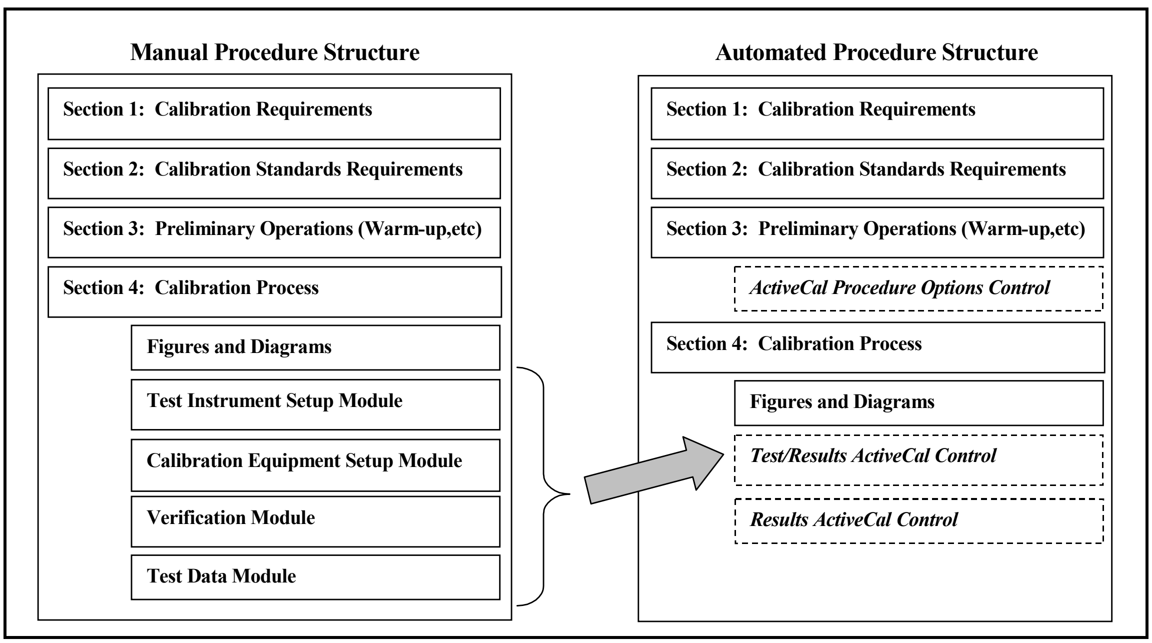
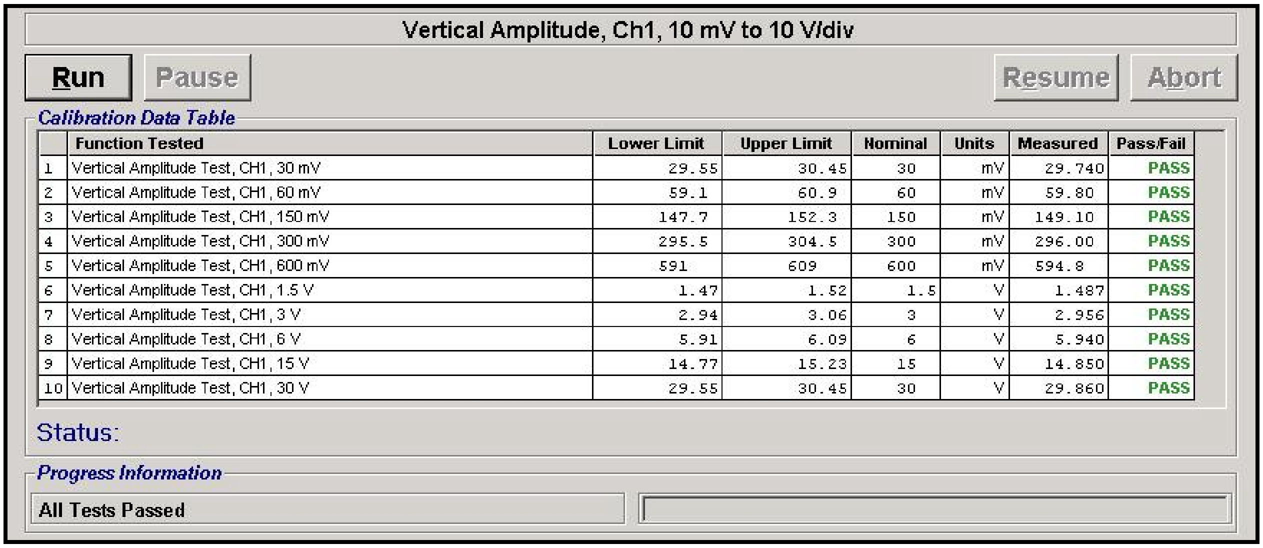


