

Distributed control system for NSC tandem-LINAC

B P Ajith Kumar, E T Subrahmaniam & Kundan Singh

Nuclear Science Centre, Aruna Asaf Ali Road, New Delhi 110 067

Received 7 November 2000

The new control system for the tandem-LINAC accelerator system at Nuclear Science Centre (NSC), runs on a network of Pentiums under the LINUX operating system. Some of the computers are interfaced to the devices of the accelerator, using CAMAC, and run a server program. On the same network there are machines providing the operator interface, by running the client program. The client computers use the X-window graphics and shaft encoder knobs interfaced to them to provide the operator interface. The system supports the monitoring and controlling of all the accelerator parameters including the beam profile monitors, from any of the clients.

1 Introduction

The Nuclear Science Centre (NSC), an inter-university research facility, has a 16 MV tandem accelerator and a super-conducting heavy ion LINAC is under construction. The tandem has been running under a PC based system¹. A CAMAC serial highway was used for interfacing the devices to the PC. The addition of LINAC has put extra requirements like more number of CAMAC Crates and multiple operator consoles. The new system was designed to meet these requirements and to support more features like automation of some of the beam tuning operations. The design was done with the following basic guidelines; low construction and maintenance costs, adherence to established hardware and software standards, flexibility and expandability. To achieve these design goals we decided to use commercial off-the-shelf components where possible. Intel 80 x 86 based PCs connected by Ethernet forms the basic computer network required for the system. CAMAC standard is chosen for interfacing considering our existing CAMAC hardware. As the software is concerned, Linux operating system is chosen because of its quality, features like X-Window graphics and TCP/IP networking and availability of development tools. Easy availability of documentation for writing the CAMAC device drivers was another factor supporting the selection of Linux. A client-server design having multiple servers and clients was adopted to achieve flexibility and expandability.

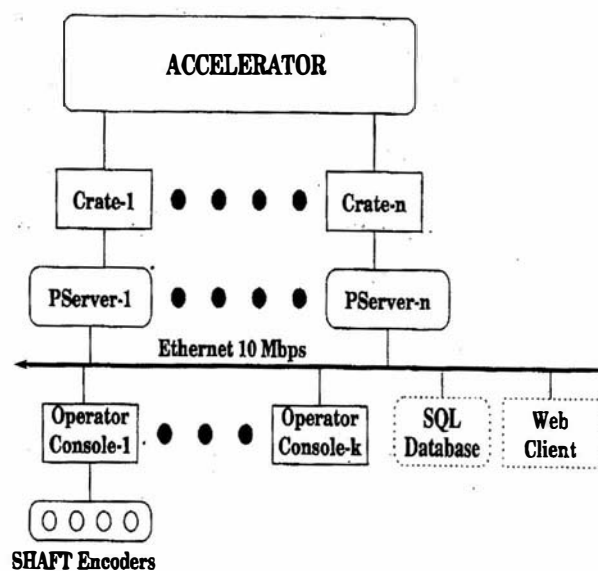


Fig. 1 — Pelletron control system hardware

2 Hardware

An outline of the control system hardware is shown in Fig. 1. A 16 port Ethernet hub is used for networking all the PCs together. The devices of the accelerator are connected to various modules plugged into the CAMAC crates. Most of them are standard modules like ADCs, DACs, output registers and input gates. Specially designed CAMAC modules are used for interfacing the beam profile monitor signals. Each crate is connected to a PC, using a home-made crate controller and

interface card. The PCs interfaced to CAMAC run copies of a server program, which routes all the requests to access any accelerator parameter signals connected to it. These are called servers. On the same network there are PCs providing the user interface. They are equipped with four shaft encoder knobs, connected to the PC using specially designed hardware, acting as incremental input devices.

2.1 BPM digitizer

The NEC beam profile monitor extracts the profile information by intercepting the beam by a rotating helical wire in X and Y directions. The output is processed and displayed on an Oscilloscope, which displays two peaks representing the profile of the beam in X and Y directions. The BPM digitizer module can select a BPM, digitize the beam profile signals, store it locally and transfer it to the computer through the CAMAC interface.

2.2 Control knobs

Optimizing the beam output from the accelerator require fine-tuning of many parameter values. The conventional input devices like mouse and keyboard are not suitable for this job. A set of control knobs is used for this purpose. When a control knob is rotated, it generates two-square waves whose frequency and phase depends on the speed and direction of rotation. A circuit containing counters processes these signals and each knob can be assigned to any analog parameter with the help of software. A circuit handling four knobs is directly interfaced to the PC ISA bus.

3 Software

The control software has to perform many functions like interacting with the user, accessing the parameter values by communicating the interfacing hardware, monitoring for alarm conditions, data-logging etc. The design goal was to write isolated modules implementing the various tasks and provide a method for communication between the modules. A client server design was chosen to achieve that. The server part of the program runs on computers connected to CAMAC and handles all the accelerator parameter signals connected. The server program takes care of the hardware details and the complexities of the signals connected to it. If any changes are made to the hardware, only the server program needs to be modified.

The authors have followed a distributed database approach and each server has a database storing the details of signals connected to it. There is no centralized database. The databases contain static and dynamic fields of information. The static fields are loaded from a file during startup and the dynamic fields are periodically updated with the actual parameter values through CAMAC. Three character strings, name, function and unit, uniquely identify each parameter in the whole system. For example "CPS031", "VC", "KV" identifies the charging chain power supply voltage control signal. All other details regarding this signal, like its CAMAC address, voltage limits, resolution etc. is only available with the server. The server program starts when the computer is started. It listens over a TCP/IP channel for client requests and establishes a client connection after verifying its authenticity.

The client programs provide the user interface, alarm services, logging etc. The client program accesses the accelerator parameters by sending command packets to the appropriate server. The packet will have the signal identification and the operation to be carried out and the reply will contain the requested information. The existing client program's main function is to provide a user interface. The GUI is implemented using X- Windows and Motif library. The total number of parameters is grouped into "pages" for the convenience of displaying. The user defines the page layout by editing text files, which will have the parameters identified by the three strings mentioned earlier. All the available pages are loaded during startup and the server for each parameter is identified. The user requests the client to access the various parameters by clicking the mouse and by turning the shaft encoder knobs. Client sends requests to the servers for getting and setting various parameters controlling the operation of the accelerator and displays the results. The user can define the interlocks and alarm conditions in a text file. The file is loaded during startup and the parameters are periodically checked.

The details of accessing the parameters are hidden from the client and all it requires accessing a parameter is its identification. A communication library containing routines to establish connection to the server is written. It is easier to write small client programs for specific purposes using this library. This feature will be used for developing programs

for partial automation of the accelerator operation. Multiple clients may try to access the same parameter but the server serializes the requests to avoid any conflict.

4 Conclusion

The system is currently running with four server machines and two clients. The numbers will

increase with the addition of the LINAC coming up. The system is highly cost effective and expandable. Adding more CAMAC crates, servers or clients does not require any change in the software.

References

1. Ajith Kumar B P, Kannaiyan J, Sugathan P, Bhowmik R K. *The NSC 16MV tandem accelerator control system. NIM A*, 343 (1994) 327-330.