

Automated Test Stand for the High Voltage System of the CLAS Drift Chambers

R. Cash, B. Woodhouse

Jefferson Laboratory, Newport News, VA 23606, USA

Abstract

Hall B of CEBAF (Continuous Electron Beam Accelerator Facility) contains CLAS (CEBAF Large Acceptance Spectrometer). CLAS uses eighteen drift chambers to determine the path of charged particles with about 35,000 sense wires. These wires are susceptible to damage if the current or the voltage fluctuates by a few micro-amps or volts. Three CAEN (Costruzioni Apparecchiature Elettroniche Nucleari) (Universal Multichannel Power Supply System SY527) units supply high voltage to the wires. One crate contains ten cards each of which has 24 channels. There are 15 tests that must be performed for each channel, making 10,800 tests for all three units. This task checks the power supply and is time-consuming to do by hand. In building an automated test stand, an interface between the computer and the CAEN was written, and monitoring hardware was built. The automated testing procedure uses LabView to remotely run the tests and control hardware devices.

1. Introduction

1.1 Drift Chambers

The CEBAF Large Acceptance Spectrometer (CLAS) uses drift chambers to determine trajectories, Cherenkov detectors to determine electron/pion separation, scintillation counters to determine the time of flight, and calorimeters to determine the particle. Three regions of drift chambers surround the target with region one closest to the target, region two in the magnetic field of the torus magnet, and region three just beyond the magnetic field. The torus magnet divides each region into six sectors.

Each chamber is made up of two superlayers; the stereo layer is canted 6° to the axial and the axial layer. The superlayers contain six layers of positive, sense wires. Six negative field wires surround each sense wire. The field wires form a hexagonal cell. Guard wires are placed at the outer edge of each superlayer. When a charged particle passes through the chamber, the particle ionizes the gases around the sense wire and sends a pulse through the wire.

Sixteen wires are grouped together and connected to High Voltage Translator Boards (HVTB). Each board has a high voltage cable that contains twelve “baby cables”: 7 for field wires, 3 for sense wires, 1 for guard wire, and 1 for ground. There are 48 HVTBs for Region 3, 25 HVTBs for Region 2, and 6 HVTBs for Region 1.

A CAEN power supply provides high voltage for the drift chambers. This power supply must function properly because the wires may break due to a slight variation in voltage or current.

1.2 CAEN Power Supply System SY527

1.2.1 Function of CAEN

A CAEN Universal Multi-Channel Power Supply System SY527 provides the high voltage for the CLAS Drift Chambers. Each SY527 crate has slots for 10 different cards and is rack-mountable. An individual card provides voltage and current. Each card can supply a preset maximum current and voltage, and it can produce a controllable constant voltage with current on demand. The Drift Chamber High Voltage (DCHV) system uses three different types of cards: positive 20 μA , positive 40 μA , or negative 40 μA , each with a maximum voltage of 2.5 kV. There are 24 channels per card and the current, voltage, voltage ramp-up/down rate, and over-current trip value can be independently controlled for each channel. A critical aspect of the SY527 is its ability to shut down a channel if the channel draws current that exceeds a preset value, a process known as tripping.

1.2.2 Interface between the CAEN and computer

The SY527 can be controlled from a remote location via a High Speed CAENET port or an R232C Serial Port. The High Speed CAENET is CAEN’s proprietary protocol and is used to control the SY527 during Hall operation. The Serial port follows the VT100 protocol and is used by the Automated Test Stand and is intended to be used as a backup during Hall operations.

The VT100 protocol is often used for human interface between devices. The RS232C can be connected to a computer’s serial port. Any VT100 terminal emulator can then be used to interface with the SY527. (See Figure 4-shows the interface to the SY527 connected via the serial port.)

1.2.3 Communicating with SY527

Connecting to the SY527 through the serial port was intended to provide manual interaction with the system from the counting house. Therefore, it is not optimized for programmatic control. The VT100 protocol communicates using hexadecimal values and ASCII character values. All ASCII character values are outputted to the screen at the current cursor location. In order for the devices to communicate commands, a command sequence is used. The sequence is opened with the ESCAPE command: hexadecimal value 1B. An open bracket follows the ESC command. The sequence contains the attributes for the command. The final portion of the sequence is the actual command. For example, the sequence “ESC[5,1H” will move the cursor to the fifth row in the first column. (See Figure 1- lists all commands used by the SY527.) These commands control the location of the cursor and the way in which the text is outputted to the screen.

The SY527 changes information in two ways. The first is through direct updating. This is used when a value changes on the page currently being displayed. The cursor is moved to a certain setting and the new value for just that setting is changed. The second way is by refreshing a page. Updating a page or moving to the next or previous page will refresh the display. When the display is refreshed, all values for the current page are written, line-by-line. This process also writes the labeling for the page.

