FULLY EMBEDDED EPICS-BASED CONTROL OF LOW LEVEL RF SYSTEM FOR SUPERKEKB

J. Odagiri, K. Akai, K. Furukawa, S. Michizono, T. Miura and T. T. Nakamura, High Energy Accelerator Research Organization (KEK), Tsukuba, Ibaraki, Japan
H. Deguchi, K. Hayashi, J. Mizuno and M. Ryoshi, Mitsubishi Electric TOKKI Systems Corp., Sanda, Hyogo, Japan

Abstract

Towards the SuperKEKB accelerator project, a new Low Level RF (LLRF) control subsystem has been designed and under implementation to upgrade existing one, which is based on Experimental Physics and Industrial Control System (EPICS). The new control subsystem comprises a micro-TCA, a Programmable Logic Controller (PLC) and an industrial PC. Each card plugged into the MicroTCA chassis as well as the PLC's CPU function as an embedded Input / Output Controller (IOC) of EPICS by running the IOC core program on the Linux Operating System. The industrial PC is assumed to run Extensible Display Manager (EDM) on Linux to serve as an Operator Interface (OPI) for both local and remote control. This paper describes the details of the design and the implementation under progress of the fully embedded EPICS-based low level RF control subsystem for the SuperKEKB accelerator.

INTRODUCTION

Since the end of 1998, KEKB, an asymmetric electron positron double ring collider for B-physics, has been successfully operated with increasing its luminosity to achieve the integrated value of 1000fb^{-1} in November 2009. In order to realize a luminosity two orders of magnitude higher, R&D efforts towards SuperKEKB are now under progress at KEK [1].

Besides the main components of the storage rings themselves, the control system, which is based on EPICS R3.13, also needs to be upgraded to catch up with the progress in the areas of CPUs, operating systems, I/O busses, FPGAs and so forth.

In this context, a new Low Level RF control subsystem has been designed with paying attention to adopt the latest technologies for better performance, easier maintenance, more flexibility and extensibility. Since the existing LLRF control subsystem is based on rather old-fashioned standards, such as VME, CAMAC, and NIM, some or all of them can be replaced with their successors. In addition, the version of EPICS [2] should also be updated from R3.13 to R3.14.

REQUIREMENTS

Three major requirements were imposed as the boundary conditions in designing the new LLRF control subsystem.

First and the most, the RF feedback control, which is the core function of the LLRF control subsystem, is required to shift from analogue-based to digital-based for higher precision and better reproducibility. It is preferable to adopt digital control for other peripheral functions for flexibility, extensibility and ease of maintenance.

Second, interoperability with the existing LLRF control subsystems in operation must be ensured because they need to be replaced with the new LLRF control subsystem one by one during the transition period. It is of importance to make the uppermost structure of the new EPICS runtime database on IOCs remain the same with current one so that the frontend I/O control becomes transparent to the Channel Access clients currently in service for KEKB operation.

The last requirement is to adopt generic and standard technologies so that the basic design of the LLRF control subsystem can be shared as much as possible by other future accelerator projects at KEK, such as Compact Energy Recovery Linac (cERL) [3], Superconducting Accelerator Test Facility (STF) [4].

CHOICE OF TECHNOLOGIES

The tasks of the new LLRF control subsystem can be divided into the following three categories:

- Fast RF feedback control and time-critical interlock.
- Slow sequence control, such as starting up and shutting down the RF power source, ramping up and down the RF power level, positioning of the piston tuners of the RF cavity, automatic conditioning of high power components, etc., and non-time-critical interlock.
- Operator interface for standalone operation at the local control room as well as remote control from the central control room.

FPGA and MicroTCA for Fast Control

Taking the requirements into account, Virtex-5 FPGA made by Xilinx has been chosen to implement the fast control. For the interconnection among the I/O cards based on the FPGA, we have chosen MicroTCA as the I/O bus for future system extensibility.

PLC for Slow Control

For the non-time-critical tasks, FA-M3 PLC supplied by Yokogawa has been adopted for its wide variety of I/O modules, reliability, compact size, ease of use and long product lifetime. The PLC system designed for the slow control consists of a main unit with CPU modules and subunits with only I/O modules to make it cost-effective.