Figure 6. Manual And Automated Procedure Structure Comparison

2.2.1 Test/Results ActiveCal Control

The Test/Results ActiveCal Control shown in Figure 7 is used in the automated procedure in place of each Setup, Verification, and Test Data module group normally found in a manual procedure. The control shown indicates the measured data taken and that the values measured were within the tolerance limits by displaying a PASS indication.



	Function Tested	Lower Limit	Upper Limit	Nominal	Units	Measured	Pass/Fail
1	Vertical Amplitude Test, CH1, 30 mV	29.55	30.45	30	mV	29.740	PASS
2	Vertical Amplitude Test, CH1, 60 mV	59.1	60.9	60	mV	59.80	PASS
3	Vertical Amplitude Test, CH1, 150 mV	147.7	152.3	150	mV	149.10	PASS
4	Vertical Amplitude Test, CH1, 300 mV	295.5	304.5	300	mV	296.00	PASS
5	Vertical Amplitude Test, CH1, 600 mV	591	609	600	mV	594.8	PASS
6	Vertical Amplitude Test, CH1, 1.5 V	1.47	1.52	1.5	V	1.487	PASS
7	Vertical Amplitude Test, CH1, 3 V	2.94	3.06	3	V	2.956	PASS
8	Vertical Amplitude Test, CH1, 6 V	5.91	6.09	6	V	5.940	PASS
9	Vertical Amplitude Test, CH1, 15 V	14.77	15.23	15	V	14.850	PASS
10	Vertical Amplitude Test, CH1, 30 V	29.55	30.45	30	V	29.860	PASS

Figure 7. Test/Results ActiveCal Control

The number of rows in the Test/Results ActiveCal Control is dependent upon the number of tests to be performed to validate the parameter being tested. The control will autosize to the number of tests present.

The Test/Results ActiveCal Control has several capabilities. One is the pausing and resumption of testing (see Figure 8). Also, the aborting of testing is available to terminate the testing sequence. If failures are detected, an appropriate note will be displayed to the user and the user will have the option of repeating the failed test or resume and proceed with the next test (see Figure 9). This is only available if the Stop on All Fails option is selected in the ActiveCal Procedure Options Control. Additionally, status messages letting the user know the progress of testing using displayed text messages is available. An indication of the progress based upon the number of tests performed using a standard progress bar is used. Finally, a Pass/Fail message for the entire test is generated at the end of the test execution for the table indicating whether all tests passed or the number of failed tests.

