IMPLEMENTATION OF PIC-BASED EMBEDDED I/O CONTROLLER WITH EPICS FOR RILAC CONTROL SYSTEM

A. Uchiyama#, H. Yamauchi, SHI Accelerator Service Ltd. (SAS), Shinagawa, Tokyo, Japan
J. Odagiri, High Energy Accelerator Research Organization (KEK), Tsukuba, Ibaraki, Japan
M. Komiyama, M. Fujimaki, RIKEN Nishina Center, Wako, Saitama, Japan

Abstract

In the frequency-variable RIKEN heavy-ion linac (RILAC), the control system includes not only EPICS-based controllers, but also handmade hard-wired controllers. For the replacement of the hard-wired controllers on the simple operation, it is required to adopt a low-cost remote controller which has some simple I/O functions. For this purpose, the PIC network interface card (PICNIC) has adopted as a low-cost remote controller and its EPICS device support was developed by using asynDriver. As a result, we have confirmed the effectiveness of the PICNIC for simple control, and we adopted it to beam attenuators control, alarm system, and a part of vacuum system.

INTRODUCTION

The RILAC is a main linear accelerator with variable-frequency folded-coaxial RFQ linac (RFQ) as a pre-injector, and Charge-State Multiplier system (CSM) located in the downstream beam transport line of the RILAC works as a booster with upgrade its performance [1]. We adopted the EPICS-based controllers such as Network-Device Interface Module (N-DIM) of RIKEN's original device, CAMAC, VME, and GPIB for the control system of RILAC [2]. On the other hand, handmade hard-wired controllers which are not supported by EPICS are used in many different locations for RILAC control system. For example, the RF beam chopper controller using the on-off remote switches with hard-wired line consists of electronic components and electromagnetic relays on a substrate. However, in order to unify the system, it is required to replace the hard-wired controllers with ones which can interface with the EPICS-based system on an Ethernet interface. Since it was not required intelligent control logic, major controllers supported by EPICS, for example Programmable Logic Controllers (PLCs), were not suitable from the point of view of the cost performance.

We expected that we could solve the complication described above using low-cost PICNIC with an Ethernet interface. To do so, the EPICS device support software for the PICNIC was developed using asynDriver provided by EPICS collaboration.

#a-uchi@riken.jp

PIC NETWORK INTERFACE CARD (PICNIC) AS I/O BOARD

Born in Japan, PICNIC designed by M. Ochiai is an I/O board with built-in PIC16F877 microcomputer made by Microchip Technology Inc (see Fig. 1) [3]. The main characteristic of PICNIC is to implement the TCP/IP protocol stack and simplified web server with its firmware inside a small PIC16F877. Thereby, it is available to configure a system to use the network using web browser for PICNIC. Similarly, we can test PICNIC's I/O control with our web browser without using other client program when it is needed. From the point of view of maintenance, it is the very important function to have user friendly interface in the controller for configuration own system parameter. The other characteristics are as follows:

- Very low cost (about 80US$)*
- Compact embedded board (70mm * 100mm)
- Available for writing simple sequential logic inside PIC16F877 with assembler language
- Own expansion board with the electronic components is necessary to implement

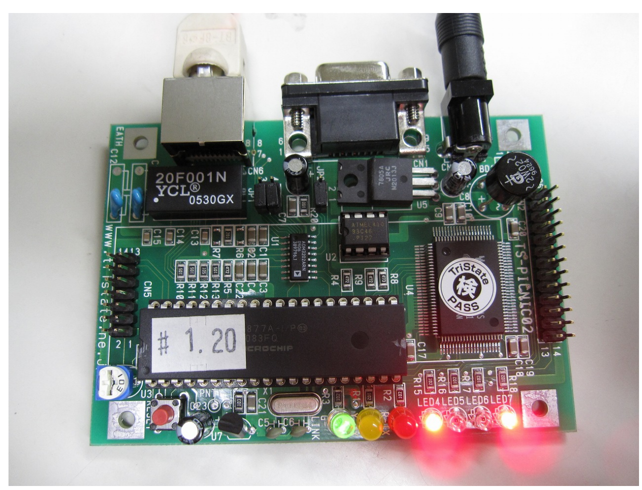

Figure 1: Photograph of PICNIC.

In Japan, the PICNIC was commercially released in 2001 and it became a famous controller, so that technical magazines and web pages introduced the examples of system development using PICNIC in detail [4]. In addition, there is a lot of flexibility for the system development, because the firmware, the connection

* 1 US$ = 89.6 yen in October, 2009

library (for GNU/Linux, Microsoft Windows), the source code and the schematic sheet are open to the public with GPL. For these reasons, PICNIC is an effective candidate for a small scale I/O control system to shorten the system development process. The hardware specifications about PICNIC version 2 are summarized in Table 1.

