

THE EPICS UPGRADE AT THE NSCL

J. Priller, L. Foth

Introduction

This paper will describe the upgrade of the NSCL control system to EPICS (Experimental Physics and Industrial Control System).

The Previous Control System

The previous control system at NSCL was based on distributed VME crates with Motorola 68000-series CPU boards, connected by an Arcnet network with a bandwidth of 2Mbps. These crates used control system software developed at FNAL and the system saw use from 1984 until the shutdown for NSCL's Coupled Cyclotron Project in the summer of 1999.

Local control consoles consisting of a keyboard, a knob, buttons and a small black & white text display were initially the only user interface available. VAX/VMS workstations and Intel x86 PCs equipped with Arcnet interface cards were added in the late 1980s. Software development for the VME crates was primarily done in assembly language, with a small amount done in Pascal and later in C. Workstation software was developed in C, Pascal and FORTRAN.

This system worked well enough over the 15 years it was in service. But towards the end of its lifetime it became difficult to obtain many of the VME boards used, and with faster and more capable workstations allowing operators to monitor more channels with more sophisticated software, limits in the system became apparent. To handle the increased complexity of the coupled cyclotron system a new control system was obviously needed, preferably one that would allow us to re-use much of our existing VME hardware.

Decisions, Decisions

After considering several real-time operating systems available for VME, and after briefly entertaining the notion of writing our own control system on top of one of these, EPICS on top of Wind River's VxWorks operating system was finally decided upon. VxWorks is widely used in the industrial control and automation field, and EPICS widely used in experimental physics and astronomy. The existing EPICS community was very helpful in our initial investigation and deliberation phase in pointing out the strengths and weaknesses of the EPICS system, and in giving us an idea of the work we faced in converting to it.

Our next task was choosing a new VME CPU board. Boards based on Motorola's PowerPC and newer 68000-series processors were considered, in addition to boards based on Intel's x86 family. We selected a Motorola PowerPC board (in particular, the 32Mbyte 200Mhz Motorola MVME2302) as PowerPC offered the best performance for the price of the three contending architectures.

We also had to select the development platform to use for VxWorks, from between Windows NT and UNIX. The control system at NSCL had for over a decade been using Windows-based PCs for its operator interface workstations, and we had little or no recent experience with UNIX. At that time the Windows NT option was a relatively new and untried offering from Wind River, however, and the EPICS community had no experience working with it. The UNIX flavor of VxWorks development environment was ultimately chosen, and a Sun Sparc workstation purchased to run it on.

Initial Installation

EPICS was rumored to have a rather steep learning curve, and more than one person we spoke to in the EPICS community suggested that we borrow an expert from another EPICS site to assist us in getting our

first system up and running. We decided to take this advice, and were extremely fortunate in borrowing Johnny Tang for a week from Karen White's group at TJNAF. We would like to thank both of these individuals for this assistance, without which we would have had a far more lengthy and difficult time getting our first system off the ground. In the week we had this expert help we were able to install and configure VxWorks, EPICS and a host of needed GNU tools, get our first VME crate (referred to in EPICS as an IOC, for I/O controller) up and running and also complete an EPICS driver for a VME A/D board used at NSCL that was not included in the EPICS base software.

First Production System

With our first test system running, the next step was bringing up an actual production system in place of an old-style IOC in the field. We selected the IOC controlling the main magnet of our K500 cyclotron, as this provided an opportunity to test almost every kind of component we would eventually use in IOCs throughout the lab (analog, binary and serial I/O, as well as a specialized control application for ramping and regulating the magnet currents). This test went extremely well, and we were finally assured that EPICS could fulfill the control system needs at NSCL.

Additional Hardware Purchased

In addition to CPU boards, other new hardware was purchased in the course of our EPICS upgrade.

Separate Ethernet Network

EPICS uses TCP/IP for communications between IOCs and user workstations, so an additional Ethernet network was installed. We wanted a network separate from the lab's existing Ethernet network, both to keep traffic on one from bogging down the other, and also to make it easier to protect our IOCs from malevolent intent by placing routers between their network and that of the lab so that only legitimate NSCL users could communicate with them.

Local Boot Devices

One of the downsides of using EPICS is that VxWorks IOCs normally boot up from a small kernel stored in the CPU board's flash ROM, and then proceed to download the remainder of the operating system over the network from the UNIX development host. If all the IOCs were to try to boot simultaneously, as might happen after a building-wide power failure, the resulting contention for these boot files might keep the entire system from coming up for 20 minutes or more. Our solution to this problem was to purchase Micro Memory PCMCIA-card adapters for each IOC's CPU board, and to outfit these with a 20Mbyte ATA flash ROM disk to hold the required boot files. Each IOC can then boot independently of all the others, and also boot somewhat faster (in approximately 60 seconds, as opposed to 90 seconds for a network boot).

Serial Boards

NSCL has a number of internally designed serial devices in its control system, such as beam-current monitors, motor controllers and power supply controllers. In the previous control system a separate serial board with its own software and CPU serviced these, but the board we were using had since become obsolete. A new serial board meeting our requirements was found in MacroLink's MVCQuick board, which has its own Motorola ColdFire CPU, can be programmed in C/C++ using the same GNU compiler used by VxWorks and EPICS, and can be dynamically reloaded with software from the main CPU board. Moving to these boards necessitated purchasing additional backplanes for our IOCs with serial boards, as our previous system required only the J1 backplane and the MVCQuick board requires both J1 and J2 backplanes to operate in 32-bit addressing mode.

