High Voltage Control of The Modular Neutron Array

National Superconducting Cyclotron Laboratory
Michigan State University
REU Fellowship
Scott Marley
August 8th, 2003

Abstract

The Modular Neutron Array (MoNA) is a high efficiency, large area neutron detector being completed at the National Superconducting Cyclotron Laboratory for detection of neutron energies between 50MeV and 250MeV. MoNA is comprised of 144 scintillating detector modules in a nine by sixteen configuration and stands 0.9m × 2.0m × 1.6m. Two high voltage power supplies are used to provide the modules the potential needed to detect neutron events. An effective method of controlling these power supplies remotely is needed for use during developmental and experimental use of the detector. A C++ program was created to provide an adaptable user interface with the high voltage power supplies.

Introduction

The Modular Neutron Array (MoNA) is a high efficiency neutron detector located at the National Superconducting Cyclotron Laboratory on the campus of Michigan State University. MoNA is designed to detect neutrons that come from the break up reactions of rare isotopes. The more that is know of these nuclei the better the understanding of matter and give light into the processes that occur inside stellar bodies. The construction and development of MoNA has been a collaborative effort consisting of nine undergraduate institutions with funding from the National Science Foundation. The participating colleges are Ball State University, Central Michigan University, Concordia College at Moorhead, Hope College, Indiana University South Bend, Michigan State University, Milikin University, Western Michigan University, and Westmont College. Florida State University is also contributing though the construction of a sweeper magnet to bend the beam of charged particles away from the detector and provide a more accurate account of neutron events.

The detector is comprised of 144 scintillating detector bars that are of the dimensions 0.1m x 0.1m x 2.0m and have a photomultiplier tube on each end. The current configuration of MoNA is nine "layers" of sixteen modules, which gives the detector a total volume of 0.9 x 2.0 x 1.6 m^3.

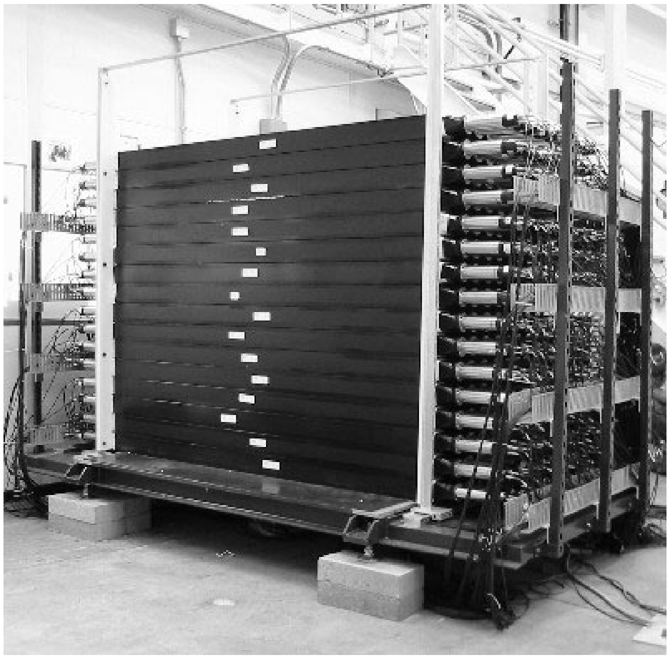

Figure 1 - MoNA set up in nine layers of sixteen bars each

The detector bars are composed of an organic scintillating material, BC-408, which can is sensitive to electrons, protons, alpha particles, gamma rays, and neutrons. Since neutrons have no charge, their detection must be done through indirect means, specifically through their reactions with other particles. When a neutron interacts within the bar, it collides with a charged particle, which is usually a proton. The energy transferred to the proton causes it to move about the detector and ultimately cause scintillation. The photons that are emitted travel down both ends of the bar either directly or by reflection. Located at each end is a photomultiplier tube where photons of visible light are converted into a functional signal.

The photon that passes through the window of a photomultiplier tube hits a photocathode, which is being held at a very low electrical potential. The product of this interaction is one or more electron being freed from the photocathode due to the photoelectric effect. The photocathode, holding a large negative charge, causes the electron(s) to move further into the photomultiplier tube and toward the first dynode, which is at a higher potential. When an electron strikes the dynode, more electrons are released and move away from the first dynode toward the next dynode that has an even higher electrical potential. This process is repeated, electrons freeing more electrons, through a total of twelve dynode stages and ends at the anode of the photomultiplier tube where a gain on the order of 10^7 is reached. At this point the anode signal can be used to attain several different forms of information about the neutron event. The potential difference needed for each of the photomultiplier tubes is approximately 2 kV, which is provided by MoNA's high voltage system.

