

SNS* DIAGNOSTICS TIMING INTEGRATION

C. Long, Innovative Design, Knoxville, USA

M. Sundaram, University of Tennessee, Knoxville, USA

J. Pogge, W. Blokland, D. Purcell, and D. Murphy,
Spallation Neutron Source, Oak Ridge, TN, USA

Abstract

The Spallation Neutron Source (SNS) accelerator systems will deliver a 1 GeV, 1.44 MW proton beam to a liquid mercury target for neutron scattering research. The accelerator complex consists of a 1 GeV linear accelerator, an accumulator ring and associated transport lines. The SNS diagnostics platform is mostly PC-based running Windows XP Embedded for its OS and LabVIEW as its programming language. Coordinating timing among the various diagnostics instruments with the generation of the beam pulse is a challenging task that we have chosen to divide into three phases. First, timing was derived from VME based systems. The second phase, described in this paper, timing pulses were generated by an in house designed, PCI timing card installed in ten diagnostics PCs. Because each timing card has up to four available triggers and we could duplicate triggers using a distribution fan out module, enough triggers were generated for all instruments. This paper describes how the Timing NAD (Network Attached Device) was rapidly developed using our NAD template, LabVIEW's PCI driver wizard, and LabVIEW Channel Access test screens. The NAD was successfully commissioned and has reliably provided triggers to the instruments. This work supports the coming third phase where every NAD will have its own timing card.

INTRODUCTION

One of the challenges with distributed diagnostic instrumentation as it exists at SNS is synchronizing the collection of data with the arrival of the beam pulse. Most diagnostic instruments in use at SNS are controlled by programs written in National Instruments LabVIEW software running on a windows based PC. The collection of the diagnostic instrument and hardware and the associated controlling PC is called a Network Attached Device (NAD). A NAD is connected to the accelerator network and communicates to the central control system using the Experimental Physics and Industrial Control System (EPICS) data transfer protocol. The timing system at SNS supplies two types of data. The Event Link (EL) specifies events as they happen during the beam generation cycle. These include system format events (CYCLE START) and functional events (BEAM ON). Trigger pulses generated by a timing card are generally specified as the event to fire based on, and the

delay after the event to fire. The Real Time Data Link (RTDL) contains data about the pulse such as a time stamp unique to that pulse, and system level information, such as (once it is completed) ring frequency.

During the front end commissioning run of the LINAC section of SNS, VME based timing cards were used to decode the timing signal and generate a trigger pulses. This was phase one of a three phase approach to diagnostics timing at SNS. The NAD only received a TTL pulse to trigger its data from the VME timing. None of the other information, such as timestamps, was used. The timestamp is needed to allow later analysis to compare measurements made on different instruments along the accelerator during the same pulse event. Since diagnostic instruments at SNS are all PC based, it was decided to develop a PCI based timing card, to be installed in each individual NAD. This way the NADs don't share a single point of failure, and can have access to the RTDL data, including timestamps. The production PCI based timing card was not complete at the time of the Fall 2004 commissioning run, but several prototypes were available. While there were not enough to install in each NAD, using signal fan outs it was possible to at least replace the functionality of the VME timing cards and test the prototypes during the Fall 2004 commissioning run. This use of the prototypes was phase two of timing at SNS, leading to an eventual third phase when each NAD will have its own PCI based timing card installed.

IMPLEMENTATION

The prototype PCI based timing card has 4 separately programmable triggers.

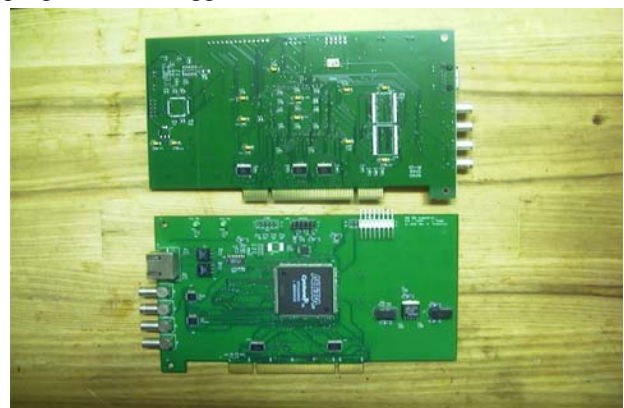


Figure 1: Prototype Board showing Timing signal input and four trigger outputs.

This will allow flexibility when installed on a single NAD in phase 3, but was very useful during phase 2 as each card could supply 4 separate and distinct triggers.

* SNS is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy. SNS is a partnership of six national laboratories: Argonne, Brookhaven, Jefferson, Lawrence Berkeley, Los Alamos and Oak Ridge.

