

Software Architecture of the Longitudinal Feedback System for PEP-II, ALS and DAΦNE *

R. Claus, J. Fox, I. Linscott, G. Oxoby, W Ross, L. Sapozhnikov, D. Teytelman
Stanford Linear Accelerator Center, Stanford, CA, 94309, USA

A. Drago, M. Serio
INFN - LNF, Frascati, Italy

Abstract

We describe the software architecture of the Longitudinal Feedback System being built for the PEP-II B-Factory at SLAC, the ALS light source at LBL and the DAΦNE phi factory at Frascati. This VME/VXI based system utilizes commercially available embedded CPU controller boards running the VxWorks real time operating system. The operator interface for PEP-II and ALS is based on the EPICS control system package. Embedded processors are used to load, monitor and diagnose various components of the system. The feedback function is implemented using digital filtering techniques on a DSP farm residing in the VME crates. The operator interface is written to allow the loading of applications, e.g., accelerator diagnostic functions, system hardware integrity functions, etc., without intervening controller reboots.

1. INTRODUCTION

The Longitudinal FeedBack (LFB) system [1] [2] is intended to be easily portable to various storage rings. In order to accomplish this, its control system software has been made capable of interfacing with the indigenous accelerator control system without too much additional

coding effort. The first guinea pig for the prototype system is the ALS at LBL. The Experimental Physics and Industrial Control System (EPICS) package [3], already in use there for beamline experiment control, was chosen for the operator interface of the first implementation of the device.

2. DATA FLOW

The block diagram of the LFB system hardware is shown in Figure 1. The VXI crate contains the fast front-end and back-end electronics. The VXI standard is being utilized because of its good electromagnetic shielding properties, cooling capacity and system power choices. In the initial incarnation, the control processor is a National Instruments VXIcpu-030 with an internal disk drive. This module also has a GPIB interface, which will be used to tune programmable delay lines that adjust the phase of the input and output signals of the system with respect to the beam timing. After the VXI processor has set up the hardware, it does very little other than periodically checking module status bits in order to gauge the health of the system.

The VME crates are fitted with VSB backplanes. The VME and VSB buses are used as the data and control paths, respectively. Because of the six slot limitation of the VSB standard, there are two 10 slot VME backplanes per VME crate. Five of the VSB slots can contain DSP boards. These boards were designed in-house and contain four 80 MHz AT&T 1610 Digital Signal Processors. Each DSP chip has an associated 16 KB dual-port memory through which it can exchange data with the control processor over the VSB bus. There is one control processor (Force CPU-40s in the initial instance of the system) per VME/VSB backplane pair. Its job is to set up the hardware, load the DSPs with code and retrieve data from the DSP when requested.

Bunch phase-error information derived from a Beam Position Monitor and the master oscillator is sampled at the bunch crossing frequency (500 MHz in the ALS case) with 8 bit resolution. Four samples are packaged together (packetized) along with addressing information from a software programmable look-up table by the Down Sampler Module. It then sends the packets out on four configurable Gigabit serial data links to the VME backplanes. The same data may be sent out over multiple links in order to implement, say, a data spy mode. Data Relayer Modules (sometimes called Interface Modules) receive the data and pass it onto the VME bus with 32 bit Read-Modify-Write operations. The Data Relayer Module sends the result of the read operation to the Hold Buffer Module in the VXI crate, also over Gigabit serial data links. The Hold Buffer

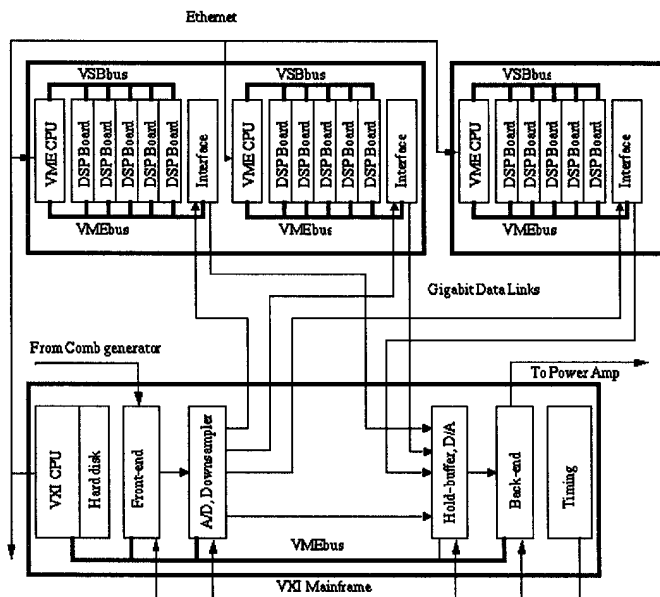


Fig. 1. System Hardware Architecture

*Work supported by Department of Energy contract DE-AC03-76SF00515

stores the phase correction values in memory locations corresponding to the bunches to which they apply and drives a fast DAC operating at the beam crossing frequency. The analog signal is ultimately sent to the Power Amplifier, which excites the Longitudinal Kicker.

