EPICS/RTEMS/MVME5500 FOR REAL-TIME CONTROLS AT NSLS

S. Kate Feng[1], D. Peter Siddons[1], Till Straumann[2], Mark Heron[3], and Steve Singleton[3]
[1]*National Synchrotron Light Source, Brookhaven National Laboratory, Upton, NY 11973, USA*
[2]*SSRL, Stanford Linear Accelerator Laboratory, Menlo Park, CA 94025, USA*
[3]*Diamond Light Source Ltd.., Diamond House, Chilton, Didcot, Oxon, OX11 0DE,UK*

ABSTRACT

At several National Synchrotron Light Source (NSLS) beamlines, open source control systems offer a modern solution for cost effectiveness and technical competence. The "Experimental Physics and Industrial Control System" (EPICS) [1] and the "Real-Time Operating System for Multiprocessor Systems" (RTEMS) [2][3] were chosen to constitute the core control system. A RTEMS Board Support Package (BSP) for the Motorola PowerPC MVME5500 board was written at NSLS [4][5]. RTEMS device drivers for a PMC2343 IEEE1394 (firewire) adapter and RTEMS libraries for IIDC 1394-based digital cameras were also written in house to be incorporated with EPICS video software [6] to provide inexpensive yet high-end image processing.

As of today, the EPICS/RTEMS/MVME5500 control system is implemented at several NSLS beamlines for real-time experiments. The RTEMS/MVME5500 BSP has been adapted at other facilities for accelerator and industrial real-time controls. The control system in use at these NSLS beamlines, and its performance, will be presented.

INTRODUCTION

There are over two thousands users of NSLS at BNL yearly. At several NSLS beamlines, EPICS as core control software and RTEMS as real-time Operating System (O.S.), both of them open source, were adapted to produce low-cost and robust VME control systems. Recently, Nobel Prize winning research was performed at one of NSLS beamlines, which utilize the EPICS/RTEMS/MVME2307 control systems. The Motorola PowerPC MVME2307 board was discontinued as of September 2003, which led us to a decision of developing a RTEMS BSP for the high-performance and cost-effective Motorola PowerPC MVME5500, which has a long-lived roadmap. The architecture of its system controller GT64260 is similar to that of GT64360, which offers off-the-shell embedded solutions on the VME/VXS platform (e.g. VPF1 or MVME6100 on 2eSST VME) for fast I/O on the custom hardware. Thus, some of the BSP software would become sharable for the planned embedded solutions.

The technical comparison listed in the next section has proven that the real-time performance of the EPICS/RTEMS/MVME5500 is comparable to that of vxWorks, a leading commercial real-time O.S.. The 1GHZ processor speed and high network throughputs of the MVME5500 offer users the advantage to integrate the RTEMS based real-time image processing software written at NSLS into the EPICS/RTEMS Input Output Controllers (IOCs), which further cut the system cost yielding compact, flexible and optimal real-time control systems.

PERFORMANCE MEASUREMENT

The following subsections present the network and latency performance comparison between RTEMS and vxWorks on the MVME5500.

Network performance

EPICS provide the “catime” application [1] for users to measure its client/server network performance (e.g. channel access). This was used to compare the network performance of RTEMS and vxWorks on the MVME5500 at NSLS on the same subnet via a PC equipped with an Intel Xeon 3.2 GHz processor and 1MB L3 cache running RedHat9.0 Linux installed with the 2.4.21 Kernel. The result is shown in Table 1 for the 10/100MHz port and Table 2 for the 10/100/1000 MHz port. As one can see the network performance of RTEMS and VxWorks are comparable to each other while tested under the same network setup. They showed similar performance for 1K channels, and for the 10K channels case, RTEMS had a slightly higher performance on both network ports.

It seems that the 1GHz Ethernet port did not show much higher performance than the 100MHz one for both RTEMS and vxWorks. It might be due to the advantage that the 100MHz Ethernet unit and SDRAM controller are integrated on the GT64260 system controller of the MVME5500, clocking the Direct Memory Access (DMA) operations via the CPU local bus at 133MHz. The 1GHz Ethernet controller is implemented on the 66 MHz PCI local bus, which was interfaced to the SDRAM via the GT64260 system controller. More tests need to be done to verify if a faster PC running Linux 2.6.x Kernel will improve its performance.

One could further enhance the performance of 1 GHz port on RTEMS by adding O.S. support for checksum offloading and TCP segmentation offload. The results listed in the tables are for scalars, not for arrays or images.