The commands that control the SY527 are application specific and are handled by the software within the CAEN unit. These commands are called application commands. (See Figure 2-shows the commands that the SY527 software recognizes.)

1.3 Labview

National Instruments LabView 6i was the software package chosen to develop the new software. This application allows for the development of powerful programs without complex knowledge of a programming language. This application also makes it easy to understand complex programs through its graphical representation of program flow. All LabView programs are contained in a virtual instrument (VI).

LabView can access computer ports through various built-in libraries. The process of reading or writing data from or to a serial port is simple in LabView. To read data, the number of bytes available is found by a built-in function. Another function then reads that number of bytes from the serial port. Inside the computer there is a buffer to hold all data until it is read. To write to the serial port there is a built-in function that places data in a buffer to be automatically sent over the serial port.

2 Procedure

2.1 Software Interface

For a test stand to be truly automated, there must be an automated process to control all necessary functions of the SY527. After looking at the requirements of the test stand, it was determined that an interface to the SY527 should be built that is usable by any process, manual or automated. The interface was written in LabView, and all inputs and outputs of the interface are accessible through variables and functions.

2.1.1 Reading from the SY527

The first step when reading data that is incoming from the CAEN is to get the data from the serial port. LabView opens the serial port and copies any available bytes into a buffer. The data is sent in a serial format, meaning they are received one byte at a time. If the serial port data is copied while it is in the middle of receiving a command sequence, only the received data is copied. This incomplete data would not be interpreted correctly.

To account for this, the program scans for the last command sequence, and cuts the data before the sequence starts. The data before the cut is sent to the interpreter. Any data after the cut is sent to a buffer and is connected to the next collection of data. This process makes sure only whole commands are sent. Once the data reaches the interpreter it goes through three different functions.

The first function, parses out all unnecessary strings. New-line characters, spaces, and certain status messages are removed. This ensures that all of the data for each command is in a similar form. Then, all ESC commands (hex 1B) are replaced with ESC [. This makes it easier to understand data that it going through the interpreter. Also, before every escape command, a new-line is inserted. After going through the parsing function of the interpreter, the data is in a string with each command separated by a new line.

The second function takes the data out of the string, and places each component of the command into an array. This function only accepts the command to move the cursor or to highlight text; these are the only commands important to the operation of the interface. The move cursor commands are split into three components. First the Y-coordinate attribute of the command is copied, then the X-coordinate, followed by the data after the move cursor command (this is the data to be written, and is where all the values of the SY527 are held). All this data is copied into a two-dimensional array in which each command is in its own row. This array is sent to the final function. The only time the SY527 uses highlighted text is to show the User Cursor. The highlight text command is given after the cursor is set to the new position. When the highlight command is caught, the program checks where the cursor was last set and copies that information into a global variable. This is all to allow other functions to know where the user cursor is currently located.

The third function of the interpreter, actually interprets each command and the values they are trying to change. To be quick this function only interprets commands that change an entire line, as opposed to direct update commands. The program looks at the X-coordinate to determine if a whole line is being changed, or just a single value. If the cursor is moved to column 1, it can be assumed a whole row is being written as opposed to just a single channel value. If the line written is a label, not channel values, then it will not fit the format of the channel line, and nothing will be copied. When a new line is detected, the program copies out several values. It starts copying at a certain character number, and stops reading a certain number of characters afterwards. The program copies the monitored voltage, monitored current, demand voltage, and trip current. It also reads whether the channel is on or off. It then puts all of this status info into an array. The program then reads the last 4 characters. These four characters contain the card and the channel of the values just copied. If the card is the wrong card, the values are discarded. Otherwise, the values are stored in the channel that is indicated by the last two characters. All these values are added into an array that has the most recent values for each channel. Also, the program looks to see if the cursor is set to position (2, 70), this is where the current page is written. If this position is written to, the data being written (the current page) is put into a global variable. Any other function now knows the current page. The output of this function, and of the entire interpreter is an array, which contains the most recent values for all channels. This array is placed in a global variable, accessible to any process.

2.1.2 Continuous Updating of SY527 Values

The interface needs to not only read things coming from the CAEN, it also needs to continuously update. To do this, program must move between the two pages that hold the 24 channel values (each page only displays 16 channels), and get the latest values for each channel. This process is fairly simple, but is essential for the correct operation of the interface. As soon as the program starts the connection to the SY527, the character “U” is sent. This is an application command that will update the current page. This lets the program determine what page it is on. Once the current page is known, the program sends the “n” character to move forward a page, or the “p” character to move back one page. This is done until the current page is Page 0 (assuming you are testing Card 0). The program waits 2.5 seconds, allowing all the values to completely update. The system then detects what page it is on and moves to Page 1 (assuming testing Card 0). After 2.5 seconds, the program moves back to Page 0, and loops until the program is disconnected. This function is known as paging. While this process is running, another thread is continuously reading values that are coming from the SY527. There is also a thread running that looks for error from any process and either fixes them or presents them to the user. The paging function, and continuous reading is one of the most important parts of the interface. Through these two processes, all 24 channels are available for use in a simple array with constantly updating values.