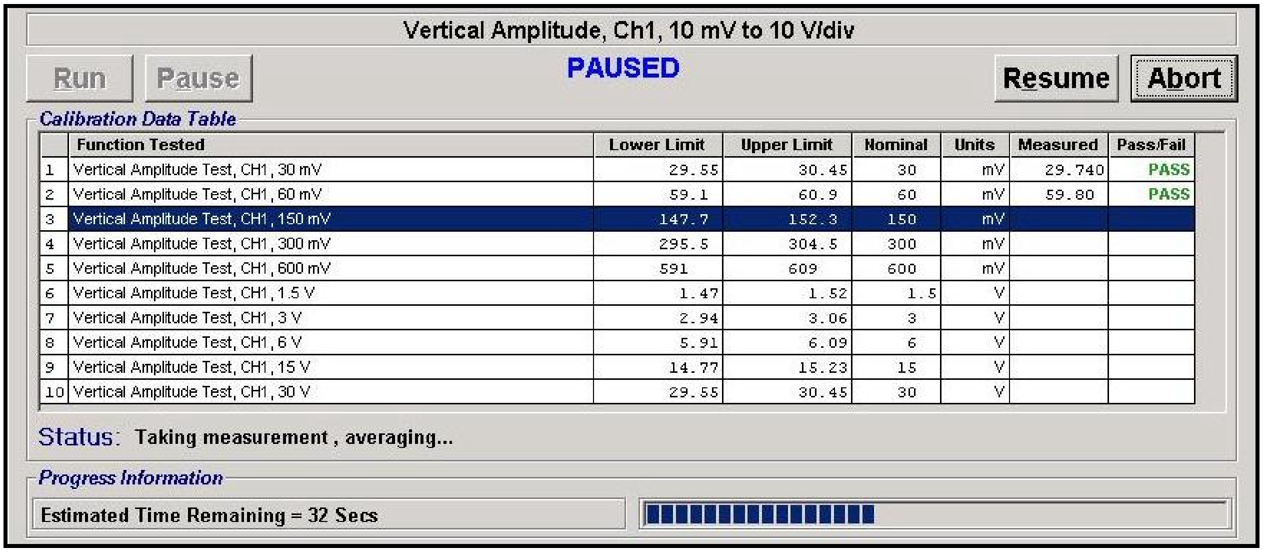


Figure 8. Test/Results ActiveCal Control - Paused

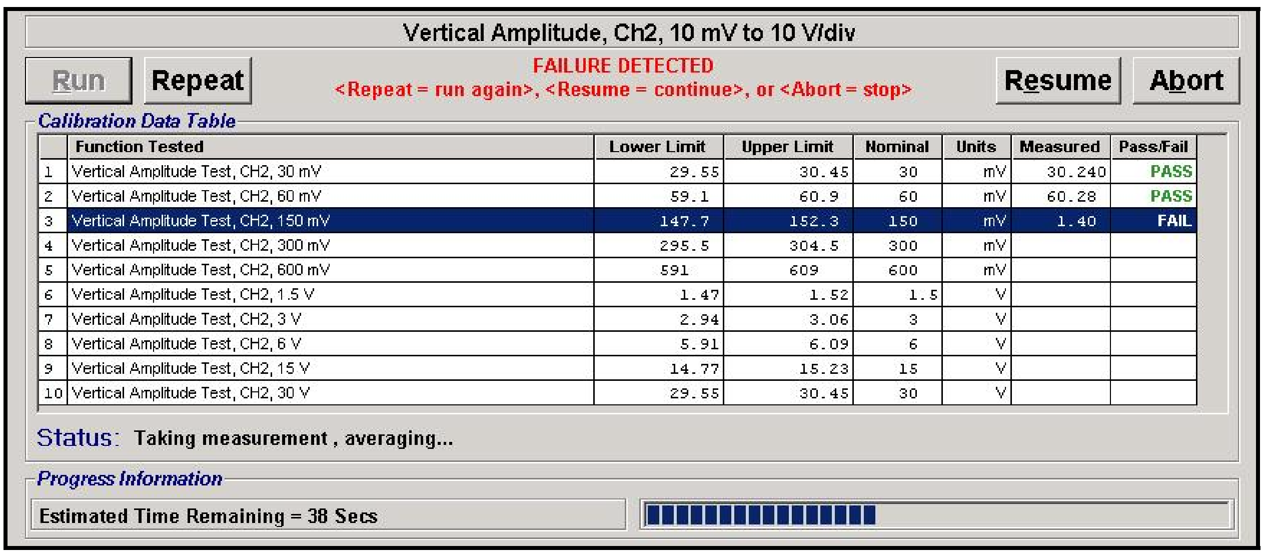


Figure 9. Test/Results ActiveCal Control – Fail Detected

The Test/Results ActiveCal Control has various interface methods for communicating with the user to display messages or to get data from the user. A status line is used to display the current status of the testing, i.e. what operation is currently being performed by the automation code. This status line is under the data table in the Test/Results ActiveCal Control. Furthermore, a note area is available to display messages that may have to remain for an extended period of time, i.e. warnings about the presence of high voltage during the tests. The note area is located between the Pause (Repeat) button and the Resume button.