High Voltage Power Supplies

In order for all of the detector modules in MoNA to function properly, all 288 photomultiplier tubes must be connected to a high voltage source. This source is actually two power supplies located in the electronics racks behind the detector. The CAEN SY1527 power supplies are multi-channel systems that can be configured to output high or low voltages. This is done by the installation of output boards in the slots provided in the rear of each power supply. Since MoNA requires high voltages for the photomultiplier tubes, eleven CAEN A1733B high voltage boards were installed. Each board can output 28 channels of up to 3kV at 3mA of current. Large high voltage cables are connected to these boards and from there into small high voltage distribution boxes where each channel is separated to a SHV cable connector. SHV cables are then connected to the small high voltage boxes and then to the photomultiplier tubes.

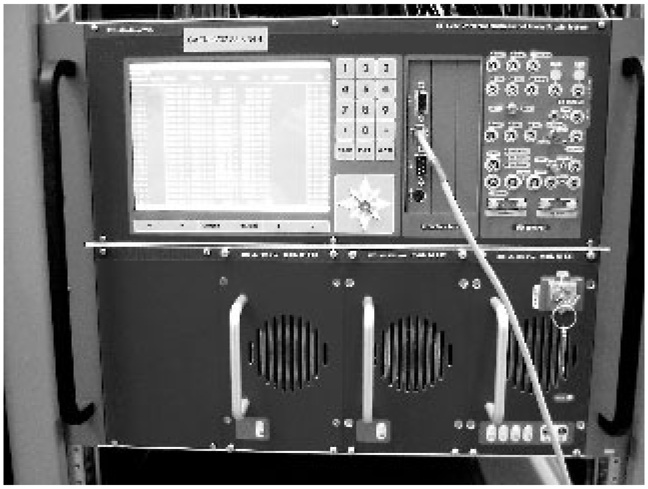

Figure 2 - One MoNA's CAEN SY1527 Power Supplies

The functionality of MoNA is dependant on the high voltage power supplies connected to the array. Thus, if control of the power supplies could be done in a straightforward and effective manner it would make the detector easier to control as well. The readily available methods to operate the CAEN power supplies both involve the same interface. One method would be to connect a keyboard to the front of a power supply and using the LCD screen, modify the parameters from there. This option works well during calibration and other testing procedures, but it would not be practical during an experiment that involves beam time due to the fact that it would require a physical presence inside the vault. The other method is to control the power supplies via a terminal connection. The power supplies have a built-in terminal server that can be accessed through a telnet client. Once connected the interface is identical to using the keyboard to change the parameters. A positive aspect of the terminal server is that it can be controlled over a network with relative ease. In order to control all of the high voltage channels one would need to open two separate telnet connections which would mean two terminal windows. Module operations would require parameters to be set in one window and then the other (e.g. powering up a full layer in the detector). A third option exists that is not as readily available, but potentially more efficient than the first two: writing a program. A program could be designed to whatever specifications that are needed for basic or advanced control of the power supplies.

Program Layout

The profile for this program was that a graphical user interface (GUI) written in the Tcl/Tk scripting language could be used to control the voltage and power (on/off) of the channels on high voltage power supplies. Tcl/Tk GUIs are not difficult to construct, however the power supplies cannot be directly controlled by Tcl procedures. CAEN has developed a C/C++ library of functions for the SY1527 model power supply that can be used to create a control structure. The interesting part was coming up with a way for a Tcl/Tk GUI to communicate with a C++ program. The design that was chosen was that the GUI would write out its information to a Tcl parameter file. These parameters pertain to which power supply parameters (voltage and power) are to be set and for which channels. When called by the GUI, the C++ program reads the parameters in from the file and interprets them in a way that can then be sent to the power supplies. The C++ function also has the ability to read the current status of channels and output that data into standard output, which can be piped back into the Tcl/Tk GUI and displayed.