Before the control software could be completed, another decision had to be made, how to interface over the PCI bus with the timing card. Other SNS PCI projects completed at partner labs had used the JUNGO package to develop a .dll driver file that communicated with the hardware, and then made calls to that .dll file. For this application, it was decided to use the National Instruments VISA wizard to generate an interface to the PCI timing card. This reduced the number of software development packages that needed to be procured and maintained, since the wizard is part of an installation of LabVIEW. This proved to be an efficient and cost effective decision. A software engineer during his first day on the project, and with no previous experience with the wizard, was able to generate a working test application to verify the usability of the wizard and the PCI communication to the card in just a few hours.

Using the interface generated by the VISA wizard, a library of timing card control functions was developed in LabVIEW. These provided the various low level timing functions that are necessary to build a full application, such as, initialize card, change delay to *n.nnn* on channel *x*, read back polarity setting for channel *y*, etc.

The requirements for the timing NAD control software were fairly straightforward. For each of the for channels

on the board, a local user should be able to change the basic trigger settings, i.e. what EL event to trigger off of, what delay from the event to wait before triggering, the width of the generated pulse, the polarity of the generated pulse, and the ability to enable/disable that individual channel. A remote user needs to have access to the same functionality described above, but from the central control system. Finally, the system should keep its configuration in the event of a power down / restart event.

Control software on the Diagnostic NADs at SNS is written in National Instruments LabVIEW software. There is an existing bare bones template for NAD control software that provides some of the basic functionality needed to meet the requirements. The template is state machine based, and is used as the basis for all diagnostic instrument NAD software. This helps with software maintenance as the software engineers are familiar with the layout and basic configuration of all NADs.

The local settings section of the requirements was the most straight forward to implement. The individual LabVIEW controls that represent the various settings and readbacks on the different channels of the card were simply connected to the appropriate low level timing library functions, as shown in Figure 2.

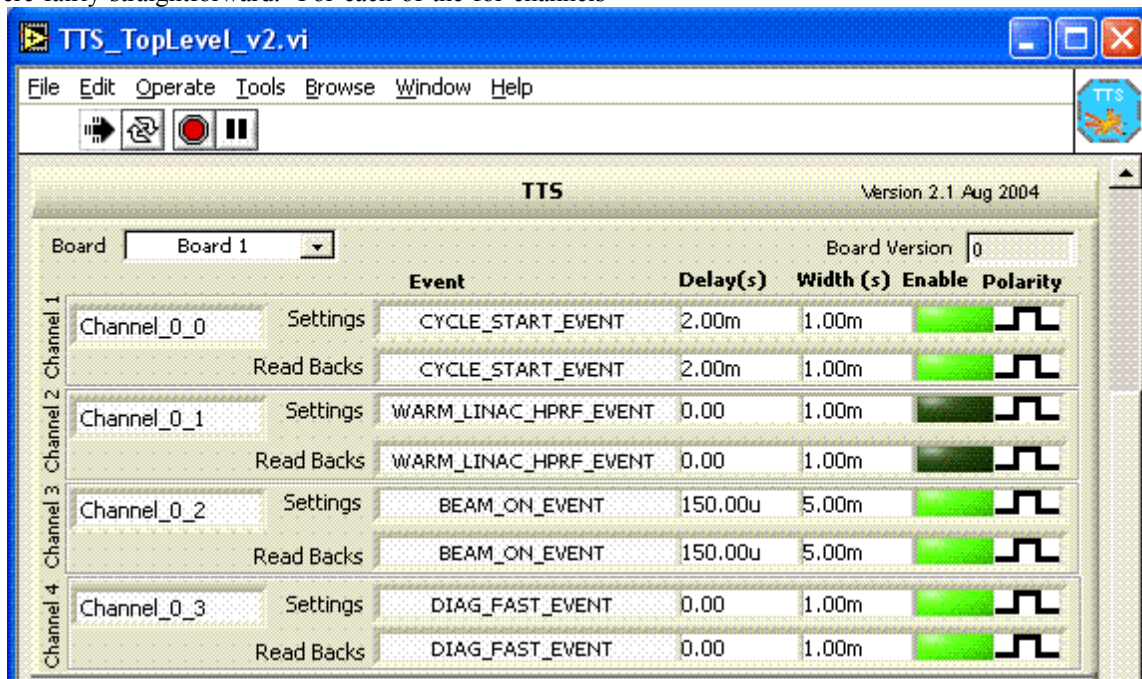


Figure 2: Local User interface Screen detail.

This allowed a local user of the machine, or someone who had transferred to the machine using remote desktop to change the settings for each individual channel on the card.

A remote user of the central control system needs access to the same functionality as a local user. At SNS communication between NADs and the central control system uses the EPICS protocol. The NAD template supplies the basic framework for using this protocol to

communicate with the central control system. All that is necessary is to assign EPICS names to certain variables that the remote user is to have access to and then link those names to existing controls and indicators on the local user screen. The central control system developer then maps those EPICS variables to functions on the central control program. See figure 3 for an example of the central timing control screen, that includes a popup of the control for an individual trigger channel.