Also, via system mnemonic commands, messages using popup boxes can be displayed to the user, the user can be asked to input data manually, and there is capability for the user to adjust an equipment parameter through automation code to look for a certain indication.

Further functionality for the Test/Results ActiveCal control (and the other controls as well) is planned to be added through the use of an external DLL library, if required. This DLL library may contain special functions related to a single instrument, thus not requiring the ActiveCal controls to be updated for specific models.

2.2.2 Results ActiveCal Control

The Results ActiveCal Control in Figure 10 provides the user with three options: print the results, save the results to a file, and view the results on-screen. Furthermore, an option is available to display all results or only those results where a failure occurred during testing.

Future implementation plans call for the test data generated to be collected after the execution of each procedure to a centralized database for eventual analysis to support intervals, failure modes, and other applications where test result data is required.

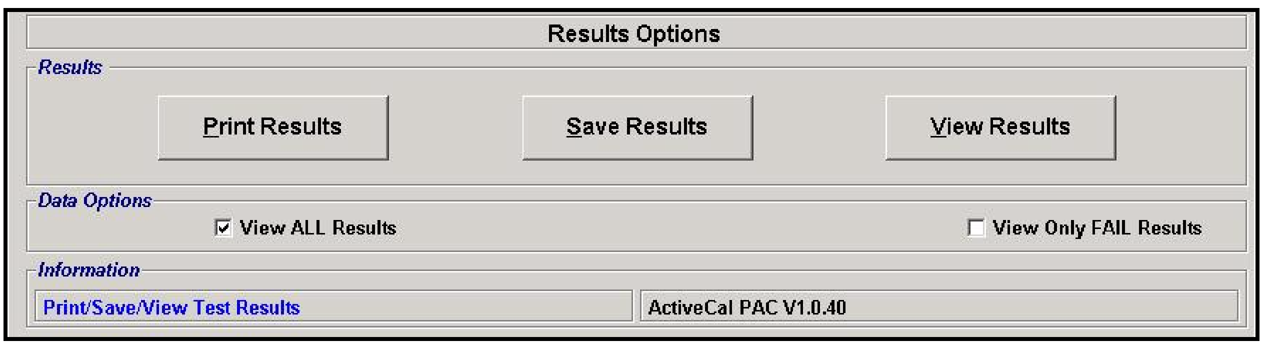


Figure 10. Results ActiveCal Control

3.0 Automated Procedure Development Environment

3.1 Automated Procedure Editor

To populate the Test/Results control, an automated procedure editor was developed to build the test sections and test sequences necessary to perform automated testing of test instrument parameters. Figure 11 shows the overall automated procedure development environment and the interaction of the procedure editor program with the data tables and the Test/Results ActiveCal Control. The procedure developer uses the procedure editor program to develop test sequences using standardized mnemonics to specify all procedure actions. These mnemonics are cross-referenced either to an instrument bus command or to a system subroutine to perform a specific task. The system subroutine tasks can vary from a simple wait statement to storing a measured value for later computation or comparison.

