A DISTRIBUTED CONTROL SYSTEM FOR NSC TANDEM-LINAC

Ajith Kumar B.P., E.T.Subramaniam and Kundan Singh
Nuclear Science Centre, New Delhi - 110 067, INDIA

Abstract

The new control system for the Tandem-LINAC accelerator system at Nuclear Science Centre is a client-server design running on a network of PCs under the GNU/Linux operating system. Various devices of the accelerator are connected to CAMAC Crates located around the machine. The CAMAC Crate controllers have embedded PCs running a server program that takes care of the devices connected to it and also listens on the Ethernet port. 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.

INTRODUCTION

The Nuclear Science Centre, an Inter-University research facility, has a 16 MV Tandem accelerator and a super-conducting heavy ion LINAC is under construction. The Tandem was originally supplied with CAMAC Serial Highway Interface and a DEC PDP11 based control system. The PDP11 was replaced by an IBM PC based system running DOS before the accelerator was commissioned [1]. The addition of the LINAC demanded more number of CAMAC Crates, multiple operator consoles and the ability to run special purpose programs to condition the resonator cavity, automatically setting the amplitude and phase of RF etc. The new system was designed to meet these requirements and also to support features like partial automation 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 wherever possible. Intel 80x86 processor based PCs connected over ethernet forms the basic computer network required for the system. CAMAC standard is chosen for interfacing, considering our existing CAMAC investment and expertise in designing CAMAC hardware. As the software is concerned, GNU/Linux operating system is chosen because of its reliability, features like X-Window graphics, TCP/IP networking and availability of development tools. Easy availability of documentation for writing the CAMAC device drivers was another deciding factor. A client-server design having multiple servers and clients [2] was adopted to achieve flexibility and expandability.

HARDWARE

An outline of the control system hardware is shown in Fig.1. A 16 port Ethernet switch 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 ADC, DACs, Output Registers and Input Gates. Special modules were made for interfacing the Beam Profile Monitor and the LINAC resonator cavities. Each Crate houses a home-made Crate Controller with built-in single board computer having PC-104 interface and Ethernet port. On powerup the controllers boot from the network and run copies of a server program, which services all the requests to access any accelerator parameter signals connected to the Crate. 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.

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. Distance between these peaks and two fiducial points provides information about the spacial position of the beam. The BPM digitizer module is capable of selecting the BPM, digitize the beam profile signals, store it locally and transfer it to the computer. It has been implemented as a CAMAC module at present but will be redesigned as a controller directly interfaced to an embedded computer.

Control Knobs

Optimizing the beam output from the accelerator require fine-tuning of many parameter values. Control knobs are more suitable for this job compared to the conventional input devices like mouse and keyboard. A set of control knobs are implemented using optical shaft encoders that 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. The shaft encoder interface was earlier designed for the PC ISA bus and later redesigned for the parallel port.

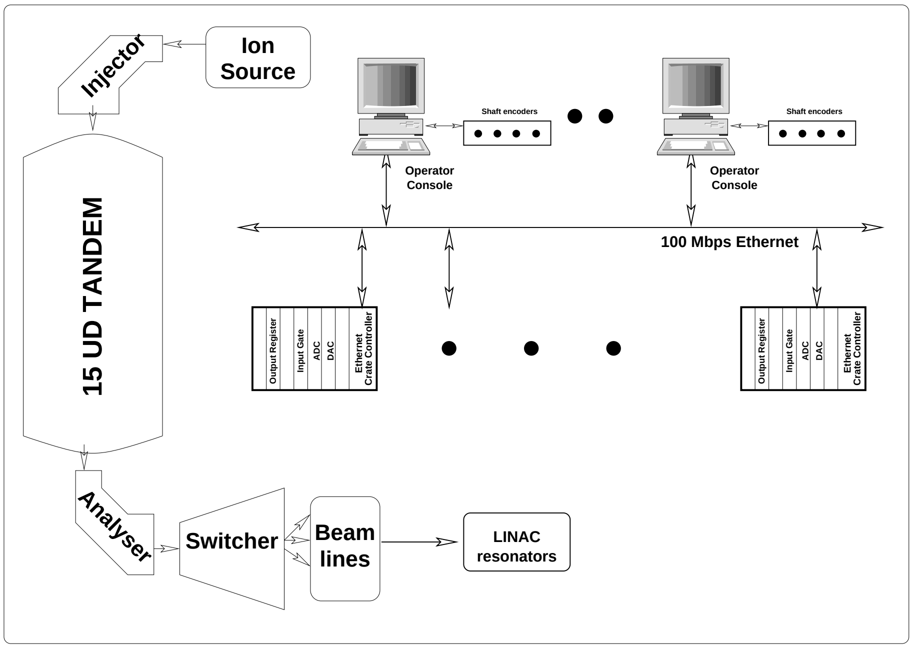

Figure 1: DISTRIBUTED CONTROL SYSTEM HARDWARE.

