NETWORK ATTACHED DEVICES AT SNS*

W. Blokland and T. Shea, ORNL, Oak Ridge, TN, USA
M. Stettler, LANL, Los Alamos, NM, USA

Abstract

The Spallation Neutron Source (SNS) diagnostic instruments at Oak Ridge National Laboratory are based on the Network Attached Device (NAD) concept. Each pickup or sensor has its own resources, such as networking, timing, data acquisition, and processing. NADs function independently thus reducing the brittleness inherent in tightly coupled systems.

This paper describes our implementation of the nearly 400 NADs to be deployed. The hardware consists of rack-mounted PCs with standard motherboards and PCI data-acquisition boards. The software suite is based on LabVIEW and EPICS, communicating through a shared memory interface. LabVIEW supports the agile development demanded by modern diagnostic systems. EPICS is the control system standard for the entire SNS facility. Program templates and documentation tools are available to the programmer. SNS diagnostics are developed by a multi-laboratory partnership, including ORNL, BNL, LANL, and LBNL. The NAD concept proved successful during the commissioning of the SNS front-end both at LBNL and ORNL.

INTRODUCTION

The basic idea behind a Network Attached Device is to implement an instrument as a single networked device with its own resources [1].

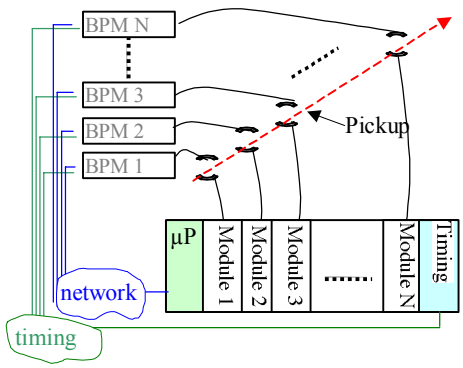

Figure 1. The NAD versus VME configuration.

For example, a typical VME implementation of a Beam Position Monitor (BPM) handles many pickups per crate, while the NAD implementation assigns each pickup its own resources such as a processor, a timing decoder, and a network interface, to make one independent device, see Figure 1. To create a set of BPMs, the NAD implementation makes copies of a single device. This leads to simpler software and reduces failure interaction. The VME implementation has to deal with managing shared resources among the multiple BPMs. If a system is expanded, a separate integration test is needed to check the interaction of the larger set of modules with each other and the shared resources. The NAD devices don't interact and no new testing is needed when adding NADs. A failure or required maintenance in one component of a VME implementation would most likely bring down all BPMs within the crate. In the NAD implementation each device is independent and any failure in or maintenance to a component would only affect that one device, thus limiting the scope of the outage. In most cases that means that the accelerator can continue to operate.

IMPLEMENTATION

The NAD can be implemented in many ways. In its ultimate form it would likely resemble a system-on-a-chip with a sensor. However, to make use of the wide variety of development and management software and low cost computer and data-acquisition hardware, we have chosen to base the NAD on a PC-based system. The PC-based systems consist of standard motherboards, rack-mounted for easy installation in the field, see Figure 2.

Figure 2. A rack mount PC.

LabVIEW is chosen as the main software development environment operating under Windows 2000 or XP. LabVIEW has a very well integrated visual development environment for data-acquisition and signal processing. Many vendors supply LabVIEW drivers with their hardware. Combined with the SNS software suite, described below, the SNS collaboration can efficiently implement the NADs.

* The Spallation Neutron Source (SNS) project is a partnership of six U.S. Department of Energy Laboratories: Argonne National Laboratory, Brookhaven National Laboratory, Thomas Jefferson National Accelerator Facility, Los Alamos National Laboratory, Lawrence Berkeley National Laboratory, and Oak Ridge National Laboratory. SNS is managed by UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy.

SOFTWARE SUITE

Especially because the software development is a multi-laboratory effort, it is important that all software is similarly structured. This will enable the small Diagnostics Group at SNS to maintain and upgrade software from the other labs and also efficiently write software for the locally implemented projects, such as the Laserwire. A software suite provides that common structure. The software suite supports the instantiating of the software for a NAD to easily create a set of NADs. Each item of the software suite is discussed in the following sections.