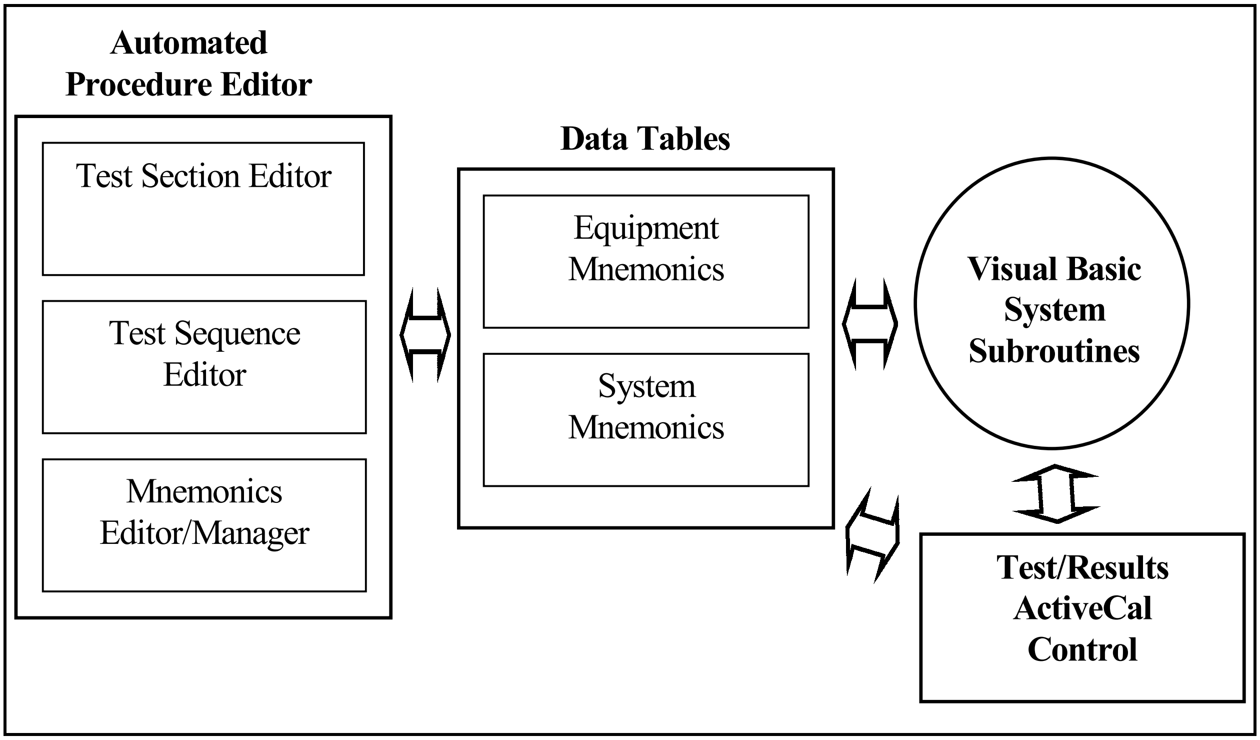


Figure 11. Automated Procedure Development Environment

The automated procedure editor is a program used by non-programmers to populate and test the Test/Results ActiveCal Control in each test section of the automated procedure. The procedure editor develops test sequences that track the module structure of the manual component of the procedure. The procedure editor performs the following functions: test section editor, test sequence editor and mnemonics editor/manager.

Figure 12 shows a screenshot of the procedure editor showing both the Test Section Editor (Labeled #1 and #2) and the Test Sequence Editor (labeled #3).