Additional Software Written

The EPICS base software provided much of the functionality we required for our upgrade, but for some things we found it necessary or desirable to write our own extensions to it.

Serial Board Software

NSCL's internally designed serial devices all communicate using a specialized protocol also developed here. In order to offload as much of the protocol overhead as possible from the IOC's main CPU board, code was written for the MacroLink MVCQuick board to handle the protocol details and VxWorks software written for the main CPU board to easily and efficiently handle communication between the two.

Specialized Record Types

Out of the box EPICS supports a number of standard I/O record types, such as analog and binary inputs and outputs. We had to code new specialized record types for all our internally designed serial devices, and also added records types for a combination analog input/output, a generic power supply record with on/off and interlock/ok status fields, and a generic magnet record with built-in current-to-field calculation and the ability to set the magnet's current by specifying the desired field.

Restoring Settings on IOC Reboot

One drawback of EPICS over our previous control system is that device settings are lost if the IOC is rebooted, at which time all outputs are generally reset to zero. Software to allow storing settings as the IOC runs and restoring them on reboot had already been written for EPICS by others, but at the time we evaluated it this software required storing the settings on the UNIX development host. In order to make our IOCs as independent of the UNIX host as possible, we wrote software to save settings to a battery-backed memory board in the IOC and then read these values back and reassert them after a reboot. No new memory cards had to be purchased, as our old control system used such boards to store its executable code and data.

Re-writing Console Control Applications

Over the years that the old control system was in service a good number of user console control applications were written to allow the operators to monitor and tune the accelerators. These needed to be converted from the old system to EPICS. All were object-oriented and had been written around the idea of a central data-serving object, so the conversion process was not difficult or overly time-consuming.

Serving Modicon PLC Data

VME IOCs form only one part of the control system at NSCL, some analog and binary I/O and safety interlocking requirements are provided through Ethernet-based Modicon PLCs. Channel access server software was written to make the data in these PLCs available as ordinary EPICS channels, using either a dedicated VxWorks IOC or a Windows NT PC as the server platform.

Serving Data from the Old Control System

One difficulty we faced in our upgrade process was how to provide seamless access to both the old Arcnet system and the new EPICS system as IOCs were upgraded one by one and the old control applications were replaced by their new EPICS-based versions. Our solution was to write channel access server software that made channels in the old Arcnet system available as EPICS channels, using dedicated Windows NT PCs as the servers.

Specialized Control Applications

NSCL has a number of specialized control applications that due to their complexity ran as system services on VMS or Windows NT workstations in our previous control system. The power and flexibility of VxWorks and EPICS allowed us to re-write these tasks to run directly on the IOCs involved, eliminating several points of failure.

Upgrade Time Line

April 1998	EPICS selected for NSCL's control system upgrade.
June 1998	Funding for upgrade project approved.
August 1998	Expert borrowed from TJNAF, first test system up and running, work begins on additional software needed for production IOCs.
February 1999	First production system (K500 main magnet) brought on-line.
August 1999	Start of Coupled Cyclotron Project shutdown, upgrade of NSCL's approximately 20 remaining production IOCs begins in earnest.
October 2000	Projected end of EPICS upgrade.
January 2001	Projected end of Coupled Cyclotron Project shutdown, commissioning of coupled system to begin.

The Pluses

In our upgrade project we've found EPICS to have these advantages over our previous control system:

- Almost all coding is done in C/C++ rather than assembly language.
- VxWorks provides a fully featured real-time operating system with built-in networking and multitasking support, making it easy to write specialized control applications that run directly on the IOC.
- 100Mbps Ethernet provides much greater throughput than the previous 2Mbps Arcnet.
- EPICS is widely used in the experimental physics community, which does a good job in our experience of providing support and advice to EPICS start-ups such as ourselves.

The Minuses

There have been a few difficulties in converting to EPICS, and as obsolete as our previous system was it had some capabilities that are missed:

- EPICS has a steep learning curve. Even starting with years of control system experience, it took us several months to master just the portions of EPICS that we use (and it has many features that we do not use).
- Selecting UNIX, an operating system we were not very familiar with as our development platform made a number of things about the project more difficult and time consuming, as we were forced to learn its intricacies at the same time we learned EPICS.
- The PowerPC architecture was a relative newcomer to the EPICS scene, and few other sites at the time had any experience in it. There have been a few PowerPC "gotchas" encountered as we've gone along, and though most all have been dealt with they've added to the frustration factor present in converting to any new system.
- The PowerPC CPU board we selected has one architectural drawback, in that its PCI-bus to VME-bus bridge circuitry is somewhat slow. This causes a greater CPU load in communicating with other boards on the VME bus, but has no other real effect.
- A VxWorks IOC takes far longer to boot up than our previous system, which kept its program and data in battery-backed memory and could reboot and reassert its settings in under one second. This has forced

us to add PLC protection for some devices that cannot be left uncontrolled for the 60 to 90 seconds it takes their IOC to boot.

Conclusions

In the end we would definitely conclude that EPICS was a good choice for our new control system, though there have been difficulties, large and small, encountered along the way. Should we have the whole thing to do over again, the only change we would likely make would be to select a VME CPU board with somewhat more efficient VME-bus circuitry.

Our advice to any other facility contemplating a switch to EPICS would be contact people in the EPICS community well beforehand and solicit their input. We found this community to be very helpful in our initial planning stages, and they have continued to be of great assistance to us as our project progresses.