

## INEXPENSIVE IOCS FOR GPIB SUPPORT IN THE TRIUMF/ISAC CONTROL SYSTEM

R. Nussbaumer, G. Waters, TRIUMF  
4004 Wesbrook Mall, Vancouver, B.C., V6T 2A3, Canada

### Abstract

The EPICS control system for the TRIUMF/ISAC Radioactive Beam Facility has to support an increasing number of distributed instrument clusters, which are read and controlled via the GPIB bus. The standard VME-based IOCs of the ISAC control system can address this problem only with a large cost overhead. As a much more cost-effective solution, dedicated EPICS IOCs were implemented using Intel x86 CPUs and GPIB controller cards in a PC/104 form factor using vxWorks as a real time operating system. The PC oriented architecture has the benefit of being well supported by software and hardware vendors. Device driver software written for other operating system platforms is readily available for porting. From several available EPICS GPIB support packages, the streamDevice package [1] was chosen, which provides a convenient, flexible, and extensible way of supporting GPIB peripherals. Providing support as dedicated 'GPIB Appliances' in a small package allows the control system to easily move with peripherals that are often portable and frequently taken offline for use as stand-alone instruments.

### INTRODUCTION

Integration of GPIB attached instruments in a control system poses some difficult challenges. Instruments are usually intended for independent off-line use, and may be removed from service in the control system in an unsynchronized way. GPIB instruments frequently take measurements that require lengthy time periods, or are delayed pending some type of trigger condition, making real-time data updates difficult. VME hosted IOCs are frequently distant from measured signals or transducers. Proliferating VME hosted IOCs to satisfy the IEEE-488 cable length restrictions is prohibitively expensive. Finally, communication with GPIB instruments requires a high level of intelligence on the part of the host CPU due to the more conversational nature of message-based communication.

For these reasons, deployment of GPIB-interfaced instruments as part of the ISAC control system has been difficult. Previous implementations [2] have included the use of National Instruments (NI) Labview running on Solaris workstations, as well as purpose-built driver software to communicate with GPIB hardware attached to NI GPIB-Enet interfaces. An attempt was made to write driver software for the GPIB-Enet as an EPICS device support layer, for use in conventional VME based IOCs, but was never completed.

The present strategy is to use small IOCs, each dedicated to a GPIB instrument cluster. This helps to

isolate the instruments from the overall control system, lessening the impact of spontaneous disconnects. The small footprint allows the GPIB interface to be located and move with the instruments. Reconfiguration of the EPICS database to reflect changing measurement requirements can be done without disturbing the rest of the control system.

### HARDWARE

The IOCs used are ZF-micro based PC/104 form factor CPUs. The GPIB hardware used is:

- NI GPIB-Enet10 interfaces
- NI GPIB-PC/104 interfaces

The PC/104+ standard provides an ISA-bus compatible expansion bus, as well as a PCI compatible interface bus. Each of these buses is implemented using a stacking pin and socket style of connector. This provides a more rugged connection and smaller physical footprint than a standard desktop PC motherboard. The small form factor allows the combination of CPU and GPIB interface to be positioned in close physical proximity to GPIB test and measurement instruments.



Figure 1: The GPIB Appliance: PC/104 CPU with GPIB interface mounted on TRIUMF backplane.

### CPU

The MZ-104+ CPU's are industry standard PC/104 form factor motherboards. The ZF-micro 'PC-on a Chip' provides all the basic functionality of an industry standard PC architecture. The CPU purchase price included a VxWorks board-support-package and runtime license, enabling easy support of EPICS control system software.

The ubiquitous nature of the PC architecture carries with it a large body of mature development tools, as well as a vast knowledge base accessible through the Internet and printed publications. The PC/104 standard implies a

relatively consistent level of functionality across manufacturer product lines, ensuring long-term availability of compatible hardware to support the GPIB appliance.

For development work, the PC/104 based system can be equipped with standard off-the-shelf expansion peripherals such as video adapter and disk storage devices. This was found to be a useful capability during the development of a VxWorks loader module.

A wide array of data acquisition and control peripherals conforming to PC/104 standard architecture exists, and is frequently more economical than a VME based counterpart. A PC/104 digital IO module will be used in conjunction with a CPU+GPIB package for controlling specialized in-house designed equipment.

A more detailed description of the MZ-104+ is the subject of another paper [3] presented at this conference.

### *GPIB-Enet10*

All initial deployment of GPIB instruments was accomplished using the NI GPIB-Enet10 interface. This device attaches to the GPIB bus and has a 10Mb/s ethernet port. A TCP/IP connection to the device allows reading and writing to attached instruments. An advantage of using this style of interface is that it can be physically situated near the instrument(s), and does not need to be near the host computer.