Table 1: catime performance of the 10/100 MHz port on the MVME5500 between EPICS3.14.6/RTEMS4.6.x and EPICS3.14.6/vxWorks5.5. Units are "Mega bits per second (Mbps)" and "Items per second (It/s)".

O.S.	RTEMS	vxWorks	RTEMS	Vxworks
# of channels	1,000 channels	1,000 channels	10,000 channels	10,000 channels
Search	5.8 Mbps	5.8 Mbps	27.1 Mbps	18.2 Mbps
Pend	413.6K It/s	470.3K It/s	445.2K It/s	447.6K It/s
async put	15.4 Mbps	13.0 Mbps	30.4 Mbps	22.0 Mbps
async get	19.1 Mbps	15.0 Mbps	51.4 Mbps	32.5 Mbps
sync get	1.4 Mbps	0.8 Mbps	0.4 Mbps	0.3 Mbps

Table 2: catime performance of the 1GHz port on the MVME5500 between EPICS3.14.6/RTEMS4.6.x and EPICS3.14.6/vxWorks5.5. Units are "Mega bits per second (Mbps)" and "Items per second (It/s)".

O.S.	RTEMS	vxWorks	RTEMS	Vxworks
# of channels	1,000 channels	1,000 channels	10,000 channels	10,000 channels
search	5.7 Mbps	4.8 Mbps	24.6 Mbps	21.1 Mbps
pend	453.1K It/s	450.0K It/s	453.5K It/s	450.9K It/s
async put	15.6 Mbps	15.6 Mbps	31.3 Mbps	30.7 Mbps
async get	19.5 Mbps	18.3 Mbps	54.0 Mbps	43.8 Mbps
sync get	1.1 Mbps	0.7 Mbps	0.4 Mbps	0.3 Mbps

Interrupt latency and context switching

An essential consideration of a hard real-time system is its "response time" (e.g. the time it takes for the system to react to an interrupt under worst case conditions). **Interrupt latency** is the time it takes from a device asserting an interrupt line until the system dispatching the corresponding Interrupt Service Routine (ISR). **Context switch delay** is the time it takes to schedule a task. It involves the scheduler determining which task to run, saving the current task context and restoring the new one.

The benchmark software [3][5][7] for the MVME5500 consists of an initialization routine, an interrupt handler (ISR) and a simple measurement procedure. It was dynamically loaded [8] and executed after the EPICS IOC was initialized.

The initialization code of the benchmark software sets up one of the timers on the MVME5500, connects the ISR to the respective interrupt and spawns a task executing the measurement procedure at the highest priority available on the system under test. Because the timer has a high resolution of 1/133,333,333 second per tick, is readable at real-time and reloads itself after each cycle, a precise measurement of latencies can be easily achieved.

For both RTEMS and vxWorks, 2,000,000 timer interrupts were generated at a rate of 4k Hz and the maximal and average latencies were recorded under both idle and loaded conditions. The loaded system consisted of 'catime' client constantly accessing the network interface of the IOC from a Linux workstation, while letting a low priority thread print characters to the serial console. Thus, the loaded system was subject to heavy interrupt and kernel activities involving scheduling, synchronization, networking and driver code sections among others.

The results are shown in Table 3. The idle systems exhibit comparable figures. Under the loaded systems, RTEMS is showed to be slightly more deterministic and steadier than vxWorks.

Table 3: Latency measurement results

EPICS/MVME5500	Interrupt latency in usec maximum (average)	Context Switching in usec maximum (average)
Idle System :		
RTEMS	5.04 (3.45)	6.80 (0.96)
VxWorks	6.10 (1.58)	9.65 (0.91)
Loaded System:		
RTEMS	8.17 (3.74)	17.48 (1.69)
VxWorks	13.90 (1.68)	20.80 (1.90)

In the RTEMS users mailing lists [9], there were interesting discussions from the industry, which indicated that RTEMS/MVME5500 responded faster than vxWorks/MVME5500 in a heavily loaded system.

IMAGE PROCESSING FOR PRACTICAL APPLICATIONS

At the beamline, images need to be handled more frequently as the experiments and their detectors become more sophisticated. Imaging x-ray detectors and visible light images all need to be integrated into the EPICS system, and it is becoming important that the latencies in the data acquisition be minimized. Applications such as beam profile measurement for automated beamline tuning, and automated sample alignment using video image processing are examples where tight coupling between the signal source and the EPICS IOC is desirable. Currently available packages offer PCs/workstations/SBC running windows, Linux, or vxWorks. To obtain the best performance for image processing, a deterministic real-time O.S. (e.g. RTEMS and vxWorks) is preferable to provide direct I/Os and priority scheduling so that the image processing can be optimally (e.g. thirty frames per second of data acquisition rate) and practically (e.g. at a specified real-time) integrated into the existing beamline control system.

