PC/104 EMBEDDED IOCS AT JEFFERSON LAB

Jianxun Yan, Allison Trent, Sue Witherspoon, Anthony Cuffe,
Thomas Jefferson National Accelerator Facility, Newport News, VA 23602, U.S.A.

Abstract

Jefferson Lab has developed embedded IOCs based on PC/104 single board computers (SBC) for low level control systems. The PC/104 IOCs run EPICS on top of the RTEMS operating system. Two types of control system configurations are used in different applications, PC/104 SBC with commercial PC/104 I/O cards and PC/104 SBC with custom designed FPGA-based boards. RTEMS was built with CEXP shell to run on the PC/104 SBC. CEXP shell provides the function of dynamic object loading, which is similar to the widely used VxWorks operating system. Standard software configurations were setup for PC/104 IOC application development to provide a familiar format for new projects as well as ease the conversion of applications from VME based IOCs to PC/104 IOCs. Many new projects at Jefferson Lab are going to employ PC/104 SBCs as IOCs and some applications have already been running them for accelerator operations. The PC/104 - RTEMS IOC provides a free open source Real-Time Operating System (RTOS), low cost/maintenance, easily installed/ configured, flexible, and reliable solution for accelerator control and 12GeV Upgrade projects.

INTRODUCTION

Currently, most of the accelerator control systems in Jefferson Lab were designed to operate with a VME- or VXI-based single board computer Input/Output controller (IOC). This VME-based IOC is a central control system with one IOC controlling multiple remote devices. When systems need to be expanded, that design usually requires long cable pulls and in some cases a new VME crate installation. These expansions require expensive parts and often did not accommodate available space constraints. Additionally, a license for the commercial real-time operating system on the VME IOC must be annually maintained or purchased for any new IOC.

Design options for a flexible and low cost embedded IOC were proposed. Many avenues were explored while searching for ideal embedded IOCs such as the implementation of an EPICS IOC on a Field Programming Gate Array (FPGA) [1] chip, and the Ethernet based embedded uCdimm Coldfire 5282 IOC [2]. These solutions couldn't meet all of our requirements: industrial standard supported by many manufacturers, low cost commercial solution, hassle-free software licensing, and easy integration into existing systems. PC/104 SBCs were developed as the new embedded IOCs. The PC/104 industry standard provides compact embedded-PC modules from many manufacturers. It has the features of low cost, standard ISA and PCI buses, and many expansion cards are available. And most importantly, it is supported by the Real Time Executive for Multiprocessor Systems (RTEMS) [3], an open source real-time operation system. EPICS has been ported to RTEMS, which made PC/104 IOCs seamlessly integrate into our existing EPICS system.

This paper discusses how the PC/104 SBC module was selected, what software toolkits are used, and what PC/104 IOC applications have been and are being employed.

PC/104 IOC HARDWARE

When choosing a PC/104 or PC/104-plus processor as the embedded IOC, there are a number of characteristics concerned. First, the processor should be selected by weighing the compromise between processing power and dissipated power. It is critical to choose the microprocessor with low dissipated power (a few watts), since in most applications the processor modules would run in front-end devices without cooling fans. Second, the module should have Remote Network Boot to load the operating system and application code over Ethernet. Preboot Execution Environment (PXE) is an example of this that uses the network interface controller (NIC) to boot computers. Usually, it is firmware in an intelligent ROM built into the network card. Finally, the choice of peripherals is also important: an Ethernet controller that is supported by the operating system, serial port, VGA controller, and keyboard connecter.

Many PC/104 and PC/104 plus processor modules from different companies were tested. MOPSlcdLX modules from Kontron were finally chosen as our embedded IOCs (Fig. 1). This module is a versatile, low-cost, low-power module that conforms to the industry standard 90mm x 96mm PC/104 form factor. It has the following features.

