A PC-Linux-Based Data Acquisition System for the STAR TOFp Detector *

LIU Zhi-Xu LIU Feng
(Institute of Particle Physics, Huazhong Normal University, Wuhan 430079, China)
ZHANG Bing-Yun
(Institute of High Energy Physics, The Chinese Academy of Sciences, Beijing 100039, China)

Abstract Commodity hardware running the open source operating system Linux is playing various important roles in the field of high energy physics. This paper describes the PC-Linux-based Data Acquisition System of STAR TOFp detector. It is based on the conventional solutions with front-end electronics made of NIM and CAMAC modules controlled by a PC running Linux. The system had been commissioned into the STAR DAQ system, and worked successfully in the second year of STAR physics runs.

Key words data acquisition system, TOFp, Linux

1 Introduction

The Time Of Flight patch (TOFp) is a sub-detector of the STAR (Solenoidal Tracker At Rhic) detector at the Relativistic Heavy Ion Collider (RHIC) at Brookhaven National Laboratory, USA. The goal of this detector is to provide information that greatly extends the hadronic PID capabilities of the experiment.

TOFp DAQ interface includes specific electronic equipment, such as a personal computer, commercial modules for the interface to the TOFp detector information and other STAR subsystems (Trigger and DAQ). The goal of the DAQ system is to control the hardware and read out the data, reformat and send data properly to the STAR DAQ system, control some I/O register modules to properly communicate with the TOFp Local Trigger and STAR Trigger, and visualize the raw data. Thus, the DAQ system is very important to the TOFp project.

TOFp detector consists of total 47 Photo Multiplier Tubes (PMTs). Each tube have two signals to be digitized: one for analog signal, the other for time signal. The analog signal is digitized by a 10-bit CAMAC ADC module (LeCroy 2249A), and the time signal by an 11-bit CAMAC TDC module (LeCroy 2228A). There is a 60Hz clock[1] input to the ADC module. A32-channel scanning 16-bit A/D converter is used as temperature and threshold voltage monitor. Thus, the total data volume is very small. Linux, the free operating system, and an Intel X86 personal computer are chosen as the developmental platform for the DAQ system. Although Linux is not a hard Real Time system[2], it is suitable for our small data volume.

2 Architecture of the DAQ system

The TOFp DAQ system consists of both hardware and software. The hardware system consists of a CAMAC crate and an Intel X86 PC. The software is developed on the Linux operating system. C is the main programming language. Fig.1 is the schematic diagram of the TOFp DAQ system.

2.1 Hardware

The digitization of the signal from front-end electronics is done in a CAMAC crate. There are 4 LeCroy 2249A ADCs and 6 LeCroy 2228A TDCs in the crate. A Kinetic System 3922 parallel bus crate controller in the CAMAC crate communicates with a Kinetic System 2915 PCI Inter-

Received 26 June 2002
* Supported by NSFC (10275027, 19975021)
253—256

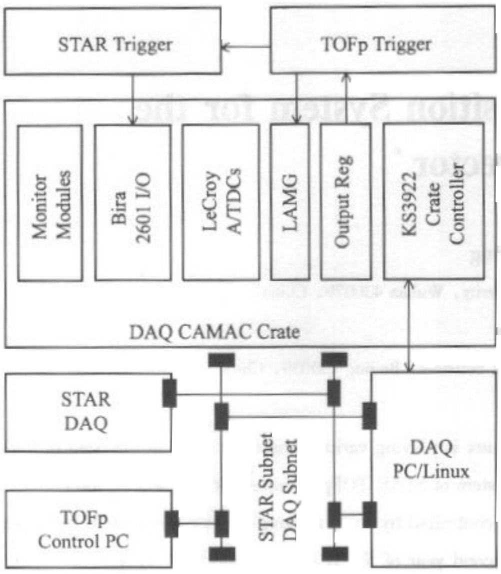

Fig.1. TOFp DAQ schematic diagram.

face in the PC's PCI bus.

The TOFp DAQ system needs to communicate with both Local trigger and STAR trigger, so some I/O register is necessary. A Bira 2601 24bits I/O register is chosen to communicate with STAR trigger, a Jorway 41 output register and a self-made LAMG module with Local trigger. LAMG is a single width CAMAC module which can accept a trigger signal and generate a LAM signal to CAMAC bus after a preset delay.

A system monitor is also very important for the normal system running. Thus, A crate verification module in the CAMAC crate is used for checking the CAMAC operations; a DTM-299 display monitor for program debugging; a Kinetic System 3516 32-channel scan A/D converter for temperature and threshold voltage monitoring.

The data readout and transfer is done by a PC with a KS2915 PCI card and two NIC cards. The function of KS2915 is to communicate with CAMAC crate and read out the data. One of the NIC cards is used for communicating with STAR DAQ in DAQ's private subnet, another for operator's control in STAR subnet.

2.2 Software

The software is an important and complicated part of the TOFp DAQ system. It consists of both the DAQ code and a local monitor system.

First, a device driver is needed for the KS2915 PCI card. The vendor does not provide a driver for Linux, but a driver for Linux 2.0 has been developed by KEK[3] and JNAL. So upgrading it to 2.2 kernel will satisfy our requirement. Based on some documents about the change of Linux device driver interface between 2.0 and 2.2[4, 1)], this has been done successfully. Also, the BUS Master (DMA) function of the card in the 2.0 driver does not work at all because of the kernel memory used are not allocated properly. After fixing this bug, the speed of reading the ADC/TDC data increases significantly.