2.2.3 Writing to the SY527

While it is important to be able to monitor channel values, an interface would be useless for testing purposes if it could not write to the CAEN. The three important values that might need to be changed for each channel are demand voltage, trip-current level, and turning the channel on or off. A separate function was written for this process. The desired channel, setting, and new value are sent to the function. The writing function sets a Wait variable, which stops the paging operation. If the paging is not stopped, the page might be switched while the interface is trying to write to the SY527. After the Wait variable is set, the function determines what page it is on, and moves to the correct page for the specified channel. The function then sends the hex commands to move the user cursor down to the correct channel, and over to the correct setting. To move up, the command is ESC[A, for down it is ESC[B, for left it is ESC[C for right it is ESC[D. Once the current user cursor position equals the desired position for the user cursor, the new value is sent to the SY527. For a new demand voltage or current trip, the value is sent as numbers followed by the return key. To turn a channel on or off the application command “c” is sent. This simply switches the status of the channel. For this reason, the interface must keep track of the channel status. If the user wants to turn a channel on and the “c” command is sent, the channel will turn off, if it was already on. The interface uses the On/Off variable (which is set during the reading operation) to determine the status of the channel. If the status is opposite the desired status, then the “c” command is sent. If the channel status and desired status are the same, nothing happens. Once the new values have been written, the Wait variable is unset. This final part of the interface allows complete control of the most important functions of the SY527. The writing function along with the reading functions creates the complete interface.

2.2.4 Interface Routines

The reading and writing routines make the SY527 interface functional. However, certain operations would still require some direct commands to the SY527. To avoid this

and control the SY527 easily, extra routines were built into the software interface. First, a routine was built to move to a specified page. This involves looking at the current page and sending “next” or “previous” commands as required. Another routine was written which would move the user cursor to a specified position, using the same process the paging routine does. On an even higher level, two routines were written to turn all channels on or off, or to change the demand voltage for all channels. These routines automate the process of calling the Write function 24 times. These routines also have built-in safety measures against overwriting channels, or errors caused by writing new values too fast. These routines make the SY527 interface more automated, and further simplifies control of the SY527.

2.2 Hardware Tests.

Several of the tests (See Figure 6) could not be done without external hardware. In particular Tests 3, 9, 11, and 15 all require an external meter. Also, Test 15, the Current Trip Test, needs an external circuit that significantly increases the current to a channel.

A chassis was built to monitor the voltages and test the current trip values (See Figure). This chassis has a 24-channel faceplate with 24 safe high voltage (SHV) connectors to test one card. Inside the chassis, 7 National Instruments Field Point (FP) Modules are attached to the bottom of the chassis using a rail. A FP-1600 provides power and network connection, 3 FP-AI-110’s monitor the voltages, and 3 FP-RLY-420’s switch the JC1aF relays on and off. Two 7.5x20 inch CAEN High Voltage Test Boards are raised 4.5 inches off the floor of the chassis. Each board holds the circuitry for 12 channels, with each channel having four 100-Mohm resistors, one 1-Mohm resistor, one JC1aF relay, and one diode. When the relay is activated, it bypasses the last two 100-Mohm resistors. This doubles the current and allows the trip sensor to be tested. The circuit also uses a diode to prevent chatter from the relay. To measure the voltage of the circuit an FP-AI-110 is placed in parallel with a 1-Mohm resistor. The 1-Mohm resistor creates a voltage drop though the parallel circuit at a constant ratio of 1/401. This lower voltage allows proper operation of the field point module.

2.3 Chassis Interface

2.3.1 Reading from the FieldPoint Modules

In order for the chassis to do its job, the Test Stand Software must be able to communicate with the hardware. The Test Stand software must be able to read the channel voltages and turn the relays on or off. The interface to the hardware is similar to the SY527 interface. The first step for the Hardware Interface is to initialize the field point modules. FieldPoint Explorer functions are used to open the connection to the AI-110s and RLY-420s. The 24 voltages being received by the FieldPoint Modules are copied into an array every tenth of a second (this is more than enough since the modules update only every .17 seconds). Each value in the array is then multiplied by a calibration value (see section 2.3.2) to account for the approximate 1/401 voltage drop. These monitored voltages are then available to any process through a global variable. This hardware interface is started as soon as the SY527 interface is started.

