A PC/Linux-based Data Acquisition System

E. Leonardi[1]
Università degli Studi di Roma "La Sapienza" and INFN – Sez. di Roma

G. Organtini[2]
Università degli Studi Roma TRE and INFN – Sez. di Roma

ABSTRACT

We report on our experience on the realization of a data acquisition system based on a PC running the Linux operating system. The system uses a custom device driver to connect the PC to a CAMAC crate controller. Data acquisition software is entirely based on the CERN standard libraries and other freely available packages.

1 Experimental setup

The aim of our experiment, part of the R&D programme for the electromagnetic calorimeter of the CMS detector at CERN, is to measure several aspects of the scintillation properties of lead tungstate ($PbWO_4$) crystals.

The light emitted by a 23 cm long crystal when excited with a radioactive source is collected by a photo-multiplier tube (PMT) and converted into an electric pulse which is then sent to our data acquisition (DAQ) system. For calibration purpose, a LED, controlled by a CAMAC LED driver, is glued to the PMT and can produce light pulses analogous to those coming from the crystal.

[1]leonardi@roma1.infn.it
[2]organtini@roma1.infn.it

The signal produced by the PMT is then split: one signal is collected by a CAMAC charge integrating ADC, a Lecroy 2249W, and converted into a digital value; the other is sent to a NIM discriminator and logic chain in order to provide the gate signal to the ADC module.

A "pedestal" functioning mode is obtained by generating a random gate signal. This signal is vetoed by the presence of some real signal on the PMT. This running mode is needed in order to estimate the contribution of the electronics to the signal coming from the PMT.

The DAQ chain functioning mode (PMT signal coming from the crystal, PMT signals due to LED flashes, or pedestal only signal) can be selected by activating the corresponding bit pattern on a CAMAC Input/Output Register.

As most of the needed electronics was already available in our laboratory and given the reasonably low DAQ rates expected, we chose the CAMAC standard as the best compromise between cost, reliability and performance.

To interface the DAQ electronics to our 120MHz Pentium PC, we employed a relatively cheap CAEN set composed of a C111 crate controller connected to an A151 ISA-bus board. The producer provides a simple software interface consisting of a C and Basic library of standard CAMAC functions working under MS-DOS: this allowed only a mono-task use of the PC and introduced a very high software overhead if one wanted to develop an integrated DAQ and monitoring program.

2 The Linux Operating System

A possible alternative to the use of the MS-DOS or Windows operating systems (OS) on a PC is represented by Linux.

Linux is a free Unix-like OS, originally developed by Linus Torvalds for Intel 386-based platforms and now ported to several architectures (Alpha, Power-PC, M68000, SPARC, etc.). The system consists of a kernel plus all the software utilities available from the Free Software Foundation GNU project.

Linux offers several advantages to an HEP environment:

- the OS presents a very high degree of software homogeneity with higher level Unix workstations (Digital, HP, Sun, etc.), widely used in most HEP experiments;

- free versions of the system software, compilers (C, C++, Fortran) and of several applications are readily available under the GNU licence and do not need the acquisition of additional software licences, thus reducing the total cost of the system;
- most of the CERN standard libraries are available[3] thus allowing the use of very popular packages such as GEANT, Minuit, and PAW on low cost PC's;
- all of the kernel source code is public so that the development of new device drivers is straightforward: this implies a rapid appearance of drivers for new peripherals and the possibility of developing one's own drivers for specific applications;
- a real multi-user/multi-tasking environment allows a transparent use of the machine to several users even during data acquisition;
- all the TCP/IP based network tools (telnet, ftp, NFS, etc.) are available natively on the OS, thus allowing a smooth integration of the PC in any existing network;
- POSIX compliance guarantees an immediate porting of new software packages to all Linux based platforms.

In view of all these considerations, at the beginning of 1996 we installed the Slackware 2.0 distribution of Linux on the PC available in our laboratory and upgraded the kernel to version 1.3.43 in order to support the AHA-2940W SCSI controller board present in our machine.

At this point, we turned to the problem of interfacing the PC to the CAMAC controller board under the new OS.

3 The C111 CAMAC controller device driver

A device driver is a software module which allow san interaction between user programs and some hardware device connected to the PC. Due to the low level of the software involved, these modules are usually part of the Unix kernel itself. From the user program point of view, the device is then seen as

[3] A Motif licence is needed to compile the Motif-based version of PAW++.

a simple file and the interaction with the peripheral proceeds via standard I/O operations (open, read, write, ioctl, close) directed to this file.

