

A TEST SYSTEM FOR BEPCII POWER SUPPLY CONTROL

J.Liu, C.H.Wang, J.Zhao, IHEP, Beijing, P.R.China

Abstract

There will be about 400 magnet power supplies in the storage rings of the BEPCII. These power supplies will be controlled by the quantities of PSC/PSIs which were developed by BNL. These PSC/PSI modules need to be first tested in the lab after they are shipped to IHEP from the United States. So, we built a set of test system for the PSC/PSI modules which are based on the serial port communication of the PSC using LabVIEW. This paper describes the serial interface driver and a LabVIEW library of the PSC. The off-line and on-line testing are also reported here.

OVERVIEW

BEPCII will construct double rings in the current BEPC tunnel. There will be approximately 400 magnet power supplies in the storage rings and transport line. According to the requirement of control precision and stability of the magnet power supplies and the investigation on the front-end hardware, we decided to use a BNL-designed PSC/PSI (Power Supply Controller/Interface) to control high precision power supplies. It not only has a good performance and many advantages, but also is commercial products available now.

The PSC/PSI is a very simple integrated system that provides all functions (setpoint, readback, control and status) with a single board. The PSI will be installed in the power supply. The PSC may reside in the VME-64x crate or in a chassis. They are connected with a pair of fiber optic cables that provides electrical isolation. One PSC can control up to six PSIs. The PSI has one 16-bit analog output, four 16-bit analog inputs, fifteen digital commands, and sixteen digital readback. The connection between the PSI and the power supply is with two cables: one for the analog signals and another for the digital signals. This installation is very simple.

A PSC has a RS-232 interface and a VME interface. It can be accessed and controlled by VME bus and a serial port. In order to test the PSC/PSI modules, we developed the serial interface driver for the PSC/PSI and built a set of PSC/PSI standalone test system.

The test system as shown in the figure 1, consist of a laptop and a chassis PSC and a chassis PSI. The laptop

running LabVIEW is connected to the PSC via a serial port. The serial port driver and its application are developed by ourselves.



Figure 1: Standalone PSC Configuration.

SOFTWARE DEVELOPMENT

The serial interface protocol of the PSC

The electrical characteristics of serial signals of PSC are composed of a baud rate 38400, 8 data bits, 1 Start Bit, 1 Stop Bit, and no parity.

The serial interface is a character encoded protocol composed of an address register and a read and write data register. The address register is a 24 bit address pointer which can be set to any random address, or it can be incremented in byte or integer offsets. The read/write data register is 16 bit wide data register which is a latch containing the 16 or 8 bits of data after the read/write operating code is executed.

These basic building blocks allow full read/write random access to any memory type subsystem with complete independence from application. Figure 2 is the functional block diagram of the serial interface.

This serial interface protocol was embedded in an EPROM on the PSC board and couldn't be changed arbitrarily. There are four functions for communicating with a PSC via the serial port. They are `get_idata()` for getting integer data, `get_cdata()` for getting byte or char data, `put_idata()` for putting integer data, `put_cdata()` for putting byte or char data.

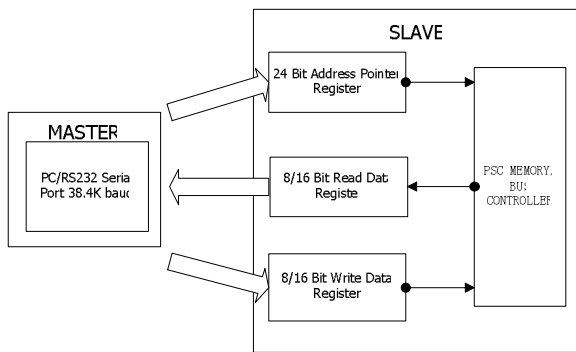


Figure 2: Functional block diagram.

All the functions mentioned above can call the basic serial communication functions on WIN32 platform, which are a initialization function ComInit(), a sending character function ComPut() and a receiving character function ComGet(). Then, they will call some WIN32 API functions. There are five common communication API functions for a serial port. They are CreateFile(), CloseHandle(), SetCommState(), WriteFile() and ReadFile(). CreateFile() function could open a HANDLE, which represents serial interface file. CloseHandle() function could close the HANDLE. SetCommState() function could configure the parameters of serial interface for communicating. WriteFile() function could send messages, ReadFile() function could receive messages.

Based on these functions, we wrote a DLL (dynamic link libraries) for developing LabVIEW library.

LabVIEW library development

Using the CLFN (calling library function node) subfunction of LabVIEW and the DLL file, we developed a LabVIEW library which includes 34 subVIs and supports all functions of the PSC/PSI via a simple serial port.

