Development of a CAMAC Distributed Control System

S.M. Tang, J.X. Yan, Y.L. YU
H. Luo, X.N.Li, J.W.Zhang
Institute of High Energy Physics(IHEP)
P.O.BOX 918, 100039, Beijing, China

Abstract

A model of a distributed control system(DCS) based on CAMAC front computers has been developed. It simulates actual control of accelerator′s equipment of BEPC. This paper describes the hardware configuration, the software structure, and the system database and data management.

1 Introduction

In existing control system of BEPC, most CAMAC I/O jobs are executed by a single host computer. A KSC 3922/2922 Q-bus CAMAC adapter connects host computer with local stations. In the future the system should be further improved and evolved into a much more distributed configuration. Under this situation we have developed a new model of a CAMAC-based distributed control system. This model can be an alternative of BEPC upgrading. In this new architecture, one distinct feature is that local stations are intelligent. A local station can control its own devices autonomously. Because more jobs are executed in the Frond End Computer (FEC), the load of host computer can be reduced. In this model, we also consider about real-time, reliability, expandability and maintainability of the system. Applications of local station are especially developed, but operator interface and system host are considered as well.

2 Architecture

2.1 System architecture

The system architecture is based on LAN. It can be described as ′man machine interface -- system server -- local control station′. Figure 1 displays the layout of the control system. A SUN workstation and its X-terminal provide a graphic interface. The system server VAX-II manages the database, schedules real-time tasks, and downloads data tables to SUN workstation or local station. The frond end computer is KS-3968Z2A, and its applications are developed under VAXELN real-time operation system. KS-3968 executes data acquisition, status monitor and local equipment control by CAMAC interface, and communicates with remote host and exchanges all kinds of messages over Ethernet.

2.2 Description of the procedure

As a remote console SUN workstation sends out commands. These commands are disposed and transferred to KS-3968 by VAX-II. According to these commands, KS-3968 completes corresponding tasks. In the mean time, KS-3968 sends packages of messages to VAX-II at a rate of twice per second. These messages are stored in the database of VAX-II, and some of them are displayed on the screen of the X-terminal. Two types of communication protocols, DECnet and TCP/IP, are employed. The protocol between KS-3969 and VAX-II is DECnet, and between VAX-II and SUN workstation is TCP/IP. Among these computers VAX-II routes all kinds of messages.

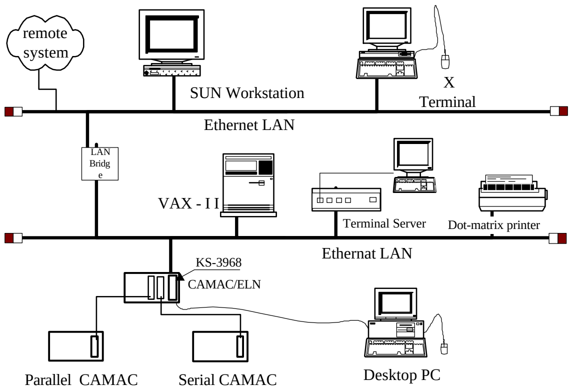

Figure 1. Control System Architecture

2.3 Local station

The local control station is an intelligent front-end system. It is composed of front-end computer, CAMAC I/O interface and local console(figure 1). Front-end computer KS-3968 contains Digital Corporation′s 20 megahertz rtVAX300 processor module and 8M bytes on-board system RAM. VAXELN runtime applications, developed on a VMS™ system, are downloaded over Ethernet for execution. The CAMAC crate in which KS-3968 inserts is called as system crate. The parallel Branch Driver(BD) and Serial Branch Driver(SBD) are inserted in the system crate, so that a parallel CAMAC crate and a serial CAMAC crate are extended. The BD connects with the parallel crate controller CCU2(crate Controller Undefined-2 type). The SBD connects with the serial crate controller SCC (Serial Crate Controller). Since serial CAMAC crate can be allocated far from the system crate, so the KS-3968 controls long-distance equipment by SBD and SCC. The local console is an IBM-PC/486 which connects with the first serial port of KS-3968, and is mainly used for local monitor and tuning.

The architecture and functions of this model system are suitable for all local stations in principle, but now it simulates the power local station of BEPC. There are 24 devices, 48 analog I/O signals and 120 digital signals in the system.

3 local Applications

The software on the local station should be highly reliable and real-time, the applications should have strong ant-disturbance and error tolerance. So we take following measures:

- modular structure of process
- management of public data area.
- function library of devices
- execution of multitasking

The development of applications is divided into three steps. First is to build the function library for I/O devices. These functions are used to access CAMAC I/O modules, therefore they are fundamental. Next step, simulating actual requirements, the real-time multitasks were programmed and tested. Third step, the whole system was commissioned and availability of applications was improved.

3.1 CAMAC function library

The CAMAC function library is very important that the procedure of programming can be simplified and sharable for other programmers. The library is divided into two sections, basic CAMAC routines and extended CAMAC I/O routines. Basic CAMAC routines correspond to basic CAMAC I/O operations, such as CAMAC read and write. Extended CAMAC I/O routines are used to execute parallel and serial CAMAC I/O operations.

3.2 Multitasking

Applications of local control station are developed under VAXELN Toolkit which is a powerful software product for real-time controls. VAXELN Toolkit is compatible with VMS and is installed on VAX-II. All programs are written in VAX C.