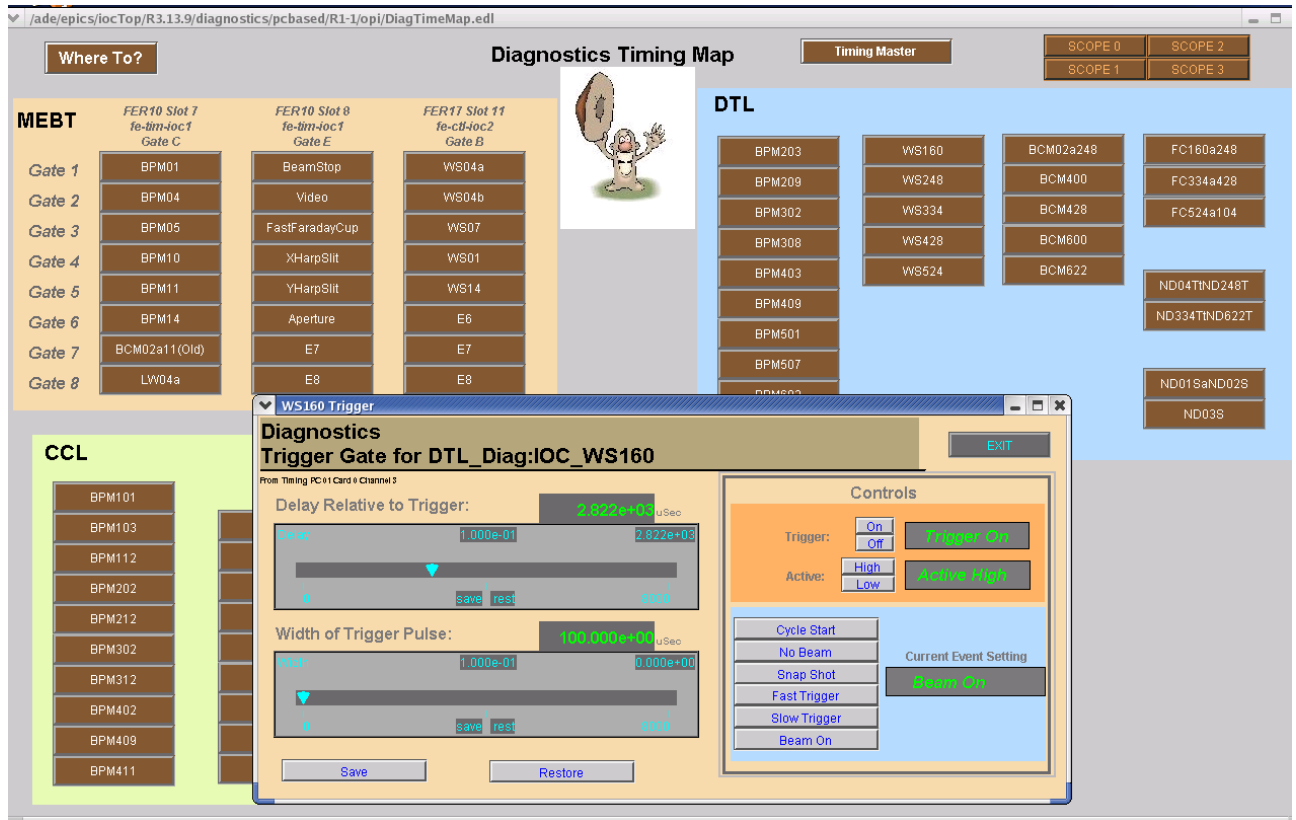


Figure 3: Central Control User Screen

The third requirement, saving configuration information was completed in a simple way. When the timing NAD boots, it reads its configuration information from a file. Timing configuration changes are relatively rare (at most 10s per day) and today's disks are huge, so rather than simply overwriting the exiting configuration file when a change is made, the existing configuration file is renamed to a name that includes the current date and time to show when the file was valid through, and then it is moved to a storage directory. Then a new version of the configuration file is written, with the new settings included.

FUTURE

The next phase of timing control as SNS plans to put a production PCI timing card into each diagnostic NAD. Rather than having a separate control program for timing and the specific instrument controlled by the NAD, timing control will be integrated as part of the basic NAD

template, complete with already configured EPICS controls, to allow remote user control of NAD timing.

CONCLUSIONS

The total control software development time for the phase 2 timing system was on the order of three weeks. The combination of existing NAD template to build on, the ease of the development using the VISA wizard as opposed to other methods, and the speed of LabVIEW software development in general was enough to demonstrate the power and versatility of this development method.

REFERENCES

- [1] R. Nelson et al, "SNS Timing System," ICALEPCS 01, San Jose, CA
www.slac.stanford.edu/econf/C011127/FRAT001.pdf.