Some Block Diagram of basic communication subVI between the PSC and PC/LabVIEW is shown in the figure 3. At first, Init.vi is for the initialization of the PSC/PSI hardware. Then Set_spt.vi is responsible for sending out a Setpoint, Set_cmd.vi is responsible for sending out a Command. The Setpoint Readback can be read by Get_ADC.vi and the Command status can be read by Mem_His.vi.

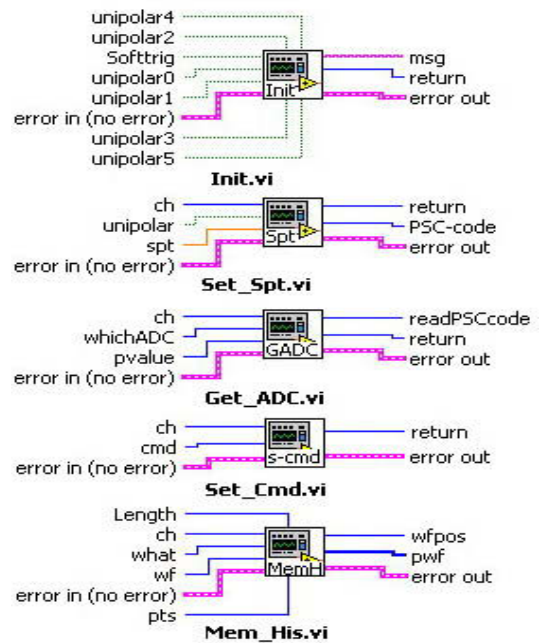


Figure 3: Parts of LabVIEW library.

Users can easily build their own applications to handle up to 6 power supplies based on the LabVIEW library.

OFF-LINE TEST

The off-line test includes digital signal output/input channels and analog signal output/input channels. We have developed applications for the off-line test, such as test programs and graphical man-machine interface, including control panels and parameter display windows.

Digital channel test

The connection between the command output channel and the status input channel is via a 37pin "D" type connector with pin jumpers. There are four commands button in the top left corner and status monitors in the right side as shown in the figure 4.



Figure 4: Six PSIs Digital channel test.

The PSC sends one command to six PSIs and reads all status of the PSIs at the same time.

Analog channel test

The connection between the setpoint output channel and the readback input channel is via a 9pin “D” type connector with pin jumpers. The figure 5 shows Setpoint and Readback. Setpoint and AverageTimes can be configured. There are four curves and data for setpoint Readbacks displayed with different colors in the right side of the figure.

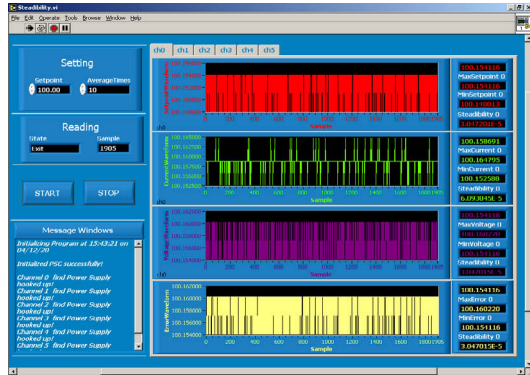


Figure 5: Analog channel test.

ON-LINE TEST

This system has also been tested with a proto type power supply including current setting and readback in normal mode.

The figure 6 shows the on-line stability testing result. There are four Command buttons and six Status lights in the top left corner. Two curves are displayed in the middle of the figure, the top one is Setpoint Readback, the bottom one is current Readback.

During on-line testing, we found the LabVIEW program can be only running in a lower rate because of the serial limited communication rate.



Figure 6: On-line test with a prototype power supply.

SUMMARIZE

The testing of the most PSC/PSI modules has been done. So far, the test system using LabVIEW and serial port driver works fine.

ACKNOWLEDGEMENT

We’d like to thank Apogee Lab for their providing the serial interface protocol and example routines. And we also would like to thank Mr. Ray Claflin III, Claflin Associates for his helpful discussions.

REFERENCE

- [1] Jayesh Naik, “Serial Interface For Debugging Memory Based Subsystems”, Apogee Labs, 01/11/2001
- [2] S.Peng, R.Lambiase, B.Oerter, J.Smith, “SNS Standard Power Supply Interface”, ICALEPCS’01, California, November 2001
- [3] R.Lambiase, B.Oerter, J.Smith, “Power Supply Control for the Spallation Neutron Source”, EPAC’00, Vienna, June 2000
- [4] C.H. Wang, J. Zhao, J. Liu, S. Huang, “A Proto type of the Power Supply Control System in BEPCII”, ICALEPCS’03