The ethernet interface to the host computer presents a more loosely coupled interface than is traditionally available on conventional GPIB expansion adapters. Of particular note is the inability of the device to support asynchronous SRQ signaling to the host CPU. This is one of the chief disadvantages to using the devices in an EPICS-based control system.

### *GPIB PC-104*

The GPIB-PC/104 manufactured by NI is the principle GPIB interface that we want to use at TRIUMF. It is a full Talker-Listener-Controller capable interface, based on the NI AT-TNT-488C chipset. The board interfaces with the CPU via the ISA-bus compatible PC/104 connector. Because the PC/104 interface electrically mimics the ISA bus, the board can be used in a standard ISA bus PC for testing and development, by using a connector adapter. The standard suite of NI test and diagnostic software for MS-DOS and Windows was therefore available for troubleshooting. This was found to be particularly useful when troubleshooting a board that was received dead on arrival from the manufacturer, and no known-good test software had yet been developed.

A custom TRIUMF backplane provides physical mounting, power supply, and electrical connections to the PC/104 CPU. Also mounted on the backplane is a RS-232 to Ethernet converter module, which allows TCP/IP connection to the EPICS/VxWorks console interface. The entire package mounts on an industry standard DIN rail.

## SOFTWARE

Two layers of pre-existing software were combined with a small layer of custom software to provide device and driver support of GPIB instruments and GPIB interfaces. At the lower level are vendor supplied driver packages for GPIB interface hardware. At a higher level is the Stream Device layer, which provides device support for EPICS.

### *NI Driver Packages*

NI provides source code driver packages for each of the two GPIB interfaces described herein. These packages are generic enough to be compiled and run directly on most platforms, including the target VxWorks operating system that supports EPICS.

The API provided by the driver packages closely follows the conventions of other NI software packages for accessing GPIB devices, so anyone with experience using the MS-DOS, Windows or Solaris hosted GPIB support packages will find the functionality very familiar. Additionally, the GPIB-Enet support package includes the standard NI *ibic* utility for interactive control of the GPIB bus and instruments.

Initially, it was expected that device support software would be written from scratch, possibly based on existing open source code, such as the Linux device driver. However, it was found to be cost effective to buy a fully debugged and functional driver package from NI. The availability of this package had a major impact on the overall success of the project.

The NI software packages are not freely distributable in the sense of most other EPICS software. They are, however, complete C source code packages, allowing for full customization by the developer.

### *DELTA Controls Stream Device Package*

The challenge of incorporating GPIB instruments into an EPICS control system was dramatically reduced through the use of the University of Dortmund Stream Device package [1]. The package provides a straightforward API for EPICS integration of devices that are inherently message based, or in general, use serial data streams such as is typical in GPIB instrumentation. While not dedicated to GPIB support, the package includes all of the necessary software hooks to easily integrate new GPIB board drivers. In addition, there is a good level of separation between the GPIB interface and the instruments that are connected to the bus. The authors feel that this package represents a significant advantage over other GPIB support packages for GPIB integration with EPICS. Of particular note is the late binding of instrument command and addressing information to the EPICS database, avoiding the need to recompile driver source code to modify the commands and addresses used to access instruments. Full support for asynchronous record processing is provided in the Stream Device package, a necessary feature for use with GPIB in the EPICS environment.

### *TRIUMF Glue Layer*

Integrating the two layers of software was relatively straightforward. In most cases, the API of each package has a corresponding function in the other package. A simple shim to translate data structures and parameter passing conventions is all that is required. Moreover, the similarity of APIs provided in the two NI driver packages allows for a relatively standard code base.

### **CONCLUSION**

PC/104 based IOCs can be coupled with GBIP interfaces to provide effective integration of GPIB interfaced instruments into an EPICS based control system. The impact on the rest of the control system is

minimized or eliminated, and the financial cost is significantly reduced over a VME hosted system. The use of well written and debugged driver and support software significantly reduces the effort and expense of creating stand-alone GPIB support systems.

### **REFERENCES**

- [1] <http://www.delta.uni-dortmund.de/controls/pub/doc/streamDevice/>
- [2] R. Nussbaumer, "Laser Stabilization Controls For The Isac Beam Polarizer" ICALEPCS 2001, San Jose, November, 2001.
- [3] G. Waters, "TRIUMF/ISAC EPICS IOCs Using a PC/104 Platform", These Proceedings