A Real-Time Data Acquisition System with UNIX and CAMAC ACC

Yoshito TANAKA [*], Masaya HASENO [†]and Masaharu NOMACHI [‡]

ABSTRACT

Software Packages for a real-time data acquisition system using CAMAC Auxiliary Crate Controller (ACC) and UNIX based computer have been developed. The software is able to be developed in C language without any commercial development tools, with support for KEK standard library and communication library between the UNIDAQ data acquisition system and the ACC. In this paper, an application with this software as well as aspects of the development environment and performance are discussed.

1. Introduction

Many portable data acquisition systems have been developed for specific application in many places. This development style always have a disadvantage to keep up with advances in computer technology in addition to port a different platform. As a solution to overcome this difficulty, a data acquisition system named UNIDAQ with standard technology of UNIX system has been developed [1,3,4].

Recently not only many drivers have been developed for this system [5] but also the performances of this system under various OS on various platforms have been measured [6,2]. To achieve equivalent access speed on a CAMAC crate to the speed using the J11 ACC system, we have developed the software packages for Kinetics ACC system keeping the compatibility with the UNIDAQ system.

2. Hardware Configuration

The simplest hardware configuration with ACC is shown in Fig. 1. The UNIX workstation is connected to a crate controller (Kinetics Model 3922) with an interface card via the Kinetics parallel bus (K-Bus). This system have been implemented on a FORCE VME board computer (2CE or 5CE) running under SUN OS, on a personal computer running under Linux, and on a DEC Station running under Ultrix. We used the Kinetics 2917 interface card for the FORCE system, the Kinetics 2927 ISA card for the PC system and TurboChannel card for the DEC Station, respectively.

*Associate Professor, Department of Electrical Engineering
†Graduate Student, Department of Fluid Mechanics
‡Associate Professor, RCNP, Osaka University, Mihogaoka, Ibaraki-shi, Osaka 567, Japan
Received November 30, 1996

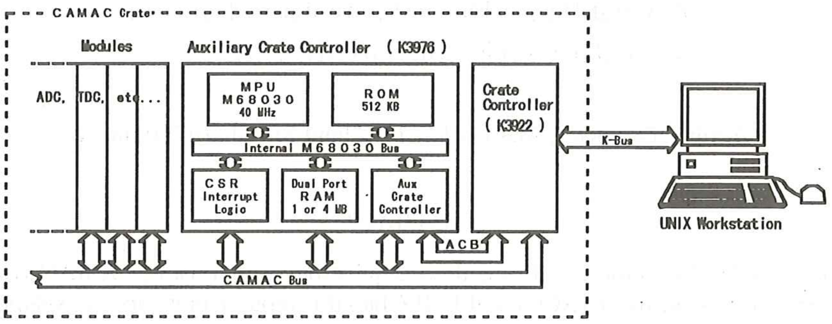

Fig. 1. Hardware configuration.

The ACC (Kinetics 3976) is a single span CAMAC module based on a Motorola 68EC030 40 MHz processor with 1 MByte or 4 MByte dual port RAM memory. The RAM memory is addressable from both the 68EC030 and the CAMAC Dataway and provides the capability of sharing data between the ACC and the crate controller. CAMAC modules can be controlled with both the 68EC030 and the workstation. The 3922 supports DMA block transfer and LAM interruption to the workstation.

3. Software

The data acquisition software packages,[a] ACC CAMLIB and ACC-UNIDAQ LIB, written in C language, have been ported to the UNIDAQ system. The functions of both the packages are compatible with KEK standard driver interface library (CAMLIB) and provide access the existing CAMAC drivers through the CAMLIB. The ACC CAMLIB is used as function library for coding the program of the ACC. The ACC-UNIDAQ LIB provides functions easy to control the ACC from the UNIDAQ and to communicate between UNIDAQ and ACC.

The layered structure of UNIDAQ software with the ACC is shown in Fig. 2. Since the architecture of the software was incorporate with UNIDAQ development philosophy, most of platform dependency is hidden in the kernel-level driver code with common interface library CAMLIB, so that the UNIDAQ program to control the ACC can be implemented on various platforms. While the ACC program can be developed to be independent of platform with almost same fashion as the CAMLIB.

3.1. Software Development Environment

All of ACC programs can be developed with GNU `gcc` cross compiler and `ld` linker

[a]The source files of these software packages can be obtained through WWW with URL http://www.elc.nias.ac.jp/~daq/software.

Application	UNIDAQ			
Library	CERN LIB	CAMLIB	ACC CAMLIB	ACC-UNIDAQ LIB
System	UNIX OS / X window System			
Kernel	Ultrix Driver	Linux Driver	Sun OS Driver	
Hardware	TurboChannel	K2927(ISA)	SFVME100 and etc.	

Fig. 2. Layered structure of the data acquisition system.

under any OS if the compiler is installed as 68030 target. C source code is compiled to relocatable object code with this cross compiler and linked to ACC CAMLIB with `ld` linker. The start addresses of executable code and data segment are changed with options of `ld`. The executable binary code is transferred to the ACC memory with a downloading tool which is developed for the GNU linker. The platform dependence of the downloader could not be found in our test.

Usually the downloader is called in the initialize routine of the UNIDAQ collector, after the developed program is transferred to the same platform as the UNIDAQ running if it is developed on the different platform. On the other hand, the UNIDAQ collector program must be developed with CAMLIB and ACC-UNIDAQ LIB on the same platform. The compilation works with either original or GNU gcc compiler.