The tasks include communicating with the system server, command processing, alarm handling and equipment control. The concurrent execution of processes is known as multitasking. Multitasking uses resources of the CPU more efficiently and enables the application to response to events quickly. According to actual conditions, all processes are included in two jobs, CAMAC job and communication job. Each job consists of a master process and several subprocesses. Master process can create subprocesses. The scheduling of processes is a method called preemptive priority scheduling.

When a remote command arrives, CAMAC job immediately creates a relative subprocess which executes serial CAMAC operations such as setting and sampling device parameters or status. All processes of CAMAC job use a mechanism of synchronization to access the same CAMAC channel concurrently.

Communication job receives the initial information and commands to local station from the system server, then sends these commands to CAMAC job. It also sends all kinds of messages to the system server. In order to improve real-time capability and reliability, communication job is implemented as a concurrent server. This server can create multiple processes for communication, and simultaneously connect with remote hosts.

3.3 Public data area

Jobs (or processes) executed on the same VAXELN node share data in a common memory region called an area which resides in memory. This area is called public data area. In this area the data of local station are stored. Some of them are downloaded from VAX-II, including module tables, operation tables, mode tables and other state values. The relations among area and jobs are shown as Figure 2.

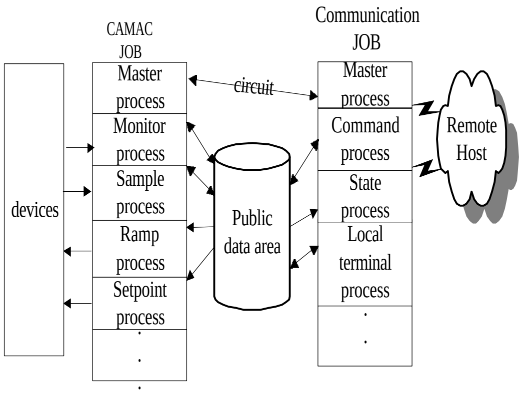

Figure. 2 Public Data Area and Processes

3.4 Local test

When the application software was written down, system test was taken. These tests are mainly aimed at the local controller′s speed of operation and overhead of memory. On the local station, the software, which controls 24 devices, occupies only 10 percent of memory. It costs less than 10ms to execute a *setpoint* operation of all devices. These results proved that the software design meets the system requirements and equipment under the control of one KS-3968 can be expanded about 4 to 6 times.

4 System server

In the system model, VAX-II is regarded as system server which manages the database, coordinates and checks all computers in the system. The applications of system server include two main sections, database and communication processes. Database is essential for system, and multi-processes can access it. Communication processes transfer commands and data between operator interface and local station.

There is no available database toolkit for the VAX-II, so a centralized database was generated by a dedicated program of C language. It creates tables, insert or delete entries, and also have the functions of input, modification, display, printing and so on. The database consists of two parts: initialized data tables and dynamic data area. Initialized data tables are stored in the hard disk in files. These tables include four tables: local station operation table, console display table, machine mode table and device value table. They can be downloaded to the local station or SUN workstation. Dynamic data area is installed in P0 virtual address space. In order to increase the speed of data exchange, run-time data are stored in this area.

Commands and exchanged data are carried in packages. The formats of these packages among hosts are coincident. There are two types of packages: command package and local message package. Command package consists of seven commands: *ramp*, *stdz*, *setpoint*, *set_on_off*, *monitor*, *abort* and *update*. Each command corresponds to a specific control process. Local message package includes all local messages which are sent from the local station to the system server. It consists of periodical data message, alarm message and operation validity message.

Communication processes are used for downloading data tables, transferring command package and local message package. Four processes perform these functions. Since the data formats between VAX-II and SUN workstation are different, so the conversion is taken to resolve this problem.

5 Operator interface

A SUN workstation and X-terminal are used as an operator interface. The operator application is developed with JAVA under a UNIX environment. JAVA is object-oriented, architecture-neutral, portable and multithreaded, and its Abstract Window Toolkit(AWT) provides the standard graphical user interface(GUI) elements. Now Java is extensively used for Web pages, but it is also feasible to develop other applications. Although we are employing Java to develop applications in our control system is just an attempt, according to the results we have got, it came up to our expectations.

6 Conclusion

In the standard CAMAC system, whether parallel or serial can not communicate each other directly, but by using intelligent controller and network this problem is overcome. If several local controllers are connected by Ethernet, jobs can be distributed among these local stations, and local tasks will be balanced among these stations.

To preserve the existing hardware and software investment in BEPC control system is a major concern of this project. Our main aim is to propose an architecture that permits smooth upgrading. Since VAXELN is compatible with VMS^{TM}, if an original non-intelligent controller is replaced with KS-3968, the change of local devices is slight, and it permits partial updating. We hope this design will be to employ in BEPC control system.

Acknowledgment

We would like to thank Dr. H.Y.Luo for her contribution to this work.

Reference

[1] Introduction to VAXELN, Version 4.0, Digital Equipment Corporation. June 1989

[2] Luouq, etc. The New Accelerator Control System of GANIL. Processing of the Third International Conference on Accelerator and Large Experimental Physics Control System, Berlin, Germany, October 18-23, 1993. (P96-102)

[3] Wang, D.W. Liao, Design and Applications of Distributed Control system. 1993

[4] Yi, Y.P.Cheng, JAVA Manual. 1997