2.3.2 Calibration of FieldPoint Module Voltages

Modern electronics involve slight tolerances for error. The resistors used in our hardware are all of 5% tolerance. This number can be misleading, because of the 1/400 voltage drop. Five percent tolerance when multiplied by 400 gives a significant error. This fact was recognized, and a calibration system was developed. While the resistors do

have a margin of error, their resistance is constant. It proved difficult to measure the exact resistance of the 100M resistors, so it was decided to deduce the errors. The program looks for a user file that has an offset voltage for the SY527. This offset file is an excel file that holds 24 values in the first 24 rows. These values are found by ramping the voltage up to around 100 volts, and comparing indicated voltage by the SY527 to a meter measured voltage. This relative error of the SY527 is generally constant, even as the voltage increases. The calibration program then ramps the SY527 to 1000V followed by 1600V. The indicated voltage of the SY527 is adjusted by the offset value(giving the true voltage) and then compared to the FieldPoint voltages. The slope of these two points is then taken and used as the calibration voltage. The values for the 24-channels, generally between 380 and 420 V/mV, are put into an array. The values in this array are then used as the calibration numbers that are multiplied to each FieldPoint Reading. After calibrating, each FieldPoint is accurate to within 10 volts.

2.3.3 Hardware Relay control

The JC1aF relays are controlled by the FP-RLY-420s. An interface to each of these relays was also written. When the Relay Control program is started, it takes the desired channel (25 indicates all channels) and the new value for that channel. The RLY-420s are then opened and the new value written to them. These relays have a trip time in the range of milliseconds. As with the SY527, when the hardware reading and writing functions are combined, a fully function hardware interface is available to any process.

2.5 Test Controller

The goal of the interfaces is to allow the tests for the SY527 to be written in LabView very easily, with no direct interfacing to the SY527. While, the interfaces make this task possible, a program is still needed to run each test. Running simultaneously with the interfaces is a Test Controller. First the controller waits for all connections to be made, and for all calibrations to be conducted. Then, the program runs all tests in a predefined sequence. Once one test, or one set of related tests is finished, the results of the test(s) are displayed for the user, and the next test is run. Once all tests are completed, all channels are turned off, and all of the connections are shut down. Each test must follow a standard format and has access to all interface functions and variables. (See Appendix B.3.1-for information about writing tests.) When the SY527 interface, the hardware interface, and the test controller are running, the Automated Stand is running.

3 Conclusion

The tests required to ensure proper operation of the CAEN are time consuming when done manually. It was estimated that it would take about 16 hours to test one card by a CEBAF technician. Due to the amount of cards that must be tested, 30 in all, many channels and tests were skipped to save time. The Automated Test Stand is a quick and efficient way to test an entire card with the click of a button. All 24 channels of one card can be plugged into the faceplate of the Test Stand and tested with the LabView program. The hardware software runs all the tests and informs the user the channels that have failed which tests. This procedure takes approximately 3 hours and checks every channel and runs every test. The Automated Test Stand saves about 13 hours and hundreds of dollars worth of labor.

The CAEN interface allows a user to easily operate the CAEN through LabView and runs independently from the actual hardware tests. Both interfaces were

programmed to run independently of each other so any other program could utilize the CAEN interface.

The chassis was a successful prototype that will be used in the near future to streamline CAEN testing procedures. Its accuracy could be improved by using 1% tolerance resistors instead of the 5% tolerance resistors. In addition, a chassis that can test more cards or multiple chassis will probably be needed considering that there are 30 cards for the system.

- Commands: ESC [x; x; ... com
- \1B (Hex code 1B, dec 27) = ESC
- [= Open Command Sequence
- x;x...= However many attributes (function (a,b))
- Com = The actual Command
- Ex. : \1B [5,1H = Move Cursor to Position 5,1
- Command H = Move Cursor to position
- Command K = ClrLineCommand 7 = Store Cursor Position
- Command 8 = Recall Cursor Position (7 & 8 make cursor blink)
- Command 7M = Following Text Bold.

Figure 1. SY527 ASCII commands

CAEN Commands

“U” Refreshes the Current Page
 “N” Moves to Next Page and Refreshes
 “P” Moves to Previous Page and Refreshes
 ESC [A : Move Cursor Up
 ESC [B : Move Cursor Down
 ESC [C : Move Cursor Left
 ESC [D : Move Cursor Right
 “C” Changes Current Value (On/Off)