EPICS IOC and Shared Memory Interface

The SNS control system uses EPICS. The NADs' first version of the EPICS interface was based on the ActiveX Channel Access Server interface by LANL [2]. The NADs switched to the full IOC when this became available for Windows. The IOC has a more mature and well supported code base. While early LabVIEW code had a shared memory interface to the ActiveX interface, the new shared memory interface connects LabVIEW directly to the IOC. The shared memory interface and IOC combination increased performance and reliability over the ActiveX interface. On the LabVIEW side, a call to write a 100 doubles to the shared memory interface, including buffering, takes about 0.03 milliseconds on an 800Mhz P3, about 7 times faster. 100 floats generated and communicated to a remote Channel Access client at 1000Hz take less than 5% time of the same CPU. More performance data will be presented in [3].

The shared memory interface implements functions to

- create, find and destroy variables,
- read from and write to variables,
- set and receive events, and
- retrieve information about variables.

All these functions can be called from LabVIEW. The events are associated with EPICS interrupts. For example, if the LabVIEW program needs to know whether an output PV has been set, an associated interrupt awakens a LabVIEW task. This task then queues a message to any other task to schedule any action that needs to be taken. This method avoids the inefficiency of the polling. The IOC initializes the Shared Memory Interface using the EPICS database (.db) syntax. After LabVIEW has started the IOC, the IOC reads in a .db file and, while creating the PVs, also creates the shared memory variables. A LabVIEW routine then obtains a reference for each variable to use with the shared memory functions. This same routine is used to automatically generate the .db file. This way the programmer does not need detailed knowledge about EPICS database syntax. The utility also generates the command file to start EPICS and a table for documentation purposes, see Figure 3. The macro substitution feature of the database files is exploited to keep the PV names the same in each NADs LabVIEW program but different in each NAD's IOC.

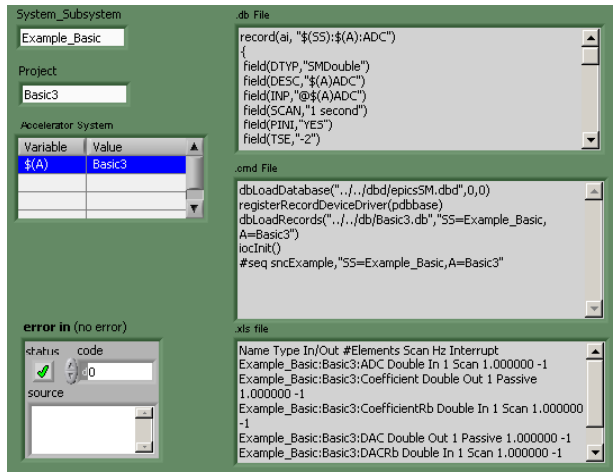

Figure 3. The utility that generates a .db file.

Programming Template

The template concept implements the common software structure for all NADs. Not only will the common structure ease the maintenance, it also gives the programmer a head start. A typical template implements

- multiple tasks,
- queues to synchronize and communicate between tasks,
- state machines to organize program execution.
- a LabVIEW task to process shared memory interface events,
- PV referencing to automatically generate db files,
- error handling,
- configuration file setup, and
- user interface using the event structure.

The front-panel of a template example with a continuous cycling task is shown in Figure 4.

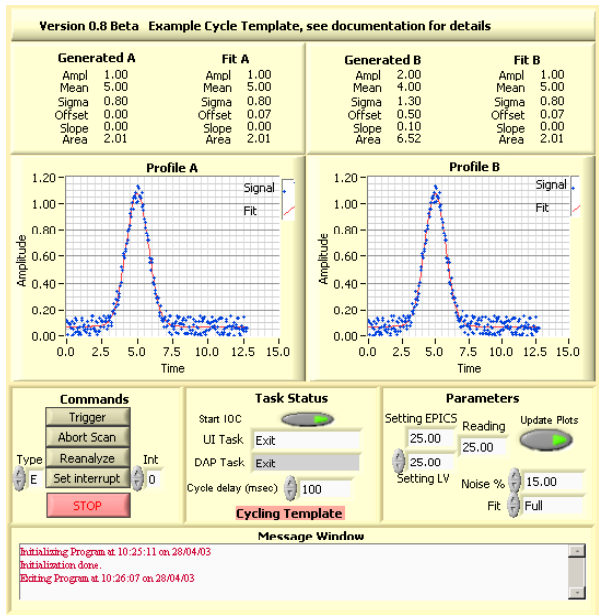

Figure 4. Front panel of the Cycle Template VI.