CAMAC Crate Controller running GNU/Linux

Initially each CAMAC Controller was connected to standard PCs using the ISA bus. It required the installation and maintenance of the operating system and the server program on each PC. The Crate controller was redesigned [3] to eliminate the need of standard PC by employing the single board computers with PC104 interface and Ethernet support. The crate controller circuits communicate to the computer through eight bytes in the PC I/O address space using the PC104 connector. All the Crate controllers boot from a central server and starts a copy of the server program with server specific database. This approach eliminates the local installation of operating system and software on the controllers resulting in easy maintenance. The BIOS of the SBC was modified to include the "etherboot" program for the Ethernet interface. The software for remote booting is from the Linux Terminal Server Project. [4]

SOFTWARE

The most important function of the control software is to enable the operator to access all the machine parameters from a console. The operator console should have a good graphical user interface and suitable input devices. We also wanted the capability to run several special purpose programs simultaneously for controlling/monitoring a small group of parameters. To achive these design goals a clent server design was implemented were all the hardware details are handled only by the server program and all other functions by separate client programs. Context diagram of the system is shown in Fig. 2.

The details of accessing the parameters are hidden from the client and all it requires for 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 are being 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.

The Server Program

Copies of the the Server program runs on each computer connected to CAMAC and handles all the accelerator signals connected to the Crate. The server program takes care of the hardware details and the complexities of the signals connected to it. If any changes made to the hardware, only the server program needs to be modified. We have followed a distributed database approach and each server has a database storing the details of signals connected to it. 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 communication between server and clients is mainly through four generic functions. "Get

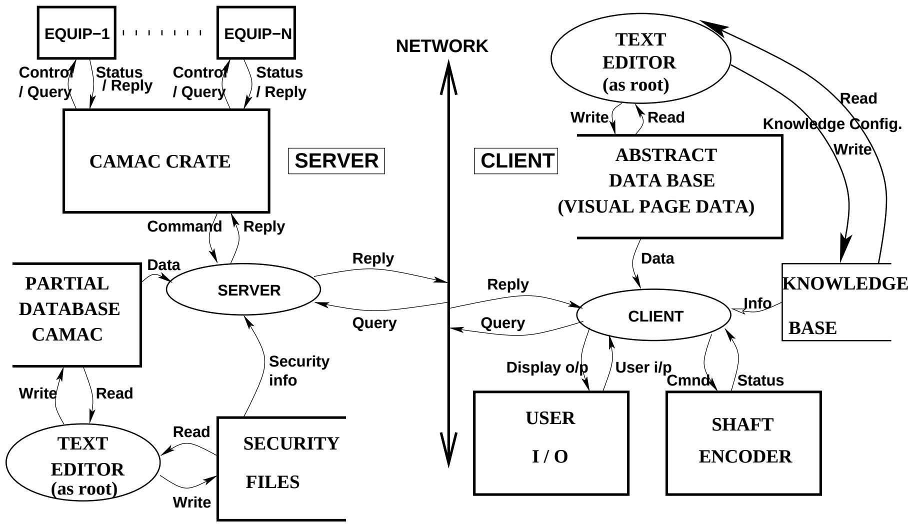

Figure 2: Data flow diagram of the Control System.

Value” and ”Set Value” for analog parameters and ”Get Status” and ”Set Status” for logical parameters.

The Control Console

There is one client program that provides the operator console. 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 during startup. The opearator can modify the analog parameters by assigning them to any of the shaft encoder knobs. Requests are sent to the servers for getting and setting various parameters controlling the operation of the accelerator and displays the results. Mutliple operator consoles are supported. If the same parameter is assigned to shaft encoders at different consoles, the idle knobs will automatically follow the knob that is currently changing the assigned parameter.

Alarms and interlocks

Dependancies between parameters can be defined in a text file that is loaded during startup and the relationships examined periodically to take necessary actions. The interlocks, alarm and a diagramatic display of the machine are currently implemented using this facility. The dependancies are specified by keywords like ”IF” ”THEN” and specifying allowed minimum and maximum for analog parameters. For example a condition can be defined to switch off the charging chain if the charging powersupply is below certain volatge level. A diagramatic display to show the status of the machine is also driven by the same facility. 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 are being 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.

CONCLUSION

The system started operating with four CAMAC Crates handling signals from the Tandem and four more has been added with commissioning of the first LINAC module. There are two operator consoles in the control room and a mobile console. The system has shown very high reliability, scalability and ease of use. Adding more CAMAC Crates, Servers or Clients does not require any change in the software. The cost of the system is very low due to the inhouse development of hardware modules and the free availability of the GNU/Linux operating system and development tools.

REFERENCES

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

[2] CAMAC Crate Controller running GNU/Linux. Ajith Kumar B.P. et al , DAE(2003)

[3] Distributed control system for NSC tandem-LINAC. Ajith Kumar B.P. et.al, Indian Journal of Pure and Applied Physics Vol 39 Jan-Feb 2001.

[4] http://www.ltsp.org/