Now turn to the DAQ code. Fig.2 is the block diagram. There are 5 programs running on the DAQ PC. To simplify the operation procedure, they are spawned by one program named tofp Mother. The functions of the five programs are run control, get-data, send-data, STAR DAQ monitor and local DAQ monitor.

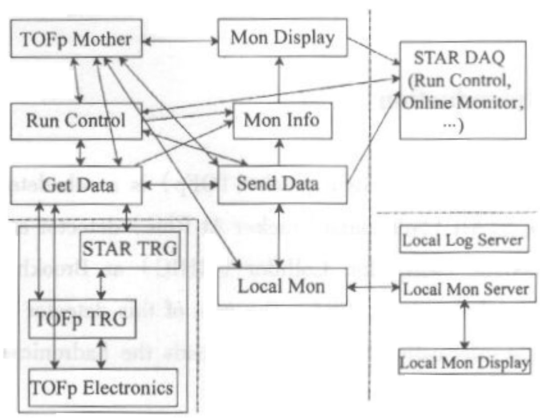

Fig.2. TOFp DAQ block diagram.

It is important that DAQ can store the event data in the local memory until further trigger information arrived. The DAQ PC has a 128M memory. By edit /etc/lilo.conf, add a line: append = "mem = 120M", the operating system can only see the 120M memory, so about 8M RAM (the actual size depends on the mother board) can be used for event buffer. The program can access the buffer by map system call on device /dev/mem. The semaphores are used to control the operation of the event buffer. They are created by tofp Mother program, so all the programs can access them.

Two shared memory segments are also created by tofp Mother. The system running status information are

1) Richard Gooch, Kernel API Changes from 2.0 to 2.2, http://www.atnf.csiro.au/people/rgooch/linx/dosc/porting-to-2.2.html

stored here.

The run control program communicates with STAR DAQ, gets command and notifies other program, then returns the execution status to STAR DAQ. It will update TOFp system status information too. The get-data program reads out the data and saves them in a shared buffer, so the send-data program can send the data to STAR DAQ. The get-data also communicates with both STAR trigger and TOFp Local trigger. When the event buffer is full, the get-data program will keep TOFp system busy, so STAR trigger will not send further trigger information to TOFp. Each time an event arrived or sent out, get/send-data program will update the information stored in the shared memory segments. The Round Robin scheduling algorithm[2] is applied to the get- and send-data program, so they have higher priority than other programs and can meet some time deadlines.

The STAR monitor program just sends some runtime status information of the TOFp DAQ system to STAR DAQ runtime system. The local monitor program sends a fraction of the raw data to the local monitor system which will display some histograms of the raw data, so one can get a quick display of the running status of TOFp detector.

The local monitor system is running on another operating PC. A local monitor server program running as a daemon receives the data from the DAQ system, and saves them as histograms to a shared mapped root file. Another program called as local monitor display can display the histograms. All these programs are written with ROOT[5] library.

3 Results of the DAQ system

During the second physics running of RHIC, TOFp is one of the new detectors in the STAR system. This running lasted for about 8 months. In the first two months the commissioning of both TOFp detector and DAQ system is very successful. As for the DAQ system, the test result shows DAQ can process up to 280 events per second. Due to the limit of other large data volume detector (TPC), the maximum speed, with which STAR can process, is 100Hz. The TOFp DAQ system is far above this limit. Also, the local monitor program helps to monitor and resolve many problems of the TOFp detector, such as the ADC failure because of the large ground shift in the TOFp tray. By the end of run in late January, 2002, approximately 4 million AuAu collision events and 22 million pp collision events were collected by the TOFp system.

4 Discussion

The running result shows that the TOFp DAQ system can handle the requirements of the TOFp system[6]. Linux is suitable for the small data volume DAQ system. For large data volume, a cluster can be considered.

The authors would like to thank W. J. Llope, Frank Geurts from Rice University USA and STAR DAQ group for their support and valuable discussions.

References

1 Llope W J, Adams N, Kainz K K. Nucl. Inst. and Meth., 2000, **A443**:451—463

2 Michael Barabanov. A Linux-based Real-Time Operating System, Master Thesis, New Mexico Institute of Mining and Technology, 1997

3 Tanaka Y et al. Nucl. Inst. and Meth., 1999, **A425**:323—331

4 Alan Cox. Porting Linux 2.0 Drivers To Linux 2.2: Changes and New Features, Linux Magazine, Spring, 1999

5 Rene Brun, Fons Radenakers. Nucl. Inst. and Meth., 1997, **A389**: 81—86

6 STAR TOFp Group. TOFp Functional Requirements

基于PC-Linux的STAR TOFp探测器的数据获取系统*

刘志旭 刘峰
(华中师范大学粒子物理研究所 武汉 430079)
张炳云
(中国科学院高能物理研究所 北京 100039)

摘要 廉价的运行开放源代码操作系统Linux的硬件在高能物理领域中正扮演着各种各样重要的角色. 本文描述了为STAR TOFp探测器设计的基于PC-Linux的数据获取系统. 系统采用了常规的解决方案:由一台运行Linux的PC来控制由NIM和CAMAC插件构成的前端电子学处理部分. 系统已经被成功的集成进STAR数据获取系统,并且在STAR的第二年的运行中工作正常.

关键词 数据获取系统 TOFp Linux

2002-06-26 收稿
* 国家自然科学基金(10275027,19975021)资助