- AMD LX800 500MHz Fanless CPU
- System ROM with BIOS
- Up to 1 GB DDR-SDRAM support
- Interrupt controllers
- Onboard Video Graphics Array (VGA)
- Serial ports (COM1 and COM2)
- Real-time clock
- 10/100 Based-TX Ethernet with PXE LAN-Boot-ROM
- ISA bus and PCI bus

VGA and keyboard connectors were required to start the PC/104 SBC and setup system BIOS. The BIOS setup utility is used to enable onboard LAN PXE ROM so that the PC/104 module can be booted over an Ethernet connection. This allows the real-time operating system to be loaded from a remote DHCP server. After the initial setup, the VGA and keyboard are no longer necessary, and the COM1 serial port is applied as a serial console

once the real-time operation system is running. Serial console servers were employed to allow users to remote access the console and monitor and debug the embedded IOC.

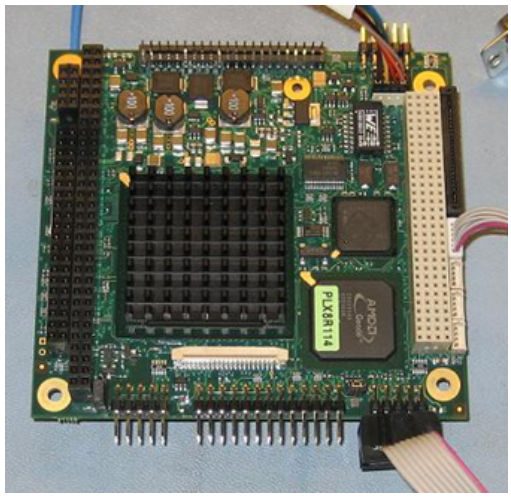

Figure 1: PC/104-plus processor module.

SOFTWARE

The first issue to resolve was how to load the RTEMS operating system on the PC/104 IOC. Jlab's control system standard requires IOCs to remotely boot the real-time operating system, dynamically load the objects, database, and EPICS, just as VxWorks does. In order to accomplish this, the following is required: PXE firmware, a Dynamic Host Configuration Protocol (DHCP) server, a Trivial File Transfer Protocol (TFTP) server, Bootloader, and RTEMS shell. GNU Grand Unified Bootloader (GRUB) is the bootstrap program used to load the RTEMS operating system. When the IOC is started, PXE firmware broadcasts a request packet on the network and then obtains an answer with an IP address from the DHCP server. After parsing the answer, the firmware asks the TFTP server for the file path of GRUB, downloads this initial bootstrap program and executes it. GRUB then loads the RTEMS firmware from the TFTP server and initializes it. Finally, RTEMS take control of all IOC resources. A generic "shell" tool called CEXP was applied as the RTEMS console interface. CEXP is a C-expression interpreter with integrated symbol table access and object file loading and linking. CEXP dynamically loads application object files, database files and EPICS. Since the original RTEMS toolkits did not include CEXP shell, the CEXP toolkit and RTEMS GeSys were obtained from SLAC [4]. The GeSys source code was modified and complied with RTEMS. Both RTEMS version 4.7 and 4.9 have been compiled with CEXP and run in the accelerator control system. EPICS version R3.14.9 and R3.14.10 were compiled with RTEMS 4.7 and RTEMS 4.9 respectively.

Some EPICS device drivers were developed, such as the functions for the PC/104 I/O address access and memory address. Device drivers were also written for some PC/104 expansion cards such as digital I/O, ADC/DAC, and serial communication cards. All these drivers can be reused by any PC/104 IOC. The configuration and development environment for the PC/104 IOC applications have been standardized to share the same procedure as the familiar VME IOC applications.

APPLICATIONS

Two types of control system configurations were used in different applications: PC/104 SBC with commercial PC/104 expansion cards and PC/104 SBC with custom designed FPGA-based boards. There are many different commercial PC/104 expansion cards on the market such as digital I/O, data acquisition, and communication modules. With the availability of multiple low-cost modules, one could easily configure front-end embedded devices. PC/104 expansion modules use I/O space ranging from address 0x000 to 0x3FF. Since not all the expansion modules could meet our application requirements, some FPGA based carrier boards were designed to communicate with the PC/104 processor. This custom design combined the advantages of both the FPGA and embedded processor. The ISA bus was chosen to communicate between the FPGA and PC/104 processor due to its simplicity. A PCI bus is also available on PC/104-plus SBC modules and could be used for data intensive applications. The ISA bus uses the system memory address space, which is larger than the IO address space. For instance, memory address space from 0xE0000 to 0xE7FFF is available on the Kontron MOPSlcdLX module. Firmware code on the FPGA was written as a bridge between FPGA registers and the PC/104 SBC to provide the address mapping and communication.