Figure 2. CAEN Application Commands

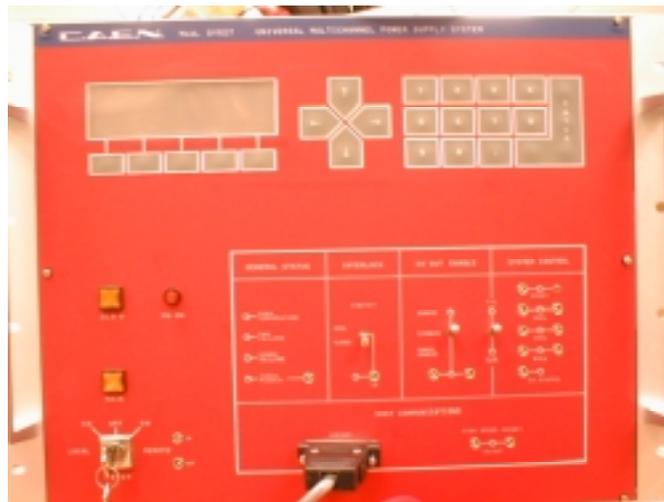


Figure 3. SY527 Front Panel

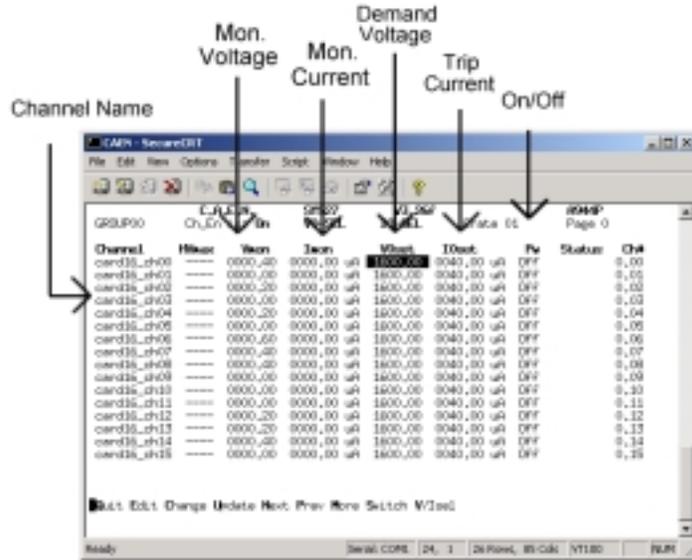


Figure 4. SY527 Interface through Serial Connection

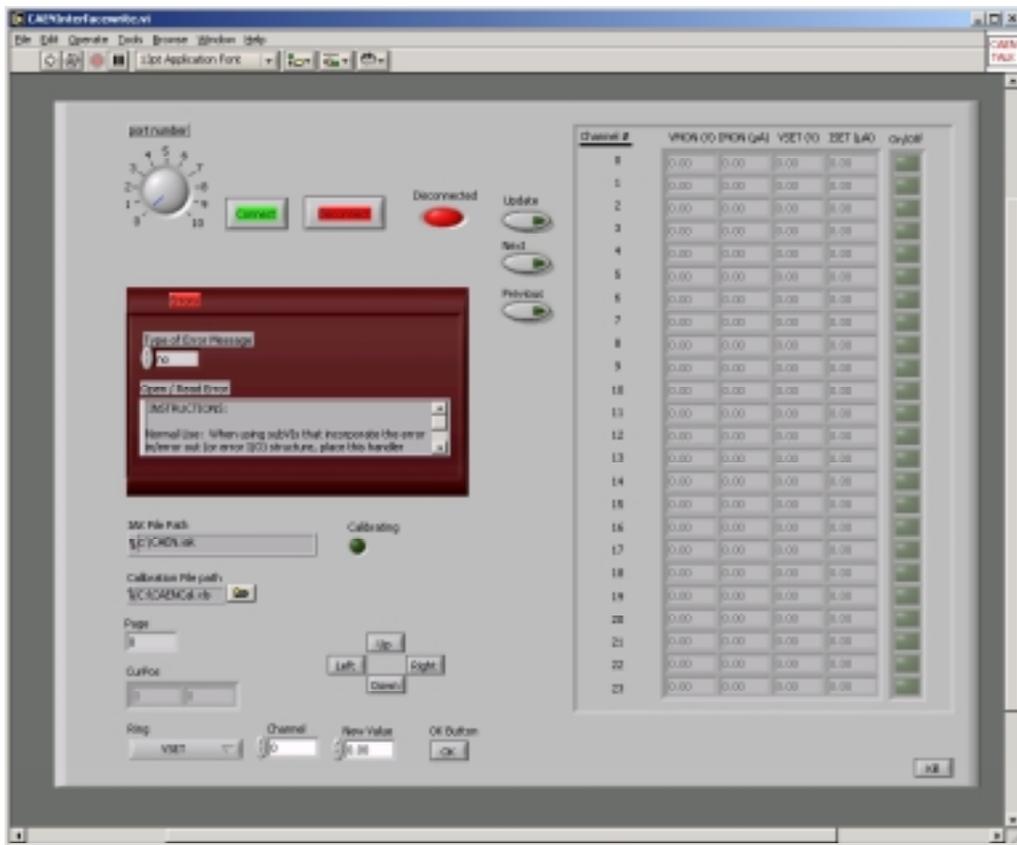


Figure 5. SY527 Automated Interface

Software Tests:

