Contribution ID: **48**

Type: **invited**

Fermilab LLRF Software: Architecture and Development

Monday, 10 October 2005 16:00 (20 minutes)

Fermilab's Main Injector, Recycler and Tevatron synchrotrons Low Level RF systems perform a wide variety of tasks in support of accelerator goals, colliding beam HEP and neutrino production. In this paper, we will focus on the architecture and features of the LLRF software, the development practices we use and the lessons we have learned in the process. These specific features are presented within the framework of a standard control model.

The LLRF systems are called upon to perform a variety of different tasks including data acquisition, beam acceleration, machine to machine beam transfer, feedback control, data transfer, and user interface. To do this, we implement separate processes that run with different periods, ranging from a few microseconds to minutes. These systems must also implement a flexible toolset that allows an end-user to reconfigure the system, combine the tools in unforeseen and useful ways, and request new features. The systems must be highly reliable, and both fault detecting and fault tolerant. In addition, changes to any system must minimally affect beam operation of all systems.

In order to meet these requirements we use a variety of techniques. We employ a set of size C VXI modules built in-house from a standard base design, which includes, SHARC DSP, ALTERA FPGA, and VXI interface. The remainder of the module is customized
to perform specific functions, such as data acquisition, frequency control, or RF switching. The common DSP platform enables significant savings in software design and implementation by allowing us to develop, test and deploy a single solution to a common system problem across multiple VXI modules. In order to meet system specific goals, such as acceleration or beam transfer, cooperation between modules is
required. We implement an object-based methodology which provides a common interface to data and services provided by a VXI module. System level software employs these module objects by requesting services to fulfill system tasks and exposing module specific data to the accelerator control system. The system level software also implements the interface to support user requests for system reconfiguration. The interface allows the user to reconfigure a single parameter, such as feedback loop gains, or the user can specify a specific timed scenario of operations for the system to perform. Error logging and reporting are tightly integrated into the system software to enable debugging and fault diagnosis.

Primary author: Dr JOIREMAN, Paul (Fermilab)

Co-authors: BARNES, Barry (Fermilab); CHASE, Brian (Fermilab); MEISNER, Keith (Fermilab); VARGHESE, Philip (Fermilab)

Presenter: Dr JOIREMAN, Paul (Fermilab)

Session Classification: Talks Session 1