DAQ with GPIB under LINUX

by Arif Mailov and Gökhan Ünel
(arif@caju.phys.boun.edu.tr
gokhan@caju.phys.boun.edu.tr)

on behalf of
BU-HEP Research Group[1]
Boğaziçi University, Physics Department,
Istanbul, Turkey

Abstract

After explaining some basics of HEP and HEP Instrumentation, we report about our experience using CAMAC crate with GPIB under Linux. We propose Linux-GPIB, as an alternative to expensive Workstation-Crate Controller approach.

The observable matter that surrounds us, consists of certain point-like elementary particles. The branch of physics which deals with the properties of these particles is called HEP (High Energy Physics). Creation and detection of these point-like particles in the laboratory require high energies thus short time intervals and expensive equipment which are obtainable only with a huge budget.
The equipment used in HEP Data Acquisition (DAQ), basically consists of crates and the cards plugged into these crates. The properties and operations of these crates (NIM, CAMAC, VME, etc.) [1] are well defined by International Committees.
The crate that we are using is a CAMAC crate by LeCroy, which has slots for 24 modules. The general operation principle is to send commands to the *Crate Controller*, which in turn communicates with its internal modules and gives a digital output as an answer. This chain is called a *CAMAC Cycle* . To access the crate controller, we use a NI-GPIB (General Purpose Interface Board) module that allows a maximum theoretical speed of 1 MBytes/s for 'read' and 'write', which is low but acceptable.

The modules which are used to collect data, are basically Analog-to-Digital Converters (ADC) and Time to Digital Converters (TDC) designed specially for HEP. This means that they are very precise (and expensive) devices. For example the time resolution of the TDC that we are using at the moment is 50 picoseconds in a range of 100 nanoseconds and the ADC is sensitive to charges as small as 0.25 picocoulombs.

After spending a lot of money for the crate and the DAQ modules, the only possibility to economize is to use a relatively cheap computer to control these equipment. Using a PC under DOS for device controlling is a common solution, but we consider it impossible because of the many limitations of DOS.

---

[1] http://caju.phys.boun.edu.tr/goups/highexp/highexp.html

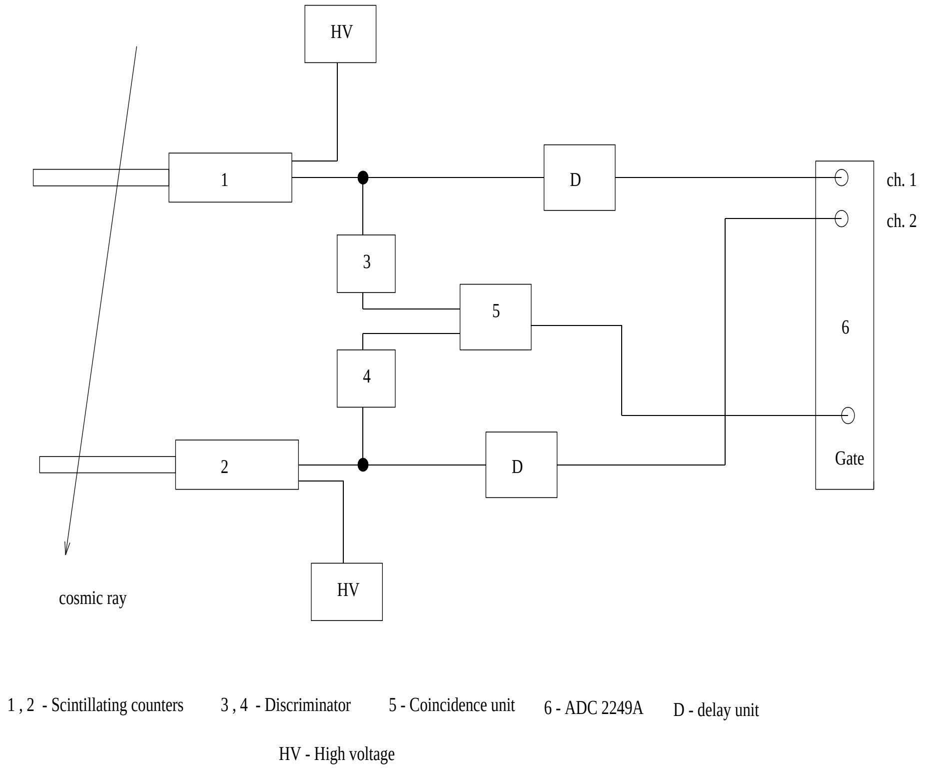

Figure 1: Setup for detection of cosmic rays

At this point the operating system used on the computer becomes important. As a matter of fact, PC hardware manufacturers have in their mind that DOS is the only environment for PC's. DOS/Windows oriented systems are widely supported and popular. But they are single-user and single-tasking systems. However, we wanted to access our system in the lab and to be able to control it from our main research laboratory, CERN (Conseil Européen de Recherce Nucléaire in Geneva). At that time, Linux Lab Project [2] was already mature and it was basically trivial to use our NI-488.2 card under Linux. To test each module, we have written some generic C programs using Linux-GPIB libraries. (We hereby want to express our thanks to Claus Schroeter, for developing these libraries.) We have seen that the speed of data acquisition with same algorithm implemented under DOS and Linux, differed by a factor of two. In addition to this speed increase, the possibility to work remotely, has made Linux [3] a definite choice. The availability of the HEP software under Linux (CERNLib basically [4]) has answered to the analysis requirements.