1. Voltage Granularity- Ensure that channels do not effect other channels' voltages by testing one channel at a time.
2. Voltage Variation- Ramp all channels to a voltage. The voltage cannot vary by one volt from channel to channel.
3. Voltage Setting and Resolution- The demand and external voltage meter values cannot vary by more than six volts.
4. Voltage Monitoring Resolution and Accuracy- The demand and CAEN voltage monitor values cannot vary by more than six volts.
5. Voltage Reproducibility- Ramp the channels to the same voltage repeatedly. It cannot vary by more than one volt each time.
6. Voltage Stability- The voltage cannot vary by more than one volt for each channel for a 24-hour period.
7. Voltage Ripple- Tests for an acceptable voltage ripple during the voltage ramp.
8. Turn-On/Turn-Off- Make sure supply turns on.
9. Ramp Rates- Ramp the voltage at a given rate and record time to reach voltage. The times should equal each other.
10. Maximum Output Current- Make sure a current cannot be entered that is above the maximum current the card can supply.
11. Current Monitoring Resolution and Accuracy- Compare the CAEN read back current to the electrometer value and the difference cannot be more than 40 nano-amps.

Hardware Tests:

12. Channel Isolation- Set current trip level low and trip a channel. Make sure other channels do not trip.
13. Voltage Range- Set the hardware limit and verify the voltage cannot be raised above value.
14. Voltage Regulation- Test CAEN's ability to handle variations in provided voltage.
15. Over Current Protection- Make sure channels shut off when current goes above trip level.

Figure 6. SY527 Tests

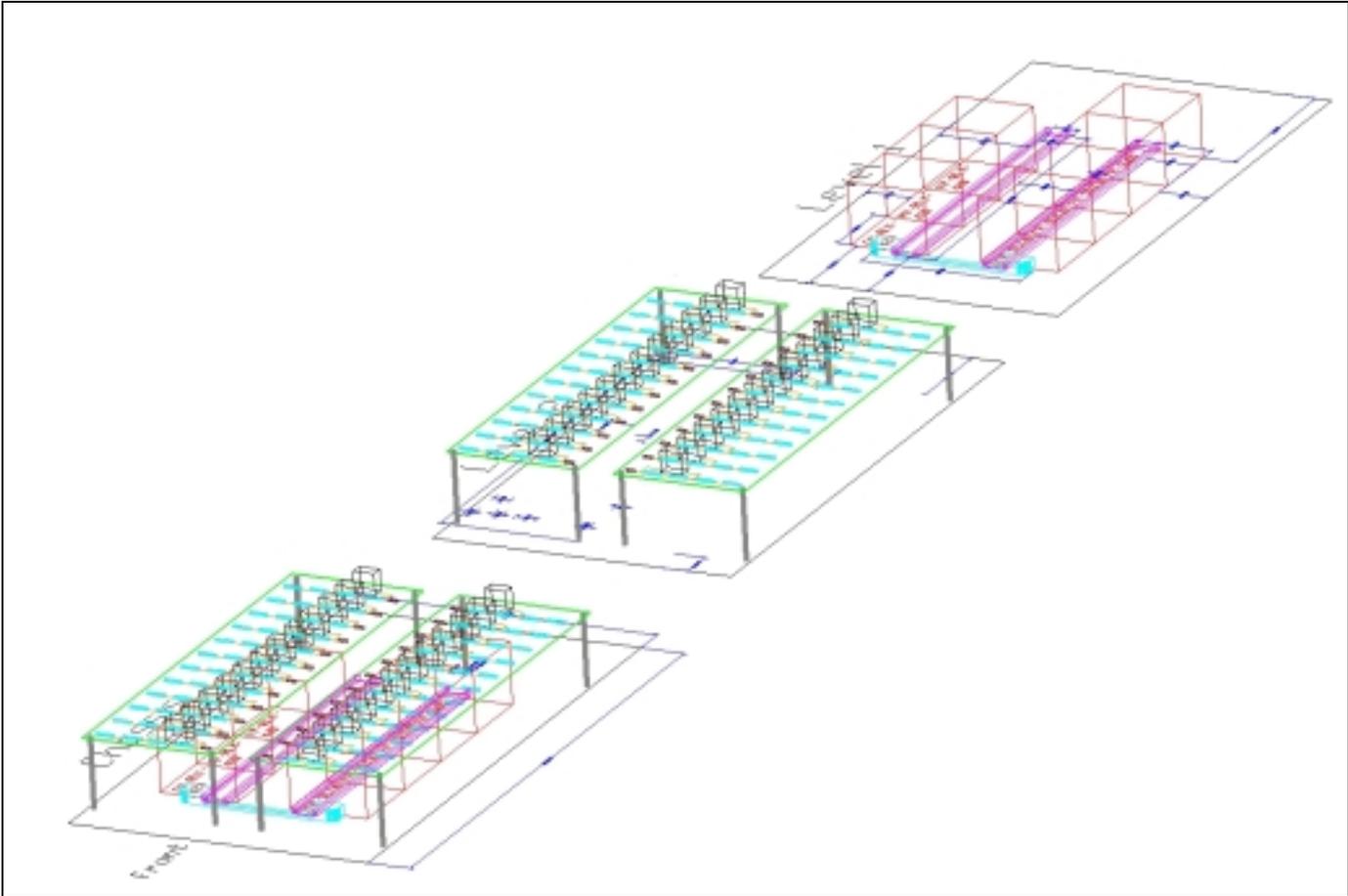


Figure 7. Chassis AutoCAD Design



Figure 8. Completed Chassis

Automated Test Stand Interface User Guide

The software interfaces used in the Test Stand can be used to write more tests or as a stand-alone SY527 interface. This document explains how to use the SY527 interface, the hardware interface, and the Test Controller. All interfaces are contained in the LabView Library ATSIInterface.llb.