Figure 2: The new FEL beam viewer control crate.

Small PC/104 embedded IOCs have the flexibility to fit into different form-factors. There have been three different general configurations of PC/104 IOC installations. The first one was installed in a 3U card crate. The PC/104 processor mounted on a carrier card in the master slot and controlled all the other I/O cards in the crate via the backplane. The advantages of this structure were that one PC/104 IOC controlled a group of devices beyond the limits imposed from stacking PC/104 cards and is much cheaper/customizable than an off-the-shelf crate solutions. Figure 2 shows the FEL new Beam Viewer control crate, where a PC/104 IOC controlled 12 cards and each card controls 4 Beam Viewers. The second was to install the PC/104 in a chassis with PC/104

expansion modules or custom FPGA-based boards and a power supply. The inexpensive chassis could be easily moved and installed close to the equipment, eliminating the need for long cable runs to a central location. This also makes hot-swapping easier as each chassis stands alone opposed to a densely populated crate system with single points of failure. Control systems using this structure include the four LLRF chassis (Field Control, Stepper Motor Controller, Interlocks, and High Power Amplifier Control), RS-232 Serial Port Controller (Fig. 3), Distributed Data Acquisition, and so on. The third configuration put the PC/104 SBC on a custom designed carrier board in a closed box. This box is compact, has good radiation shielding and was designed to be installed and operated in radiation environments. The new BPM controller uses this configuration.

Some applications, such as the LLRF Stepper Motor Controller and FEL BeamViewer controller, have already been running PC/104 IOCs for accelerator operations. Many new projects are designed to employ PC/104 SBCs as IOCs with over 200 PC/104 SBC IOCS being installed over the next few years.

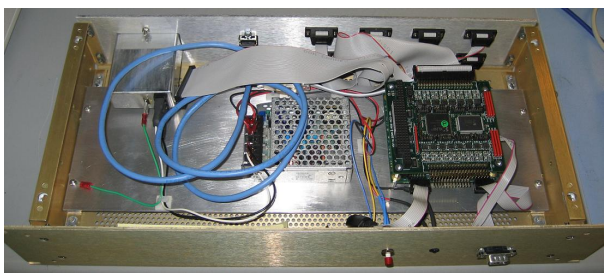

Figure 3: The chassis of the RS-232 serial ports control. One IO card controls 8 RS-232 ports. This chassis could be used for BCM temperature control and other applications that need serial communication.

CONCLUSIONS

PC/104 SBCs have been developed as a standard front-end device embedded IOC. The PC/104 processors card was chosen to meet our hardware and control system requirements. The GRUB, RTEMS, CEXP, and EPICS software were successfully compiled and built to run on PC/104 IOCs. Some PC/104 IOCs have been running accelerator controls, and provides a reliable and easily maintained embedded IOC solution. A number of 12 GeV Upgrade and other applications will use PC/104 IOC to control many of different devices in the Accelerator.

ACKNOWLEDGEMENTS

We would like to thank Till Straumann at SLAC for providing GeSys, CEXP source code, and help on compiling/building; and Eric Norum from Argonne National Lab for RTEMS installation instructions. Thanks to the Instrument and Control group and Software group in Jefferson Lab for their technical support.

REFERENCES

[1] J D. Durry, etc. “Implementation of an EPICS IOC on an Embedded Soft Core Processor Using Field Programming Gate Arrays”, 10th ICALEPCS Int. Conf. on Accelerator & Large Expt. Physics Control System, Geneva, 10-14 Oct. 2005, Po2.055-4.

[2] J. Yan, etc. “Ethernet Based Embedded IOC for FEL Control System”, 11th ICALEPCS Int. Conf. on Accelerator & Large Expt. Physics Control System,15-19 Oct. 2007, Konxville, Tennessee, USA.

[3] http://www.rtems.com/features.html.

[4] http://www.slac.stanford.edu/~strauman/rtems.