The next step of this project is to unite the programs written for different data acquisition units. Here, depending on the computer system used, two alternatives can be considered. In a multi CPU system, letting the single programs run and collect data individually and perform data collection with a shared memory system seems to be the best way to evolve. This approach, permits the development of the DAQ units separately, saves the rest of the system if one module crashes.

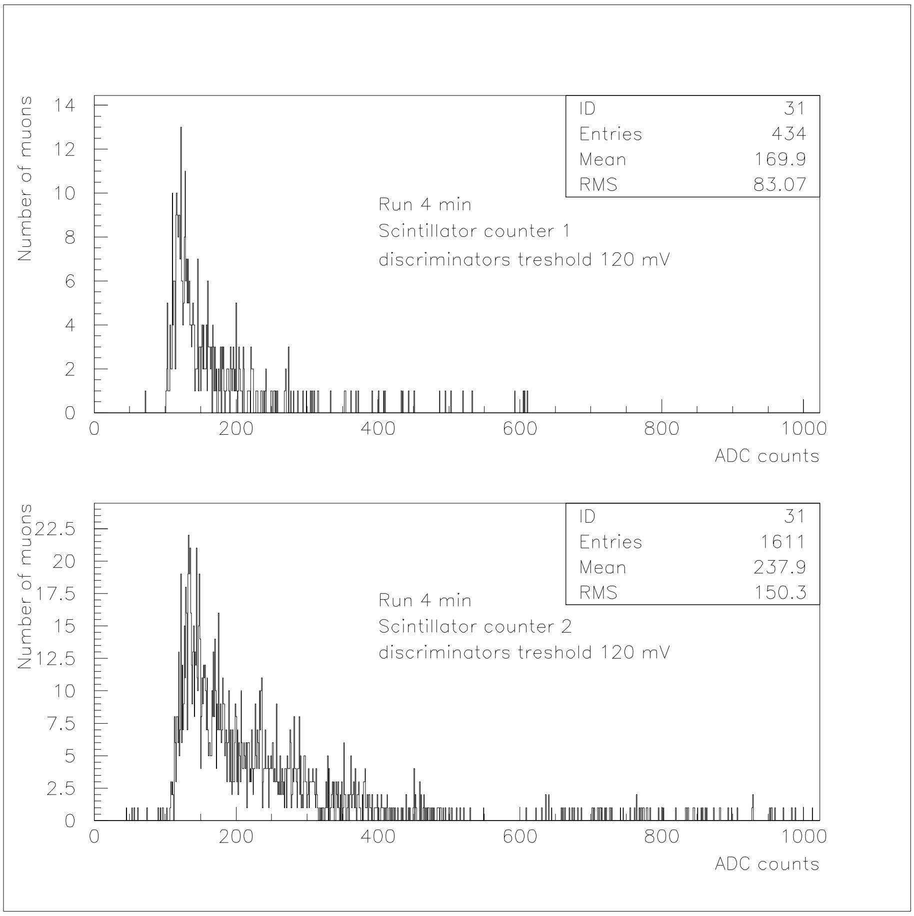

Figure 2: Pulse-height distribution of cosmic rays detected with scintillating counters

It also reduces the response time of the total system since one module does not have to wait until the end of the acquisition cycle of the previous one. For single CPU systems, we consider monolithic code as a solution. This monolithic code would be recreated for each application depending on the needs. As Graphical User Interface (GUI), since the speed is not so important, Tcl-Tk [5] based solutions are chosen and it is in preparation.

The first test application of our GPIB-CAMAC system was the readout of an input signal generated by HP 8001-A pulse generator. The readout was done with the LeCroy ADC 2249A which has a gate width of 70 nsec. The gate signal was obtained from the input signal via a discriminator. The input signal was splitted into two, one was fed into the discriminator and the other one which was read from the ADC, into a delay unit of 25 nsec to compensate the time spend in the discriminator unit.

The same analog signal was digitized 1000 times and the histogramming was done during online DAQ process. For the histogramming facilities, CERNLib program package, especially PAW [4], was used.

This simple application permitted us to evaluate the resolution of the ADC card, through the pulse-height distribution graph. Accepting our signal generator as an ideal one, we used the overflow region of the ADC to calibrate it.

Another application of the GPIB-CAMAC system, was the measurement of the

pulse-height distribution of the cosmic rays using a muon telescope(fig.1).
The signal from the cosmic particles was measured using two scintillating counters [6] in coincidence. The signals from two scintillating counters were digitized and plotted separately, if they were detected at the same time. These plots (fig.2) permitted us to estimate the ionizing energy of these particles.

In conclusion, we can say that Linux as a working environment for HEP applications, represents a robust, efficient and therefore promising alternative to other operating systems. The POSIX 4 conformance [7] of Linux, planned for future releases, will help to have it as a more common choice. The Linux Lab Project is an important step for this evolution.

References

[1] CAMAC standard , NIM Committee Report TID-25877, 1973
Wade D.Peterson, The VMEbus Handbook, VFEA International Trade Assotiation, 1991.

[2] For further information see: http://www.fu-berlin.de/~clausi

[3] For further information see: http://www.linux.org

[4] For further information see: http://wwwcn.cern.ch/pl/index.html

[5] John K.Ousterhout, Tcl and the Tk Toolkit, Addison Wesley,1994.

[6] For further information see:
http://www.phys.boun.edu.tr/groups/highexp/local/fiber/fiber.html

[7] Bill O.Gallmeister, POSIX.4, O'Reilly and Associate, 1995 ;
For further information see: http://luz.cs.nmt.edu/~rtlinux/