Table 1: PICNIC Specifications

Communication method	Ethernet 10Base-T
Analog I/O	Input 4ch, (and 1ch for temperature) Input range 0~5V Resolution 10bit
Digital I/O	"Output 4ch, Input 4ch" or "Output 6ch, Input 2ch" *selected
Other Interface	Serial (RS232C)
Protocol Stack	ARP, IP, TCP, UDP, HTTP, ICMP, DHCP

DEVICE SUPPORT FOR PICNIC

Development Method

First we have to develop the device support to implement PICNIC in EPICS-based control system. For controllers using Ethernet connections, the EPICS device support should be developed with asynchronous access, so that a sequence of actions till I/O completion takes more time compared with controllers using Compact PCI and VME. In this case, EPICS collaboration provides asynDriver as a framework for development of asynchronous device support [5]. By contrast, NetDev as a framework to support devices with Ethernet interface in EPICS-based control system is used at KEK, RIBF and J-Parc for PLCs, N-DIM and other intelligence controllers by the same reason [6]. In order to use a PICNIC library for TCP/IP connection as it is for the low-level driver of asynDriver, we selected asynDriver as a framework for the development of device support.

Development Environment

Using the latest version of EPICS base and asynDriver in February 2008, we developed the software on Linux. The detailed development environment is shown in Table 2.

Table 2: Version of Development Environment for the Device Support

Operating System	Linux, CentOS 5.1 i386
EPICS base	R3.14.9
asynDriver	R4-9
PICNIC Library	libpicnic-20051115.tar.gz†

† Download at official site for PICNIC

Deliverables from this work

The device support named devBiPicnic, devBoPicnic and devAiPicnic were developed to control DI/DO and AI channels, and drvAsynPicnic was developed as a common low-level driver. In the program behavior, drvAsynPicnic calls the function included in the connection library for access from EPICS IOC to PICNIC using UDP. The block diagram for the device support is shown in Fig. 2. Currently, this device support runs only on Linux.

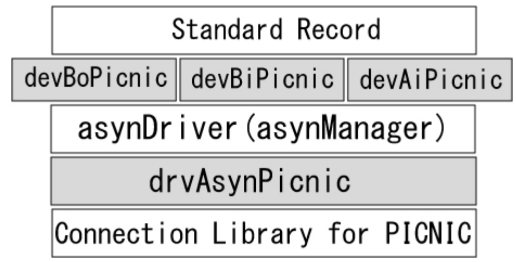

Figure 2: The block diagram for this work. The part of this developed software is shown by shaded region.

APPLICATIONS OF PICNIC

Facility for Applications

From the point of view that the ion beam has been supplied for experiments since 1981, RILAC is older facility than other accelerators in RIKEN [1]. In the facility, a part of aging controllers for accelerator operation was replaced with newer ones. Of course, we would like to replace aging controllers with the EPICS-based remote controllers completely in RILAC control system. However, the other legacy remote controllers are left in the control system till now, because of the fallowing various reasons. For example, we can implement hard-wired controller for remote control with ease in order to shorten the system process when it is needed. Because RILAC control room is away from accelerator room only by ~ 20 m and the almost of hard-wired controllers have simple function. Secondly, it is difficult to replace the aging controllers which are running with no trouble, even if the replacement of hard-wired controllers is useful in operation. Thirdly, it is not suitable for simple controlling to replace our hard-wired controller with intelligent and costly controllers from the point of view of the cost performance. To resolve these complications, we attempt to implement the application of PICNIC as EPICS-based controller with low-cost in RILAC control system.

Beam Attenuators Control

We succeeded to implement PICNIC for beam attenuators control with EPICS. A beam attenuator is installed device consisting of a tantalum mesh and rod to reduce the beam current in upstream beam line of RILAC. In the old beam attenuator's control system, accelerator operators in RILAC controlled ones by pushing the real on/off button in the control room. Since RILAC is injector for RI Beam Factory (RIBF) and RILAC control room is independent from RIBF control room, RILAC operators often controlled the beam attenuators with requirement from RIBF operators by phone. For effective operation, it was necessity to allow RIBF operator to control the beam attenuators in RILAC with remote from RIBF control room. The old controller was composed of hard-wired line from control room to the beam attenuators in accelerator room, on/off button, and 24V power supply. Since all of beam attenuators consist of three-way valves in driver system, to control beam attenuators remotely, it is enough function to control on/off switches which supply 24V power using electromagnetic relays with Channel Access (CA). Therefore, we built the own expansion board with electromagnetic relays following the procedure developed by Shinshu University (see Fig. 3) [4]. As a result, it is available to control the beam attenuates with CA from RIBF control room.