However, the image software for the vxWorks is proprietary, expensive and incompatible with our RTEMS IOCs. Since the experience with EPICS/RTEMS/MVME5500 is successful we decided that an EPICS/RTEMS device driver should be written for the PMC2343 adapter, which provides an IEEE1394 Open Host Controller Interface (OHCI) on a PMC bus. In addition, the RTEMS libraries for IIDC 1394-based digital cameras were also written to provide the controls and image acquisitions of the digital cameras we chose (Fig. 1).

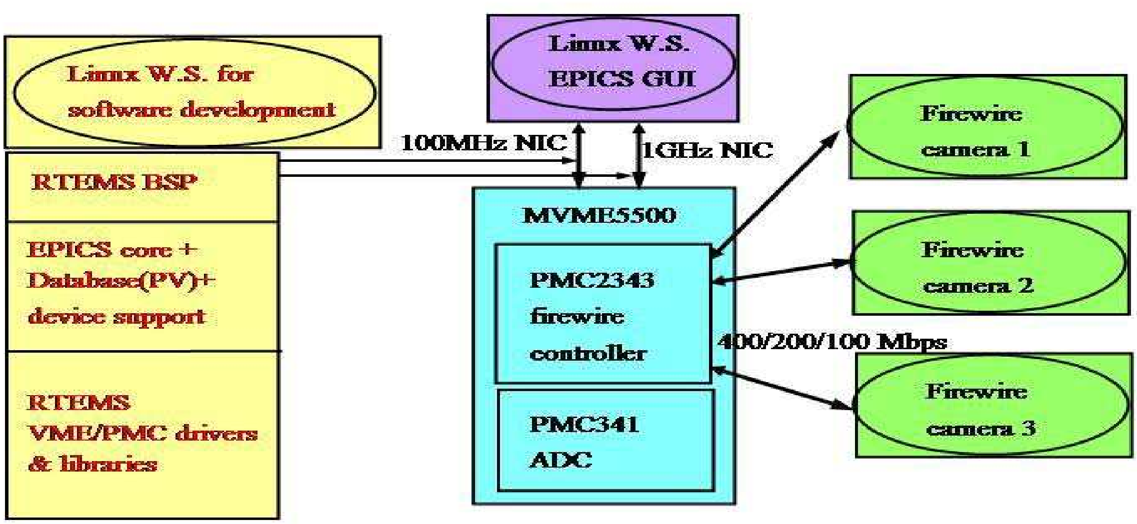

Fig. 1 Low-cost image processing for EPICS/RTEMS applications

As of today, the drivers were tested to control the camera for power on/off and for the readout of the camera format information. Although the image acquisition and performance measurement is not implemented yet, the software prototype is established and the project is close to completion.

OTHER VME DEVICES USED IN THE BEAMLINES

The ADC modules that were previously used for general purpose in NSLS beamlines were discontinued. Coincidentally, a need for a simultanenously sampling ADC arose for the purpose of reading out the four signals from an X-Y Beam Position Monitor (BPM). An EPICS/RTEMS device driver has been written for the PMC341 ADC module to suit both purposes. The 16 differential inputs of the PMC341 consist of eight 14-bit ADCs with simultaneous 8us conversion time for each of the two 8-channel banks.

While writing the RTEMS/MVME5500 BSP and drivers/libraries for the image processing of the 1394-based digital cameras are two major tasks, porting the existing drivers to the RTEMS port of EPICS/3.14.x Operating System Interface (OSI) is somewhat technically intriguing as well. Although a few RTEMS O.S. and MVME5500 related issues were encountered and resolved thereafter during the porting of vxWorks drivers, the EPICS 3.14.x OSI does facilitate these tasks. Existing EPICS device drivers for other VME devices such as OMS58 motor controllers, AVME9440 bit I/Os, VSC8/16 scalers, and IK320 encoders were ported and tested to work under RTEMS [5].

The carrot (not illusory at all) is to share the EPICS database and attractively designed Graphical User Interface (GUI) software, which is Motif Editor and Display Manage (MEDM) [1] based, and other client software such as StripTool for the beamline applications. So far, the learning curve for the MEDM based GUI software is as easy as it could be. The unified GUI further cuts the learning curve for users who run experiments among different facilities which supply the same GUI.