B.1 SY527 Interface

B.1.1 Connecting and Disconnecting

B.1.1.1 Process

The VI to interface with the SY527 is found in the VI CAENInterface.vi with the ATS library file. To start this VI from another VI, insert a CAENInterface.vi control. Two values need to be wired to the VI. The first value must be true if you want to start the SY527 interface (setting to false does not stop the interface). The second value is a boolean which determines whether the hardware interface should be started. The value defaults to true. If the SY527 Interface is to be used as a stand-alone, this value should be set to false. To disconnect the SY527 Interface at any time, the Disconnect global variable should be set to true. The variable is found in CAENVarmess.vi in the ATS library. Allow approximately two seconds for the disconnect to complete.

If the CAENInterface VI is used for manual control, the connect button must be depressed before the VI is run. Once running, the disconnect button can be used to disconnect from the SY527.

B.1.1.2 Connect/Disconnect Options

All serial port options are currently hardwired. If any serial setting needs to be set to anything other than port defaults, they must be programmed. Currently, all options are hardwired to defaults, and the port is set to Port 0. If a different port is needed, several options need to be rewired with the new port setting, as not all port options are wired to the rotary control. After disconnecting, errors may cause the VI to fail to release the serial port for use. In this case, there is a kill button that will automatically close the serial port. All these options require manually actions, either on the front panel, or in the coding.

B.1.2 Reading Values

All channel values are available through global variables. Monitored voltage (VMON), monitored current (IMON), demand voltage (VSET), and trip current (ISET), are in the global variable Voltages in CAENVarmess.vi in the ATS library. Voltages is 2-dimensional array of Double values. Channels 0 – 23 are held in array rows 1 – 24 (Caution: Array Row 0 holds irrelevant data). Column 0 is VMON, Column 1 is IMON, Column 2 is VSET, and Column 3 is ISET. The channel status (On/Off) is held in the array On/Off in the CAENVarmess.vi in the ATS library. On/Off is a 1-dimensional Boolean array with true representing a channel that is one. Channels 0 – 23 are held in the array in Rows 1 – 24 (Caution: Array Row 0 holds irrelevant data). Both Voltages and On/Off should be read-only.

If the interface is used manually, all values are visible in the array indicators on the front panel. Each column and row is labeled.

B.1.3 Changing Single SY527 Page Values

To change a single value for a channel, call the VI WriteCAEN.vi. This VI requires 4 inputs and has one output. The first input is a Boolean and must be true to run

the VI. The second input is the setting to change. 0 means the demand voltage, 1 means the trip current, and 2 means channel status (on/off). The third input is the desired channel and can be from 0 – 23. The fourth input is the new value, and is a Double precision data type. If the value to be changed is the channel status, 1 means turn the channel on, and 0 means turn the channel off. If the channel is already at the desired status, nothing happens. The output of this VI is an error cluster that contains any errors. This VI can also be run manually, if all inputs are supplied.

B.1.4 Changing Multiple SY527 Page Values

Two different routines are built into the Automated Test Stand to change all the channels values for a certain setting. These routines write the data in a way that is much faster than using WriteCAEN for every channel. AllVset.vi in the ATS library changes the demand voltage for every channel. When calling this VI, the new demand voltage must be supplied in Double form. The VI will output any errors when it is finished. Allow approximately 14 seconds for the routine to run.

AllPwr.vi in the ATS library will turn all channels on or off. When calling this VI, a Boolean must be supplied. True means turn all channels on; false means turn all channels off. If a channel is already at the desired state, nothing will happen. Allow approximately 14 seconds for the routine to run. Both AllVset and AllPwr are controllable manually. Enter the required values, and then run the VI.

B.2 Hardware Interface

B.2.1 Connecting and Disconnecting from the Hardware

The hardware interface is contained in the file FPInterface.vi. To start the interface, call the VI and supply it with several inputs. The first input is a Boolean which should be set to true to start the interface. The second input is also a Boolean which controls the relays; generally this value should be set to false when starting the interface. The third input value is a double, which holds the new value for the relays. Usually, this input can and should be ignored. The fourth input is the path to the Field Point Explorer .IAK file, which holds the hardware's Field Point Modules. The fifth input is the refresh rate for interface in milliseconds. The default (170) is usually sufficient, and this input can be ignored. The sixth input value is the path to the calibration file (See section B.2.2). The interface will begin to calibrate (takes approximately a minute and a half).