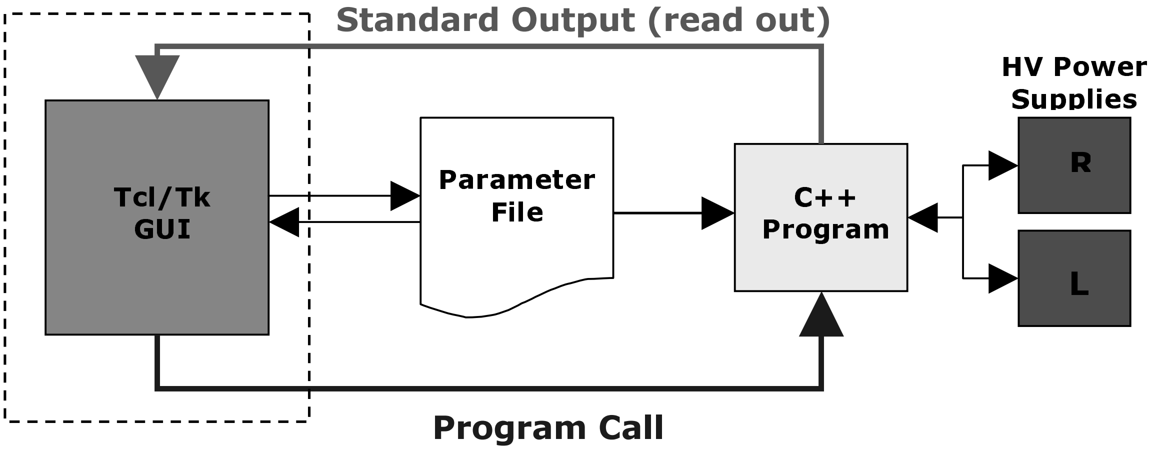

Figure 3 – Structure of Tcl/Tk GUI and C++ program interface

Tcl Parameter File

The Tcl parameter file consists of the declaration of three Tcl arrays with 288 elements each. In order to simplify how the photomultiplier tubes are referenced, each tube is assigned an integer 0 through 287. The numbering process begins on the right side[1] of the detector in the first layer at the bottom most photomultiplier tube and ends on the left side of the last layer on the top most photomultiplier tube. The parameter file is formatted as follows:

```
set voltage(0)     1000
set voltage(1)     1500
…
set voltage(287) 1000
set power(0)          0
set power(1)          1
…
set power(287)       0
set action(0)         1
set action(1)         1
…
set action(287)      0
```

Each array is a set of values for each of the 288 high voltage channels / photomultiplier tubes on MoNA. The voltage array simply holds the values of each channels voltage. The power array keeps the on/off (0/1) value of each channel. The action array is the most important of the three arrays because it determines which channels are to be changed by the C++ program when it is called. If the action array element for a photomultiplier tube channel is set to one then that channel will either have its voltage set, power set, or its status printed to the screen. If an action array element is zero then that channel is ignored, regardless of its corresponding values in the other two arrays.

HVTaskmaster.cpp

The central C++ function was given the name HVTaskmaster and was designed to have two command-line arguments. The first of these is the path of the Tcl parameter file. If this file is to be used by the program it must contain Tcl content of some sort, or it

[1] References to right and left on MoNA are made with respect to the direction of the beam line

will exit the program. The second argument is an operation flag -r, -p, or –v. The flag determines what function the program will be executing: read channel status, set power, or set voltage. The fact that HVTaskmaster is to be called by a Tcl/Tk GUI was taken into consideration and so for simplicity only one operation can be performed per program call. This program uses several functions to facilitate control over the power supplies, most of which are found in the file MoNAHVcontrol.cpp, which will be discussed shortly:

- **Param_readFile –** This function is used to open the Tcl parameter file for use by other Tcl to C++ functions in the file **param.cpp** written by Dr. Andreas Stolz.
- **ConvertTclArray –** Function that scans through the Tcl parameter file and copies array elements into a corresponding C++ array. This function also makes calls to functions in **param.cpp.**
- **Set –** A function that takes an action array and compiles a list of all the "active" channels and, with either a power array or a voltage array, sets the proper parameter.
- **HVStatus –** This function takes in only an action array and prints the present status (voltage, current, on/off) of the indicated channels to standard output.

MoNAHVcontrol.cpp

This file contains almost all the functions needed by HVTaskmaster to read from the Tcl parameter file and send commands to the power supplies through a TCP/IP connection. Some of these functions are called by other functions within the file to speed up and/or organize the progression of the data.