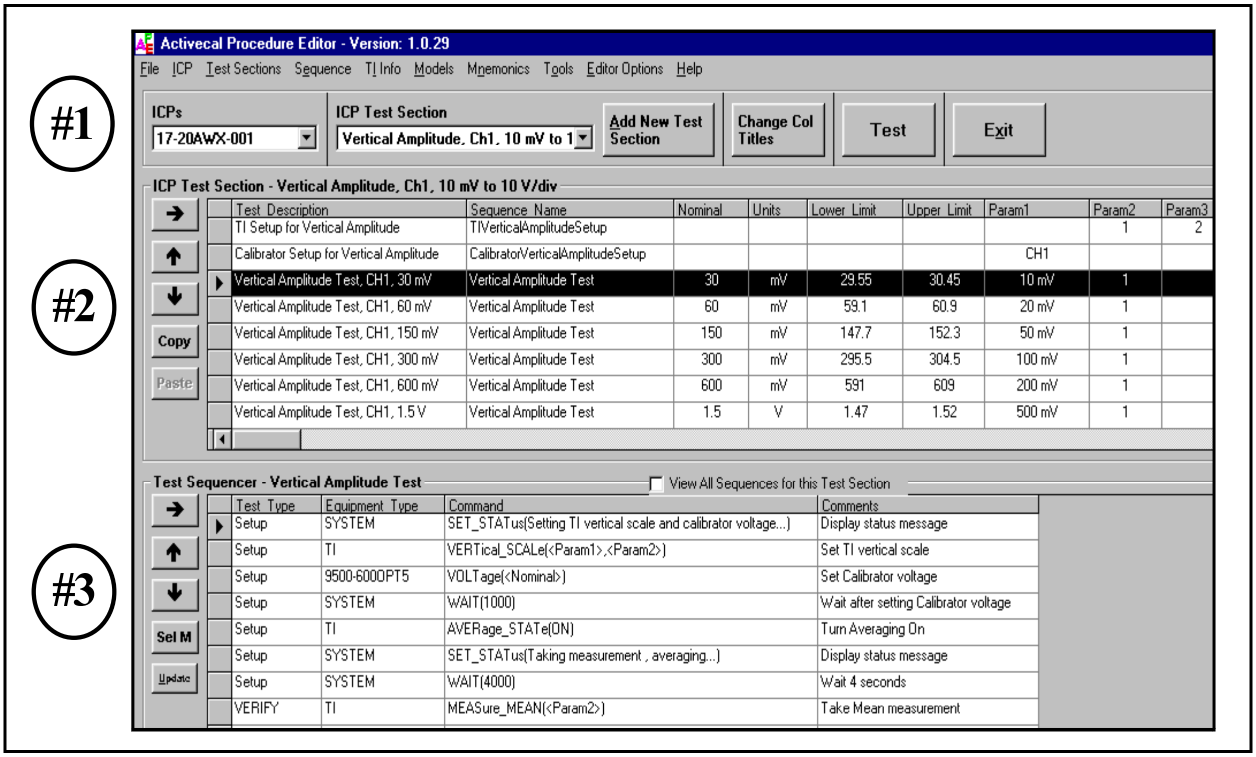


Figure 12. Test Section Editor and Test Sequence Editor Screenshot

3.1.1 Test Section Editor

The test section editor performs three main functions: manage and arrange the test sequences, manage parameter data passed to the test sequences, and edit the test tolerances. The test section editor works in conjunction with the test sequence editor to develop all the test sequences used in the procedure.

The top part of the test section editor, labeled as #1 in Figure 12, allows for the selection of the Instrument Calibration Procedure (ICP), the selection of the ICP Test Section to edit, and the addition of new tests. The bottom part of the test section editor, labeled #2, is the sequence manager. The sequence manager controls the order of the test sequences, specifies the sequence for each test, enters the test tolerances, and passes parameter variables to the test sequencer. The flow of the sequence manager is designed to emulate the structure of the manual procedure.

3.1.2 Test Sequence Editor

The test sequence editor, labeled as #3 in the Figure 12 screenshot, displays and edits the active sequence. The active sequence in the example is called Vertical Amplitude Test. The sequences are a series of commands which consists of mnemonics to perform an action.

3.1.3 Mnemonics/Editor Manager

The mnemonics editor/manager lets the procedure developer add, delete, and/or update mnemonic information including the cross-referenced bus commands.

3.2 Navy Mnemonic Structure

The environment mnemonic structure of the ActiveCal technology is based on the Standard Commands for Programmable Instruments (SCPI)[(1)] command architecture. SCPI was developed for programming consistency between instruments of the same class and functional capability. For example, SCPI has a common command set for electronic counters to measure frequency, regardless of the counter model or manufacturer. In addition, an oscilloscope can use the same frequency command as the counter to measure frequency.

The U.S. Navy programming environment uses standardized mnemonics that emulate the SCPI command structure to achieve the SCPI goal of consistent commands for the same equipment class or functionality. This is achieved by using an equipment look-up table which contains the U.S. Navy mnemonic referenced to the bus command of a specific model. The environment also uses standardized mnemonics for system commands which are not directly related to a specific instrument.

The procedure developer selects the mnemonic while developing a sequence in the test editor. Figure 13 shows the relationship between the mnemonic entered by the procedure developer in the test editor with the database tables for the equipment and system mnemonics.