Embedded IOCs for Fast and Slow Control

The most costly work in implementing software is development of the drivers for the communication between intelligent controllers. In order to avoid the burden, we have decided, for both fast and slow control, to make each of the controllers an IOC as possible as we can. It allows us to let the Channel Access library take care of the communication between the controllers, i.e., the embedded EPICS IOCs.

GUI-based Local Control

Because the new LLRF control system has adopted digital control to the fullest extent, most of the hardware knobs and meters on the local control panels need to be replaced with software ones even for the local control of the LLRF. For the platform to execute the GUI, an embedded PC was adopted. Extensible Display Manager (EDM) [5] is the most probable candidate for the GUI for its extensibility.

FAST CONTROL

In order to implement the fast logic for the digital RF feedback control, Virtex-5 made by Xilinx has been chosen. The FPGA is also used to handle time-critical interlock. Since Virtex-5 has an embedded PowerPC 440 processor, it can run the IOC core program on an OS. We have chosen WindRiver Linux 2.0 without any real-time extensions for the operating system since the IOC is not expected to do any time-critical works. Figure 1 shows the Virtex-5-based MicroTCA card (embedded IOC) for the RF digital feedback control.

Figure 1: Virtex-5-based MicroTCA card (embedded IOC) for the digital RF feedback control

Channel Access over MicroTCA Backplane

Traditional I/O-bus usage for IOCs has been to let the CPU board access non-intelligent I/O cards on the backplane with dedicated device/driver support modules. Another approach, however, is a viable option for MicroTCA, where Channel Access is available for the communication over Gigabit Ethernet on the backplane. By running an operating system on all of the Virtex-5-based intelligent I/O cards to make themselves embedded IOCs, as shown in Figure 2, existing Channel Access library as it is can be used to let the I/O cards (IOCs) communicate each other over Gigabit Ether without any special efforts. This approach considerably reduces the cost for the implementation of software on IOC by

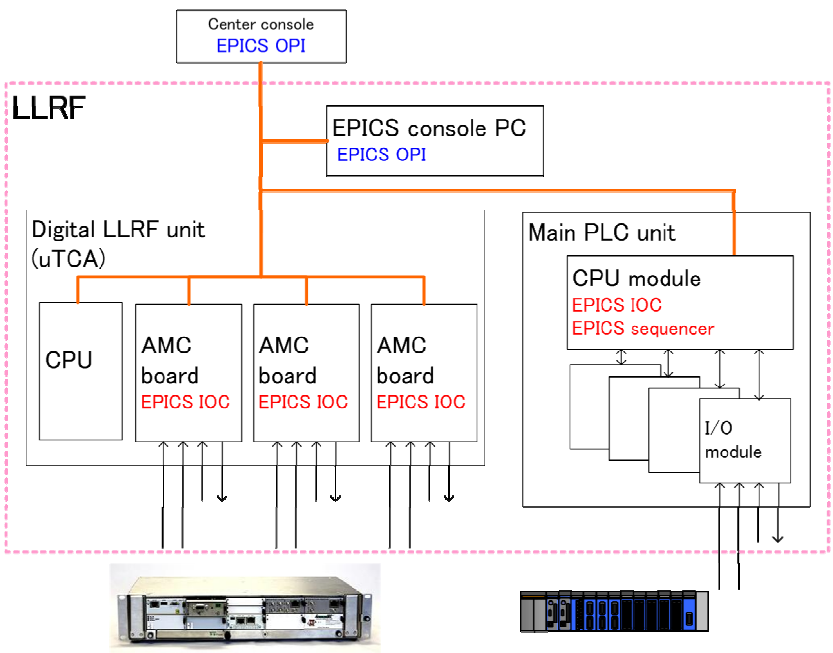

Figure 2: Configuration of the new LLRF control subsystem.

making most out of the existing well-established Channel Access library widely used and supported by the EPICS community.

In addition, using the same Channel Access protocol for both the system-wide control network and the local I/O bus makes it easier to change the configuration of the whole network structure dynamically by using routers or Channel Access gateways. The feature gives us more flexibility in designing and operating the control subsystem.