Various programs using the template structure have been made and are available as examples. A tutorial helps the programmer understand the features of the template. The tutorial consists of a series of programs that start out very basic but successively add features to arrive at the program structure of the template.

Documentation and Development Tool

Each item of the software suite is described in the Style Guide, and this is the place for the programmer to start. The Style Guide includes guidelines for commenting the LabVIEW program using the built-in description fields. The documentation and development tool VIHierarchy, written by the author at Fermilab, uses these comments to create a framed HTML document of the whole hierarchy of the program, see Figure 5. VIHierarchy also provides tools to clean up VI libraries or directories.

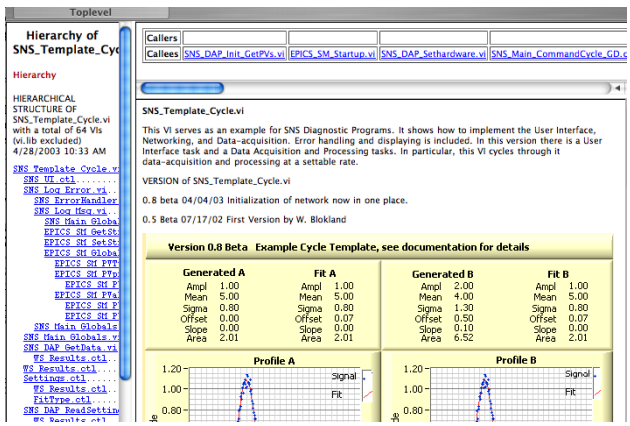

Figure 5. HTML page generated by VIHierarchy

Testing of NAD

The LabVIEW version of the ActiveX Channel Access client enables the testing of the control logic of the NADS by reading and setting the EPICS PVs on the development computer. Clients have been implemented and are part of the tutorial and template.

EXPERIENCE

Early versions of NADs have been implemented by LANL and BNL in the form of Wire Scanners, Beam Position Monitors, and Beam Current Monitors [4]. The NADs functioned well during commissioning of the MEBT at Berkeley and ORNL. Some downtime occurred due to a problem with the shared timing system. The PCI timing card was not ready yet and triggers for all BPMs were taken from one VME crate. When this crate was reset, because another device within the crate had to be reset, the timers were not restored. All BPMs then failed to trigger at the right time and did not give correct positions. If the BPM had been fully NAD compliant and had its own timing card, this failure mode would have been avoided.

NADs based on the template have been running without problems. One NAD is in use by the Accelerator Physics group to test the speed of the Java console applications. Others have been created for new MEBT diagnostics and future installations, such as the D-Plate.

FUTURE

To manage the hundreds of NADs with a small diagnostics group we are in the process of selecting a PC management package, such as SMS or Altiris. Our strategy is to use the Oracle database as the one depository for all files to be installed on a NAD and to use the PC management package to set up and maintain our NADs. The management package will also do remote monitoring to diagnose a NAD and perform inventory tracking. Replacing or building the software on a NAD is to become a push button operation.

At this point, we have used Windows 2000 or XP and not XP embedded with a real-time extension. XP embedded would give a smaller OS but would require the effort of customization. We have not yet needed the smaller OS size or the real-time attributes

A new Channel Access Client for LabVIEW is under development to be portable among the EPICS and LabVIEW supported operating systems.

SUMMARY

This paper presented the implementation of Network Attached Devices at SNS. The hardware is based on rack mounted PC (x86) motherboards. The software is based on LabVIEW, EPICS IOC, and Windows 2000 or XP. A software suite has been created to interface LabVIEW to EPICS, to provide a common program structure, to help document, and to assist development of NADs by several laboratories. The NAD concept of using a set of independent devices to implement an instrument system proved itself during the commissioning of the MEBT. Over 10 NADs have been commissioned and 30 more are being installed for commissioning this summer.

REFERENCES

[1] T. J. Shea et al," SNS Accelerator Diagnostics: Progress and Challenges," pp 512-16 PAC 2001, Chicago, IL, USA, June 18-22, 2001.

[2] K.U. Kasemir, "ActiveX: CA client and server for Active-X programs," at http://www.aps.anl.gov/epic/extensions/index.php.

[3] D. Thompson and W. Blokland, "A Shared Memory Interface between LabVIEW and EPICS," to be published at ICALEPCS 2003, Gyeongju, Korea, October 13-17, 2003.

[4] Mike Plum, "Diagnostic challenges at SNS," this conference, Mainz, Germany, 5-7 May 2003.