3. ARCHITECTURE

In addition to the normal set of VxWorks and EPICS tasks, several tasks are spawned to implement the LFB control system. As shown in Figure 2, the LFB system consists of three layers of software: the interface to the local machine control system, an LFB system managing layer and an application layer.

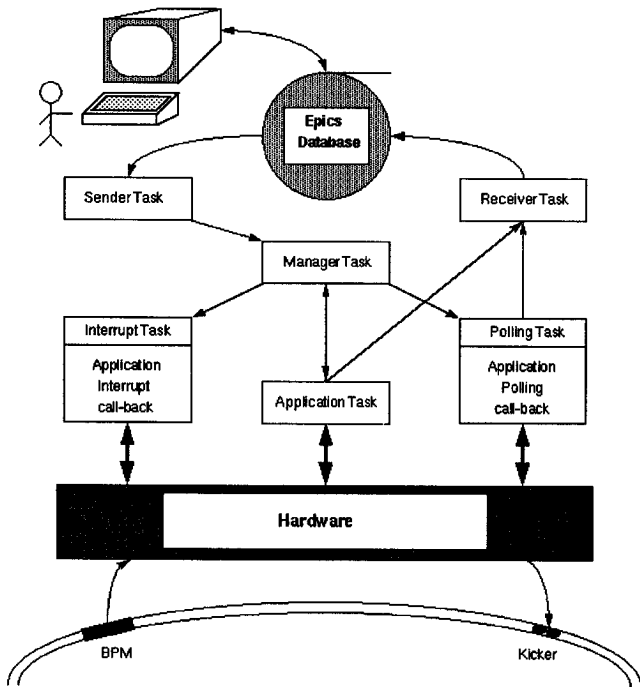


Fig. 2. System software architecture

The top-most level consists of two tasks that provide the interface between EPICS and the heart of the system, tSnd and tRcv. tSnd is responsible for taking operator input and translating it into a local message protocol. tRcv converts system information into messages that are sent to EPICS OPERator Interface (OPI) screens. These two tasks contain all the code (some 4000 fairly repetitive lines) that would need to be replaced in another machine control system environment.

The next layer of software contains the managing task tMgr and its offspring, the interrupt handling task, tInt, and the polling task, tPoll. tMgr receives and interprets messages coming from tSnd. Sometimes this means that it simply passes the message on to one of the subtasks. Legal messages are entered in a dictionary and consist of a message code, a corresponding routine and an argument for that routine. The "Load Application", "Abort Application" and "Quit" messages are always entered into the message dictionary, along with a set of general purpose hardware specific messages. Use of this dictionary concept allows for

great flexibility.

tInt and tPoll provide the services of handling the hardware interrupts and initiating periodic events, respectively. tInt installs an Interrupt Service Routine (ISR) appropriate for the hardware that it handles. The ISR packages the interrupt information into a message packet that is sent to task level code. If an application has registered an interrupt message handler with the interrupt task, then the packets are forwarded to it. This avoids application code writers having to write interrupt level routines, which are difficult to debug.

The tPoll task is often used for checking on the status of the hardware. An application may register a call-back routine with the polling task as well as set the wake-up period. The call-back can then check the hardware for error conditions, look for the existence of certain tasks, etc. The call-back generally sends status information to the tRcv task to inform the operator of the state of the system.

The application layer forms the third layer of software. Applications are loaded and started by the "Load Application" message, after the tMgr task has been established. Applications may add their own messages to the message dictionary. Only one application can run at a time. Applications define what the system does and thus generally contain different code for each of the control processors and the DSPs. Compile-time macro definitions (implemented through the *make* utility) choose the processor specific code when generating object files. Applications can, but do not necessarily have to have an associated DSP executable. Example applications are various hardware diagnostics, e.g., memory tests, bus transfer tests, etc., and digital filter codes, e.g. FIR, IIR, LQG, etc.

DSPs are booted with code loaded into their dual-port RAMs by their controlling processor. They hand-shake with their controlling processor through VSB interrupts and a communications block that they set up at initialization time in their external RAMs. Due to licensing agreements with AT&T, the DSP code assembler can not be bundled with the system like the VxWorks executables can. Thus, the filter coefficients are not part of the DSP code, but are instead loaded through the dual-port RAM using the hand-shake mechanism.

Encoded in three bits of the DSP addressing information of the data is a filter selection code. The DSP code can use this information to select different sets of coefficients in the case of, say, beam grow/damp accelerator experiments, enable/disable data recording, etc.