To disconnect from the hardware interface, set the Disconnect global variable in CAENVarmess.vi in the ATS library (this is the same variable to disconnect from the SY527 interface).

All of these inputs are available on the front panel of the hardware interface as well.

B.2.2 Calibrating Hardware Values

Because of electronic limitations there are relatively broad margins of error (5%) with monitored voltage. However, these errors are fairly constant, so calibrating certain functions will negate most of the error. All of the calibration is automated, however, the calibration program needs certain information. When starting the hardware interface, the controlling process must send the calibration file path. The calibration file is a list of offsets for each of the 24 channels. To increase the accuracy of the test stand, it may be helpful to create a new offset file for each card. To create the file, ramp each channel from the SY527 to 100 volts. Obtain a meter, and measure the voltage going through the channel. Subtract the SY527 indicated voltage from the metered voltage (cannot be done

the other way). For each channel insert the value found into an excel file starting at row 1 and ending at row 24. The path to this file must be sent to the hardware interface.

B.2.3 Reading Values from the Hardware Interface

Reading values from the Hardware interface is similar to the SY527 interface. The values are placed in a two-dimensional array named MonVoltage in CAENVarmess.vi in the ATS library. This array has the monitored values for channel 0 – 23 in rows 1 – 24. The interface makes two different values available. In column 0 is the voltage measured by each Field Point Module. In column 1 is the approximate voltage of each channel, without the voltage drop provided by the hardware circuitry, as measured by the Field Point Module. This two-dimensional array is available to all processes. These values are display on the front panel as well.

B.3 Test Controller

B.3.1 Creating Tests for the Test Controller

While the interfaces can stand alone, the Test Controller, MainTest.vi requires the two interfaces, and is built specifically for the Automated Test Stand. It is designed so that it is relatively easy to add in extra tests. Each test must conform to certain standards to work with the Test Controller. Each Test must have a Boolean input to start the test (true should mean “start test”). A path name for the results file must be provided. This is because each test should output all results to an Excel file. There are built-in routines to do this (see section B.4.1). Each test must output an error cluster for any errors. Also, the test should output a Boolean that indicates whether the test failed. A true indicates a failed test.

For every test, some addition programming must be done inside the Test Controller. Each test is called at different times. Determine the best time for the test to run and create a new frame in the testing sequence in MainTest.vi Inside the frame, wire the number of the test to the global variable CurrentTest in CAENVarmess.vi in the ATS library. Wire any required inputs to the test. To wire the Boolean result of the test, add an extra input to the Or operator. Also add an error combiner to other tests’ errors, and wire the error output of the new test to the error combiner. Front panel functionality may be added in any fashion but should generally be wired similarly to the included tests.

B.3.2 Running the Tests

To run the Test Controller, wire a True Boolean to the Start Test input of MainTest.vi in the ATS library. Also wire the path of the test result excel file. Either wire an existing path to be overwritten or enter a new excel file to be created. After the tests have completed, MainTest.vi will return a false if all tests passed, and a true if a test fails. The front panel of the Test Controller will indicate which tests failed. From there the front panel of any test can be opened and the exact results viewed. Alternatively, the excel result file can be opened. This result file will contain a copy of all the values measured by each test. The tests can be run through MainTest.vi’s front panel.

B.4 Writing to the Results File

To keep outputs very simple, the result of every test goes into a different worksheet in the same excel file. This is a very complicated process, so routines were written to simplify the process.

Before writing data, a new workbook must be created. To do this, insert CreateNewSheet.vi. The path of the results file and the name for the new worksheet must be wired as inputs. The function gives any errors as an output. Once the sheet is created.

Values can be written to the new worksheet. Call the function WriteExcel.vi in the ATS library to write values to the result file. The name of the worksheet to write to must be provided as an input. Also, the range of cells to write to must be wired to the input. The function also requires the path of the file, and an array of Double values. The array is where all the data to be written is held. Using these functions, the results can be easily outputted in a neat format.

B.5 Using the Automated Test Stand

Using the Automated Test Stand requires very little work. First, the channels of Card 0 in a SY527 must be connected to the correct SHV connection on the Chassis front panel. Then, 24V must be supplied to the power connection in the rear of the chassis. FieldPoint explorer must be used to locate the Hardware Field Point Modules and create an .IAK file. Then, start up the SY527. Once the system has booted, press “D” to display channel values. Finally load up the CAENStart.vi program. Fill in the .IAK file path, the calibration file path (see B.2.2), and the path for the result file. Run the VI and after starting all the hardware, and press the Start Interface button. Then at any time, press the Begin Testing button. Once the interfaces stop calibrating, the tests will begin. Once the tests are finished, the Test Finished light should come on. The Test Fail light will come on at any time. The Automated Stand is completely automated, and requires no interaction while it is running.