3.2. Flow diagram of the system using UNIDAQ and ACC

The flow diagram of an application using UNIDAQ and ACC is shown in Fig. 3. After a ACC program is downloaded to the ACC, the start command of ACC application program is issued and the communication between UNIDAQ and ACC is established. In our application a command register is used for controlling the ACC from the UNIDAQ. An ACC and DAQ status registers are used for indicating the status of the communication and current data acquisition, respectively. The access control of data buffers are managed with data semaphores.

Once start command is accepted, the ACC gets a buffer and polls the LAM of CAMAC interrupt register. If the interruption is triggered, data is read into the buffer. When the buffer is full , the ACC interrupt to the UNIDAQ and data is transferred with DMA. Concurrently the ACC buffer is switched and the program is accepting the data into the other buffer. The data from ACC are transferred to UNIDAQ buffer manager NOVA and handled by UNIDAQ system.

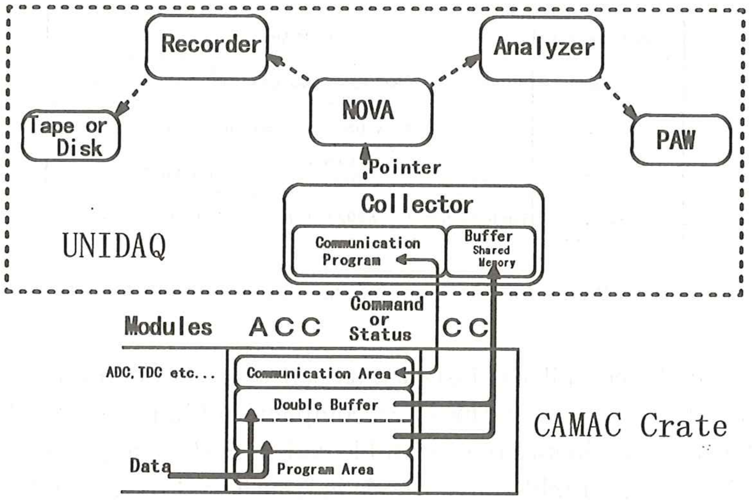

Fig. 3. Flow diagram of an aplication using UNIDAQ and ACC.

4. Performance

The performances of the UNIDAQ system working on the 5CE and the ACC has been examined and compared to the system without the ACC. The measurement is done with the same method as Y. Yasu et al. [6]. The ACC can read CAMAC data at 4.6 μs and write at 2.4 μs per word. The difference is attributed to the waiting time for setting data in CAMAC data register in read cycle. In the system without the ACC, the UNIDAQ can read and write CAMAC data around 100 μs in single action mode. The latency of interruption by generating CAMAC LAM is 10 μs or around 240 μs for the system with or without the ACC, respectively. The distribution of the interrupt latency with I/O consumer process [6] is, however, very large without the ACC because the SUN OS is not real-time OS. On the other hand the distribution of that with the ACC is negligible because the ACC handles all of CAMAC front end and is buffering the collected data on its memory. The block transfer speed from the ACC to the 5CE reaches to 650 kbyte/s on the average. Since the speed is higher than collecting speed of the ACC, this system is expected to make the smooth real-time operation for all application. The only dead time of the system, ignored the other dead time originated in the CAMAC cycle itself, is 40 μs for switching time of the ACC buffers.

The performances of the CAMAC single action for read and write, and those of interrupt latency without consumer process and its distribution in R.M.S. with I/O consumer process are shown in Table 1. The ACC has a very fast response

Table 1. Comparison with other system.

	READ(μs)	WRITE(μs)	INT Latency(μs)	INT width in R.M.S.(μs)
ACC	4.6	2.4	10.8	2.4
Linux	15	15	28.3	254.5
HP743rt [6]	22	20	70	not measured
5CE	106	106	244.2	1820.9
DEC [2]	160	150	480	not measured

Linux : GATEWAY2000 P5-150 (150MHz) Linux 2.0.0
HP743rt : HP743rt (64MHz) HP-RT 2.0.2
5CE : FORCE CPU-5CE μ SPARC (85MHz) SUN OS 4.1.3
DEC : DEC Station 5000/125 (125MHz) Ultrix 4.2

for all actions and gives the real-time feature to the non-real-time base UNIX data acquisition system in combination with the ACC.

5. Summary

We have developed the software packages for the data acquisition system with UNIX and CAMAC ACC. This software has been ported to the UNIDAQ system and added the real-time feature on non-real-time UNIX based data acquisition system with CAMAC bus. At present we have examined that the system works on DEC/Ultrix, VME/SUNOS and PC/Linux, and that the cross compiler has been succeeded to install on DEC/Ultrix, SGI/IRIX, SUN/SUNOS, HP/HPUX and PC/Linux.

6. Acknowledgements

We would like to thank E337 group members for giving the opportunity to develop the data acquisition system with ACC. We also thank KEK online group members for helpful advice and support of our development.

7. References

1. M. Nomachi et al., *Proceedings of the International Conference on Computing in High Energy Physics (CHEP) '94* **LBL-35822** (1994) 114.
2. Y. Yasu et al., *CHEP '94* **LBL-35822** (1994) 437.
3. A. Fry and M. Nomachi, *IEEE Trans. on Nucl. Sci.* **VOL.41 NO.1** (1994) 135.
4. Unidaq Doumentation Set, *Superconducting Super Collider Laboratory* **SDC-93-478** (1993).
5. Y. Takeuchi et al., *Nucl. Instrum. Meth.* **A328** (1993) 526.
6. Y. Yasu et al., *IEEE Trans. on Nucl. Sci.* **VOL.43 NO.1** (1996) 9.