It is clear that the idea can be applied to any other network-based control systems other than EPICS because any network communication facilities can work as a kind of driver software for the I/O access on the backplane of a MicroTCA chassis.

Interfacing Embedded IOC to FPGA

On each I/O card, the IOC program needs to face the FPGA to read and write a memory area of the FPGA to set parameters and to monitor values. The simplest method to achieve the I/O access is to map the memory area into a range of memory space of the IOC program being executed as a user process of Linux. Once the mapping has been established, the EPICS device/driver support module can access the memory area through a pointer returned by the kernel level driver.

Prototyping of Embedded IOC on Xilinx ML507

In order to confirm the feasibility of the scheme, we have built WindRiver Linux 2.0 and the IOC core program (R3.14.9) for the Xilinx ML507 evaluation board, which has a Virtex-5 FPGA. The test result showed that Channel Access clients on a host computer can successfully access the memory area of the FPGA on the embedded IOC through the developed device/driver support and the kernel level driver for the memory mapping.

SLOW CONTROL

All the other non-time-critical control is handled by a PLC unit. It handles sequence control and non-time-critical interlock. The Embedded EPICS approach was chosen for the slow control as well by adopting an FA-M3 PLC with an F3RP61, a PLC's CPU running Linux [6]. The F3RP61 CPU serves an embedded IOC on the PLC's bus together with various I/O modules. While runtime database logic and the EPICS sequencer is used for sequence control on the F3RP61-based IOC, another ordinary PLC's CPU is used only for the non-time-critical interlock with a ladder program that suits simple interlock logic better. The two different types of CPUs work side by side on the same PLC bus. The F3RP61-based IOC polls the ordinary PLC's CPU through the PLC bus to monitor the interlock status and report it to Channel Access clients running on the embedded PC of the LLRF control subsystem and CPU servers in the central control room.

ACCESS SECURITY CONTROL

In the new LLRF control subsystem, the local control also relies on the embedded IOCs, which can be accessed by any Channel Access clients. It literally makes the distinction between "remote" and "local" the difference in location where the operator interacts with the GUIs. For this reason, the use of Access Security Group (ASG) equipped in IOCs is indispensable in order to prevent operators from making accidental interference in the operation of the new LLRF system.

HOLDING STATUS

Another consequence of the replacement of hardware knobs with software ones is how to hold the last state of the operation upon rebooting the embedded IOCs. For this purpose, a library supported by the EPICS community, autosave, can be used. It is more reliable to have the files for autosave [7] to save and restore the last state inside the local LLRF control subsystem. Since the F3RP61 CPU can have a file system on its 4M bytes of SRAM available for user, the SRAM can be used for the storage device.

SUMMARY

A fully embedded EPICS-based LLRF control subsystem has been designed and under implementation towards SuperKEKB. Intelligent Virtex-5-based I/O cards connected by MicroTCA backplane have been adopted for the fast control, i.e., digital RF feedback control and time-critical interlock. Each of the cards runs Linux and becomes an IOC being communicate each other through Channel Access over the Gigabit Ethernet on the backplane. For slower sequence control and non-time-critical interlock, an FA-M3 PLC has adopted. The PLC's CPU which runs Linux also functions as an embedded IOC for the slow sequence control as well as the interface form another ordinary PLC's CPU dedicated for the interlock to EPICS. Human interface for the local control becomes mostly GUI-based as a consequence of the adoption of the embedded IOCs for both of the fast and slow controls.

REFERENCES

[1] H. Koiso, "Design of SuperKEKB based on the Nano-beam Scheme", in these proceedings.
[2] http://www.aps.anl.gov/epics/
[3] T. Miura, et al., "Low Level RF System for cERL", in these proceedings.
[4] S. Michizono, et al., "Digtal LLRF System for STF S1 Global" in these proceedings.
[5] http://ics-web.sns.ornl.gov/edm/
[6] A. Uchiyama, et al., "Development of Embedded EPICS on F3RP61-2L" Proc. of International Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC08), Ljubljana, Slovenia, Oct. 20-23, 2008.
[7] http://www.aps.anl.gov/bcda/synApps/autosave/autosave.html