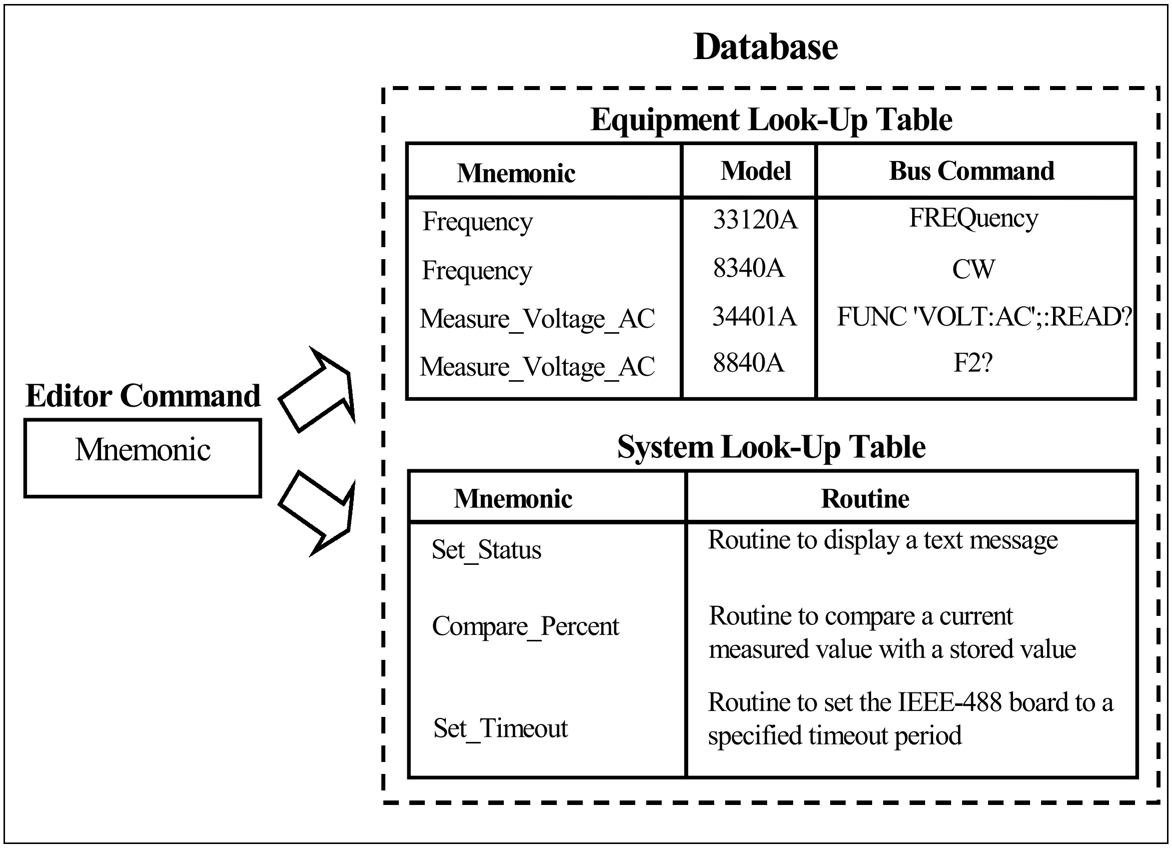


Figure 13. Mnemonic to Database Architecture

This common mnemonic structure provides several key benefits. The procedure developer does not have to work with a multitude of instrument bus commands, thereby simplifying and streamlining the procedure development. Equipment substitution is built-in by utilizing mnemonics which apply to the same instrument class or for instruments that have similar functionality.

The mnemonic command structure lends itself to the sequence driven procedure environment for re-useable standardized sequences and measurement methods. The mnemonic/sequence structure facilitates quicker procedure development while maintaining a standardized measurement method library in a database.

3.3 Development Testing Capability

The Automated Procedure Editor has a test mode that lets the procedure developer test the current test section. This mode launches the Test/Results ActiveCal Control with additional developer options to allow for sequence-by-sequence testing to examine the Mnemonic to IEEE bus command information, database re-initialization, equipment setup functions and other test execution options. Also, the test mode facilitates debugging of test sequences during development. Figure 14 shows a sample test section being run in the test mode.