RTEMS/MVME5500 DEBUGGING

Most of the exception handlers of the RTEMS PowerPC BSPs, including that of the MVME5500, would dump the register contents and print stack traces information (Fig. 2) if the system crashed. Tracing back the stacks against the disassembled dump of the code allows one to debug the software simulating the real-time debugging. Simulation of real-time debugging is a highly desirable feature for a real-time O.S..

```
Next PC or Address of fault = 1F3AE2C, Mvme5500 Saved MSR = B032
Stack Trace:
IP: 0x01F3AE2C, LR: 0x01F3AE04
→ 0x01F3AB98→ 0x01F3A114→ 0x00007C90→ 0x0000C7D0→ 0x00006E94
→ 0x0000664C → 0x00006150→  x0000463C→0x0011FE90→0x0011FD9C
unrecoverable exception!!! task 0A010001 suspended
```

Fig. 2 An example of stack trace printed by the exception handler

There is a source-level symbolic debugger built for the RTEMS/MVME5500[8], which was derived from the GNU Debugger (GDB) [10] by adding enhancements such as supporting a RTEMS/CEXP(e.g. dynamic loader [8]) target. On the target, the debugging agent must be linked into the IOC system and started as a daemon. The GDB from a host computer will find the source code lines corresponding to the PC addresses on the stack and provide other debugging capabilities such as setting/resetting breakpoints, displaying/modifying data structures and so on.

ADVANTAGE AND DISADVANTAGE OF EPICS/RTEMS/MVME5500

Advantage

Both EPICS and RTEMS have professional level user mailing lists for open technical discussions based on a large group of international users. There are no license hassles for the EPICS and RTEMS software. They both provide lifetime access to anyone even if the provider drops support on a particular platform because access is available to the source code level.

Disadvantage

As an open source O.S., RTEMS is not as widely used as other ones such as Linux, which is not designed for real-time applications. However, the strength of RTEMS is that it guarantees a higher performance for a worst case in a specified system load, while the performance of Linux or RTLinux is less predictable and much slower under the same hardware platform [3].

CONCLUSION

Open source promotes open discussion and open collaboration, enhancing technical effectiveness, which contribute to our success to a robust, competent, and inexpensive EPICS/RTEMS/MVME5500 system for real-time controls. The RTEMS O.S. is comparable to the leading commercial real-time O.S. such as vxWorks. Besides, it is flexible, reliable and suitable for even high-end real-time applications. The performance of RTEMS is thriving upwards as more people adopt it and contribute to it. As of today, RTEMS has a reasonable variety of BSPs. To further expand its horizon, more men/women power is needed to popularize RTEMS.

Mention of any commercial products in this article does not imply endorsement of those products by NSLS.

ACKNOWLEDGEMENTS

The authors wish to extend their thanks to Zhijian Yin and Ivan So of NSLS and EPICS community for the smooth O.S. transition of the IOCs at NSLS beamlines from EPICS/vxWorks/MVME2307 to EPICS/RTEMS/MVME2307, which essentially facilitated the porting of RTEMS/MVME5500 to the existing EPICS software. This work performed under the Department of Energy contract DE-AC02-98CH10886.

REFERENCES

[1] http://www.aps.anl.gov/epics

[2] http://www.rtems.com

[3] Till Straumann, "Open Source Real-Time Operating Systems Overview", cs,os/0111035 (2001) WEBI001, http://www.slac.stanford.edu/econf/C011127/WEBI001.pdf

[4] S. Kate Feng, "RTEMS BSP for MVME5500: Performance and Demonstration", EPICS collaboration meeting, 2004, http://lansce.lanl.gov/EPICS/presentations/KateFeng_ RTEM5500.ppt

[5] http://www.nsls.bnl.gov/organization/UserScience/Detectors/Software, or navigated via NSLS home page http://www.nsls.bnl.gov in case of future change.

[6]Mark Heron,"Control Systems for the Diamond Light Source", http://www.nsls.bnl.gov/newsroom/events/seminars/abstract.asp?ID=143, and Steve Hunt, "Firewire Cameras for Epics", EPICS collaboration meeting, 2004, http://lansce.lanl.gov/EPICS/presentations/Steve HuntFirewire.ppt

[7] http://www.slac.stanford.edu/~strauman/rtoslat

[8] http://www.slac.stanford.edu/~strauman/rtems/software.html

[9] http://www.rtems.com/ml/rtems-users/2005/july/msg00139.html

[10]ftp://ftp.gnu.org