- **SelSortByParam –** This function is called by **Set** to perform a selection sort on the list of channels that are to be changed. The result is a list of channels grouped by their parameter value in descending order. This allows the **Set** function to write to the power supplies as few times as possible.
- **SetHVParam –** Function is called by **Set** and takes a list of channels with equal parameter values and organizes it in a form that can be used by the CAEN library functions. Then it makes a function call that sets all the channels to that value.
- **FillBoxSlotChannel –** A function that inputs a list of PMT addresses and passes back the location of those channels organized by power supply, slot number, and channel number. Called by **SetHVParam** and **HVStatus.**
- **WriteToChannels –** This function uses the CAEN library functions to connect to the power supplies and set the appropriate channels to the given value.

- **HVStatus –** Function is similar to **Set**, however instead of writing to the power supplies, the function is called to read from them and then the data is printed to standard output.
- **ReadStatus –** Called by **HVStatus** to read and store for each channel in the list: the channel name, monitored current and voltage, and whether or not the power is on.
- **PrintStatus –** This function is also called by **HVStatus**. It takes the stored values passed back by **ReadStatus** and prints them out to standard output (screen).

Practical Information

With the HVTaskmaster program in its present form, a Tcl/Tk GUI could be made that functioned properly and thus making an easy to use control interface with the high voltage power supplies. This setup might be sufficient for the current configuration of MoNA and its high voltage system, but changes can occur. The program and its structure should be able to adapt to the changes as long as they do not exceed the capabilities of the power supplies. If MoNA's configuration was going to change into 16 layers of 9 bars, or the addition of more detector modules, et cetera. In that case, if the new configuration yields the same number of bars, photomultiplier tubes, and high voltage cards then all that would need to be done would be to keep track of which bars moved. Now if the number of bars were to change, then there would be some changes to be made in the source code. Specifically, the numbering scheme used would have to be changed. Changing the value of the predefined constant MAX_LENGTH (which is currently set to 288) in the MoNAHVcontrol.h header file would alleviate most of the problems. The rest would have to do with changes in the Tcl parameter file which would obviously need to define a different number of elements for each array. In another situation, the number of high voltage cards in the rear of the power supplies might change. This problem has the same type of solution as before, a change of constants. The number of high voltage cards located in each power supply is stored in the constants RCARDS and LCARDS in MoNAHVcontrol.h. All of the functions were written under the assumption that the high voltage cards fill up the slots on the power supplies in sequential order. If a slot were to become damaged for some reason, then a workaround in the FillBoxSlotChannel function would have to be made (i.e always setting channel population array element for the bad slot to zero). In the case of one of the power supplies undergoing complete failure and cards are rearranged to fill up the other power supply, all three of the aforementioned changes would have to be implemented. There is also the potential for expanding the number of functions that HVTaskmaster can perform. The number of parameters that can be controlled is the only limit on the number of functions that the program can have (16 total). Not to mention the functions that could be used in a diagnostic capacity if there was a necessity for such analysis. There is a fair amount of flexibility in the program with the structure it has now.

Conclusion

This summer a problem was introduced, that is, whether a remote method of controlling the high voltage power supplies on MoNA could be developed. A solution was devised in the form of a C++ program that could perform the basic functions needed to operate the power supplies. The program can be properly implemented and has benefits over other control options. As I have come to understand, that is the least of what is expected of such a tool in physics: the ability to carry out its function and lighten the load. Not to state that the program will make the research any easier, but the scope of the program should make power control of the detector less of a concern. The program written also has the ability to expand and conform to the needs of MoNA, with respect to the power scheme of the detector. If at a future date, MoNA is used in a fashion that exceeds the adaptability of the program or a more efficient means of controlling the power supplies is developed; at that point the program will have served its purpose. In the interim, the current program should provide those people using MoNA the power control needed to use the detector effectively.

Bibliography

Bauman, Thomas. (June 2003). *MoNA Technical Information.* Retrieved from http://groups.nscl.msu.edu/mona/technical.html

Bauman, Thomas. (Oct 2002). *Project Overview*. Retrieved from http://groups.nscl.msu.edu/mona/overview.html

Fox, Ron. *NSCL Data Acquisition System Tutorial.* Retrieved from http://docs.nscl.msu.edu/daq/tutorial/indexl_files/frame.htm

Ousterhout, John K. (1994). *Tcl and the Tk Toolkit.* Reading, MA: Addison-Wesley.

Overland, Brian. (1999). *C++ in Plain English.* (2nd ed.). Foster City, CA: MT Books.

Welch, Brent B. (2000). *Practical Programming in Tcl/Tk.* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.