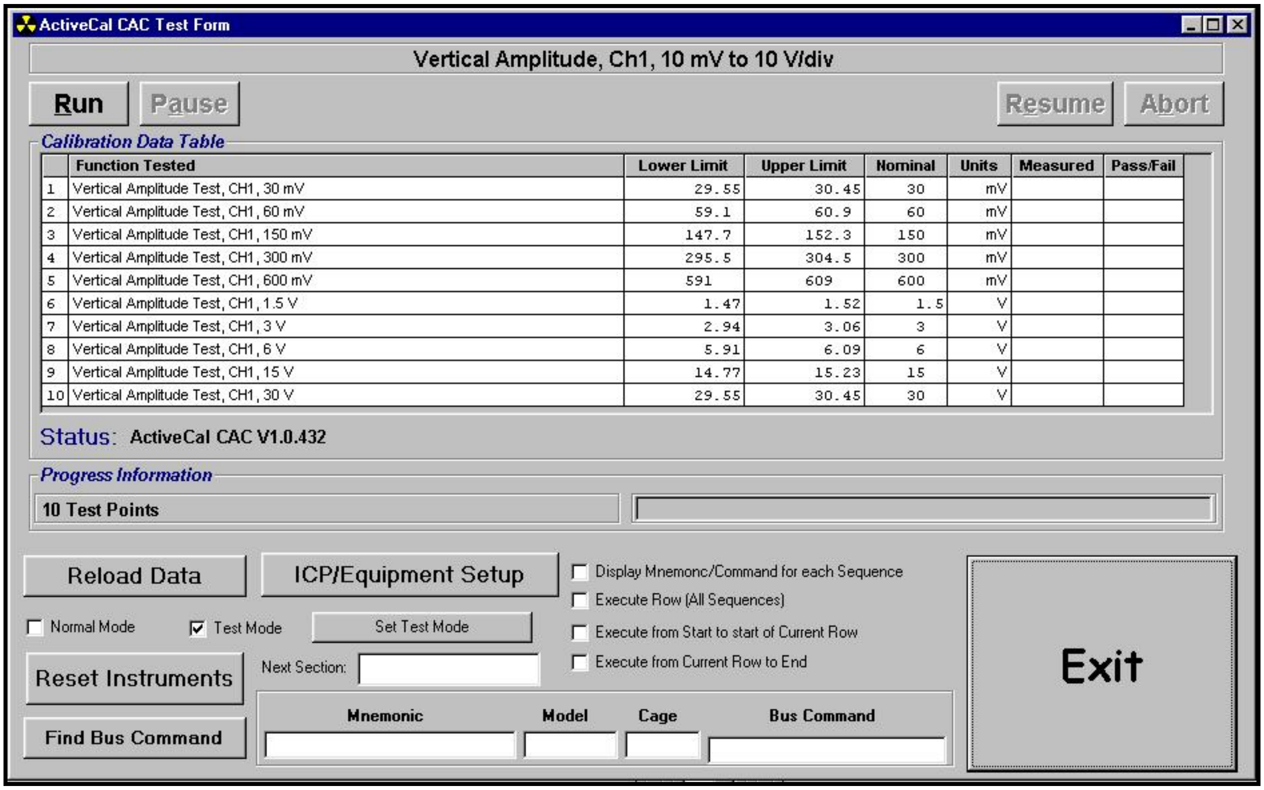


Figure 14. Test Mode Sample Display

4.0 Summary

The U.S. Navy requires a robust calibration procedure development environment to develop, maintain, and publish calibration procedure data for both automated and manual calibrations. The automation provided by embedding ActiveX components into an XML document provides cost effective calibration support for a wide variety of U.S. Navy test and measurement equipment. ActiveX technology allows for the XML document based procedure format to handle automation tasks where feasible or necessary. The environment under development utilizes a database to store and maintain all the calibration procedure data necessary to publish an XML document which contains both automated and manual procedure components.

The ActiveX controls embedded in the XML document perform the necessary automation tasks and are populated by an automated procedure editor. A standardized mnemonic format emulating SCPI is used to standardize the test commands to enable non-programmers to write automated test sequences. The XML calibration procedure consists of modules stored in a database, which provides for measurement method standardization, rapid procedure development through the re-use of test sections or test sequences, and built-in data formatting without the need for word processing. By moving to a database driven automated procedure technology, the U.S. Navy expects to realize significant savings in both development times of automated calibration procedures and their maintenance.

Acknowledgements

The author gratefully acknowledges Dan Cook, and David Kinkade as key members of the automated procedure development team at NSWC Corona.

References

1. SCPI 1997, Standard Commands for Programmable Instruments