Under recent versions of Linux, device drivers can be separated "kernel modules" which are loaded or unloaded on request, thus avoiding a recompilation of the whole kernel every time a new device driver is added to the system [1]. This, together with the possibility of using high-level programming languages such as C to create the modules, greatly improves the development process.

As stated before, we used a CAEN A151 interface board to connect our PC to the C111 CAMAC crate controller. The board uses the ISA bus memory mapping mechanism so that it appears to the OS as a block of consecutive RAM addresses: sending commands to the controller or receiving data from it is equivalent to writing and reading blocks of 3 bytes to these addresses. The device driver we wrote[4] consists of a simple translation procedure from the read/write commands sent to the device file into the equivalent read/write commands to the board memory addresses.

Once the device driver was working, we provided an intermediate level library, both in C and Fortran version, which allows a more "CAMAC oriented" interaction with the board, i.e. a CAMAC command can be issued with a syntax like

```
call c111(Crate,N,A,F,Q,X,D)
```

which issues a complete command to one of the crates connected to the board and, e.g., retrieves the resulting status and data words in the Q, X, and D registers [3].

4 Data Acquisition and Monitoring

Using the aforementioned library we then developed a Fortran program to control the DAQ. We chose Fortran in order to allow as many physicists from our group as possible to get a copy of the program and modify it to their needs.

As the A151 board was developed for 8086-based PC's, we could not use the controller interrupt line to trigger the DAQ as the interrupt schema

[4]This module is heavily based on code developed by Ole Streicher for the Aachen DBCC CAMAC controller board [2].

changed in more recent machines and the board does not provide any configuration facility[5]. Thus we had to resort to the CAMAC LAM-signal polling which introduces some (non-critical) overhead in the system.

Thanks to the hardware set-up, the program can switch to several different DAQ modes (Crystal, Pedestal, or LED) by loading a bit pattern onto the CAMAC Input/Output Register. This allows for complex DAQ schemas, involving several run types, with no need for user intervention.

A simple Tcl/Tk [4] based user interface gives a complete control on the DAQ run parameters (run type, events requested), an on-line monitoring of temperature and event rates, and DAQ start and stop functions. All of the run conditions and user information are logged to a file and later used during data analysis.

Monitoring is accomplished by using the shared-memory routines available from the HBOOK package of the CERN libraries [5]: histograms are booked in the shared area and filled during DAQ. By accessing this area, it is then possible to show the histograms without interfering with DAQ. In our case, we decided to use PAW [6] as the monitoring program in order to implement a simple on-the-fly fitting procedure.

5 Performances evaluation

Given the structure of a CAMAC instruction cycle (load the N,A,F codes, execute the operation, retrieve the result) one would expect a few μs execution time. Using a test program which repeatedly read the content of one of the CAMAC modules, we observed a sustained DAQ rate of about 70 KHz. The factor lost from the possible maximum of 2–300 KHz can probably be ascribed to the software layers (OS contex switching and high level languages libraries overhead) superimposed to the low level I/O operations. The rate obtained is, however, well above the maximum of 10 KHz expected from the experimental setup, given the ADC conversion time of 106μs, and therefore completely suitable for our needs.

[5]This limitation is not present anymore in the new version of the CAEN CAMAC controller, the C111A.

6 Conclusions

We tested a DAQ set-up based on a CAMAC crate controlled by a PC running under the Linux OS. The system proved to be fast enough for all our applications and the multi-tasking OS allowed analysis to proceed in parallel to DAQ. We provided a C/Fortran interface to the CAMAC controller to grant the end-user the widest access to the facility and the possibility of using the CERN libraries.

Thanks to the use of low-cost technologies and the free availability of all the software packages, the final system resulted very cost-effective when compared to more sophisticated solutions.

References

[1] A. Rubini, *Linux Journal* **23** (1996) 10.

[2] O. Streicher, anonymous fto to `<ftp://ftp.ifh.de/pub/unix/linux/measurement/>`

[3] The latest version of our C111 device driver and user libraries can be obtained via anonymous ftp to `<ftp://hpl3rm06.roma1.infn.it/pub/linux/local/c111.0.0.3.tar.gz>`

[4] J.K. Ousterhout, "Tcl and the Tk Toolkit", Addison-Wesley, 1994, ISBN 0-201-63337-X.

[5] CN/ASD Group documentation, `<http://wwwcn.cern.ch/asdoc/Welcome.html>`.

[6] R. Brun *et al.*, "PAW – Physics Analysis Workstation", CERN Computer Center Program Library.