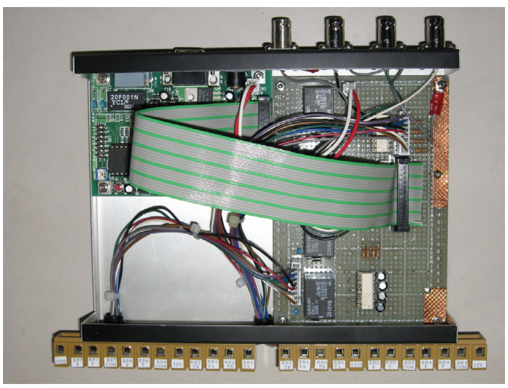

Figure 3: Photograph of the controller consisted of PICNIC and hand-made own expansion board.

Hardware Alarm System

We implemented the hardware alarm system using PICNIC, electronic buzzer, and the expansion board as well as beam attenuators control. The water leak in the ECRIS room is triggered by an alarm system using EPICS sequencer on embedded EPICS (see Fig. 4) [7]. In contrast, EPICS collaboration provide alarm handler as standard application like same purpose. Alarm handler is suitable system for alert of the error value for such as beam current, magnet power supply, and vacuum in accelerator operation. However, alarm handler cannot provide the service if the own application and X window system do not keep running on the PC client with speaker. For this reason, the real emergency alarm function should keep running at any time using hardware buzzer.

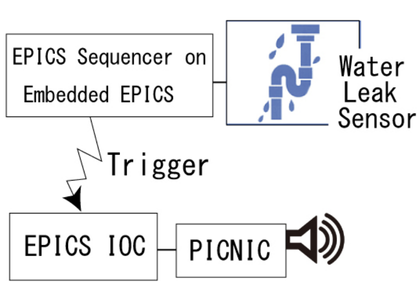

Figure 4: The system chart for the alarm.

Vacuum Gate Valve Control

For simple vacuum gate valve control, we plan to use the PICNIC as EPICS based-controller. Since the vacuum gate valve has three-way valves in the driver system as well as the beam attenuator, we can use the same expansion board for the controller directly. Additionally, some interlock function is provided by updating the firmware using assembler language inside PIC16F877 in this controller. Now this controller is in implementation test.

CONCLUSION

The purpose of our work was to implement PICNIC as low-cost EPICS-based controller in our system, and to resolve the complication of system by using PICNIC. In the implementation test, it was confirmed that PICNIC can be applied to various things in simple control, because it was available to control the electronic magnetic relays remotely with CA. We successfully implemented the applications of PICNIC for beam attenuator control, hardware alarm system in RILAC control system in April 2009. For the simple control described in the previous section, it is useful method to construct the system using the combination with low-cost PICNIC, EPICS and asynDriver.

ACKOWLEDGMENTS

The author would like to thank T. Ohki, T. Aihara, K. Oyamada and M. Tamura in RILAC team for supporting the program development time.

REFERENCES

[1] E. Ikezawa et al., "Present Status of RILAC", Proc. of the 5th Annual Meeting of Particle Accelerator Society of Japan and the 33rd Linear Accelerator Meeting in Japan, 2008, p265-267.

[2] M. Komiyama et al., "CURRENT STATUS OF THE CONTROL SYSTEM FOR THE RIKEN ACCELERATOR RESEARCH FACILITY", Proc. of ICALEPCS 2003, Gyeongju, Korea, 2003, p109-111.

[3] PICNIC official site, http://www.tristate.ne.jp/picnic-e.htm.

[4] Shinshu University, IT Technology Seminar (Masters), http://sugp.int-univ.com/Material/I18N/current/picnic-en/.

[5] M. R. Kraimer, "EPICS OPERATING-SYSTEM-INDEPENDENT DEVICE/DRIVER SUPPORT", Proc. of ICALEPCS 2003, Gyeongju, Korea, 2003, p205-207.

[6] J. Odagiri et al., "EPICS Device/Driver Support Modules for Network-based Intelligent Controllers", Proc. of ICALEPCS 2003, Gyeongju, Korea, 2003, p494-496.

[7] M. Komiyama et al., No. TUP084 in these Proceedings.