The filtering algorithms we have implemented in the DSPs typically take around 600 nanoSeconds of execution time and contain some 30 instructions. More time-consuming algorithms could be used if more DSPs were available, up to the limit that can be handled by the four available links on the Down Sampler and Hold Buffer modules. In the ALS case, there are two VMEbackplanes containing five DSP cards each for a total of 40 DSP chips. Together this corresponds to roughly 2000 MIPS.

All together, the control processor codes comprise some 20000 lines of c-code.

4. EPICS

EPICS [5] is a package with which one can construct an Operator Interface running on a UNIX platform to hardware controlled by code running on embedded computers running the VxWorks Real Time Operating System[4]. It contains many drivers to commercial VME and VXI devices. Communication between different computer platforms is achieved through a distributed database. Several Graphical User Interface (GUI) programs exist with which to build operator screens.

Due to the desire to maintain the portability of the Longitudinal Feedback System to other storage rings, it was decided to avoid becoming heavily EPICS dependent by coding EPICS device drivers for the SLAC-developed hardware. An interface to non-standard software can easily be developed by accessing database records. Some 200 records are used in the LFB system. A small subset of these records are used to cause the tSnd task to wake up and take action. These "events" cause other records to be read and written to the database.

The top-level OPI screen is shown in Figure 3. The triangles are "buttons" that cause another screen to be displayed. Each module in the system has an associated screen.

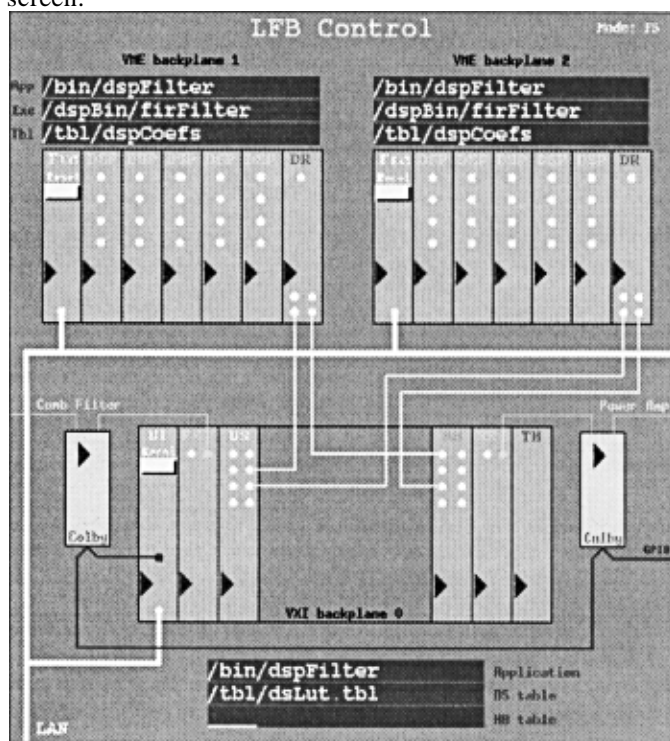


Fig. 3. An EPICS Operator screen

The system has two modes of operation: Full System and Stand-Alone. Stand-Alone mode allows one to use the control processors independently. This is useful when debugging software or hardware. In this case, the processors each have their own EPICS database and can not communicate with one another. This allows multiple users, insofar as tasks can be accomplished with a single processor.

5. CONCLUSION

Software for the control of the Longitudinal Feedback system has been developed. The operator interface of the initial system is based on the EPICS package. The EPICS specific layer is easily replaceable with another control system package so that the system remains portable to other sites.

In the coming months the Longitudinal Feedback system will be completed and installed at the ALS at LBL. By that time, some additional features will have been implemented on the OPI screens to allow for non-expert utilization of the system. Other features will also have been added to enable experts to perform accelerator experiments.

The completed system will constitute a significant advance in the tools available for accelerator control and beam diagnostics.

REFERENCES

- [1] D. Teytelman, et.al. "Operation and Performance of the PEP-II Prototype Longitudinal Damping System at the ALS" this conference
- [2] G. Oxoby, et.al., "Bunch-by-Bunch Longitudinal Feedback System for PEP-II" SLAC-PUB 6520, 1994
- [3] L.R.Dalesio, et.al. in: Proc. (ICALEPCS '93, Berlin, Germany, 1993) Nucl. Instr. and Meth. A 352 eds. W. Busse and M.C. Crowley Milling, (1994) pp.179-184.
- [4] VxWorks is produced by and a registered trademark of Wind River Systems, Inc.
- [5] EPICS is copyrighted by the Regents of the University of California, and the University of Chicago Board of Governors and is used with permission.