CONTROL SYSTEM FOR NEW COMPACT ELECTRON LINAC

A. Chepurnov, A. Alimov, D. Ermakov, V. Shvedunov. Institute of Nuclear Physics, Moscow State University. F. Nedeoglo, N. Kochetkov, D. Komissarov, A. Lisjutin, A. Nikolaev, O. Novojilov, I. Jyltsov. Department of Physics, Moscow State University, 119899 Moscow, Russia

Abstract

A task to develop a control system (CS) for the first member of a newly designed family of compact, high-current, continuous wave (CW) electron linacs being planned for industrial installation was undertaken. During the first stage of CS development, flexible modern DAQ boards were installed in conventional PC and were used to control subsystems of the accelerator. To normalise signals for acquisition and generate signals to control, existing analogue blocks were applied. All data acquisition and control algorithms were implemented under PC version of LabView 4.0 together with some simple operator interface to test and study subsystems of the accelerator. The PC at the front-end was connected through Ethernet to another remote PC. It worked under Linux and supported operator interface and simple data-base. During the second, final stage of CS development "industrial type" control system was developed. Front-end PC with analogue electronics was replaced with a few members of "smart device" family -- intelligent front-end controllers, working via CAN-bus under DeviceNet high level protocol and using the same operator console under Linux.

1 INTRODUCTION

A new CS was developed for a new small CW linac. The linac is the first that belongs to a new family of industrial CW linacs. These reliable, small, inexpensive high-power modular accelerators will produce beams with energies from 0.6 to 6.0 MeV in increments of 600 keV, each with a current selectable from 0 to 50 mA.

In the first beam tests, a single section model has provided a 600 keV, 10 mA, 6 kW electron beam at a 20 mA gun current, thus demonstrating a design with 50% capture efficiency and beam energy [1]. Principal parameters of the linac are listed in the Table 1.

Table 1.

Output beam energy	0.6 MeV
Beam current	0 to 50 mA
Maximum beam power	30 kW
Length	1.2 m
Weight	~70 kg
Gun/klystron high voltage	15 kV
Plug power consumption	~75 kW
Electric efficiency	~40%

Difficulties in designing and constructing the CS lied in the fact that new accelerator was constructed from newly designed components: new rf channel with new klystron, new accelerating section, new electron gun, and new set of high voltage power supplies. Moreover, final accelerator scheme was not fixed at the beginning of construction and changed during the process. So, CS should also support experiments to study the entire system as it changed continuously. At the same time, both the accelerator and CS should fit the requirements imposed by installations for industrial applications. Development of CS was divided into two stages and corresponded with the stages of accelerator development.

2 STRUCTURE OF CONTROL SYSTEM

The CS (Fig.1) consisted of three levels – non-real-time top level; soft real-time middle level where relatively slow algorithms are implemented; and front-end level with fast control algorithms, hardware locking, fast feedback loops, and signal conditioning hardware.

2.1 Control system during research phase

The CS fulfilled data acquisition and presentation functions other than control functions during initial testing of different parts of the accelerator: klystron, e-gun, etc. It is important because the main task at this stage was to carefully study newly created parts of the accelerator as objects to be controlled.

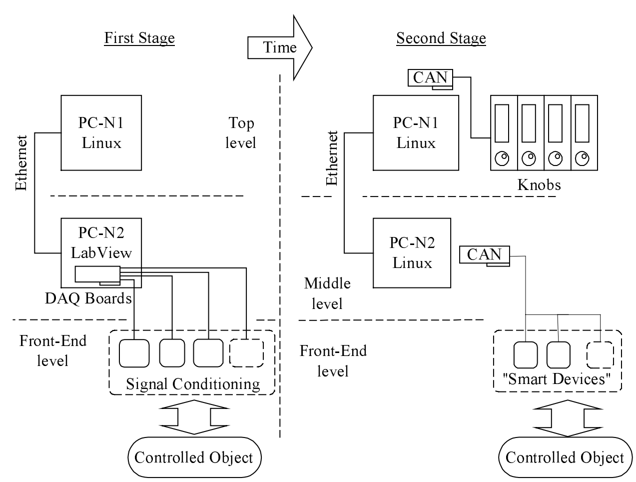

Figure 1. Stages of control system development.

The top level consisted of PC-N1 running under Linux

2.2.x. It communicated with the middle level through a separate segment of optically isolated Ethernet.

The middle level consisted of PC-N2 together with ISA-compatible DAQ add-on boards running under Windows95. Boards of two types had the following overall I/O characteristics: 32 channels of ADC with 12 bits 2 μs/channel resolution; and 2 isolated channels of DAC with 12 bits resolution and 32 digital I/O. Signal conditioning hardware and galvanic isolation were used between accelerator and DAQ boards. Interlocks and fast feedback loops were implemented in hardware.

2.2 Control system for industrial application

Finally the CS should be converted from its form during research to a design for industrial application. CAN-bus fieldbus was to be used on both levels of CS.

CAN-bus ISA compatible adapter for PC was developed. It is based on SJA1000 CAN-bus controller and provides fast access to the CAN controller by direct memory mapping. CAN-bus interface also ensures galvanic isolation.

To support man-machine interface knobs-type module was designed. Each device consisted of optical encoder, four lines of high brightness LCD, and four keys with corresponding LEDs. CAN-bus and RS-232 (optional) interfaces were used to communicate with the host computer. The device had a high level of scalability because of its internal modular structure. Separate modules such as LCD display, a module which processed signals from optical encoder and keys, and CAN-controller, are connected via I^2C serial bus and based on separate PIC16C6x/7x single-chip microcontrollers. DeviceNet high level protocol over CAN-bus was implemented in the module responsible for communication over CAN-bus [2].

CAN-bus-ISA adapter installed in PC-N2 controls embedded controllers that belonged to a family of “Smart Devices”--intelligent controllers which support functions of real-time digital feedback control, data acquisition and processing [3]. New members of this family are DeviceNet compatible now and based on new DSP from TI with on-chip CAN-controller (TMS320C24x). The controllers were designed to support local PID feedback control of RF field in the accelerator structure.

A special kind of “Smart Devices” was designed to measure current of high voltage power supplies. It consisted of two separate sub-modules connected via single direction fibre-optic link. The first sub-module was embedded directly to a high voltage power supply. It digitised the current of the high voltage power supply for the klystron and electron gun separately by measuring small voltage of low-ohm shunts relatively 15kV. Then it transmitted processed data of the current value to the second sub-module connected to CAN-bus which was represented in the network as a DeviceNet compatible device.

3 HIGH LEVEL SOFTWARE

High level application software of CS was based on an architecture with Distributed Shared Memory (DSM) [3,4].

3.1 Application and system software on PC-N2

While the software of top level was the same for both stages of CS development, the software of middle level differed from stage to stage.

Application software was developed for the “research stage” under LabVIEW 4.0 and consisted of three components: data acquisition and processing; software control algorithms (for example, slow PID feedback control); and local man-machine interface and server program to support mirroring of DSM. Flexibility of combination of LabVIEW and DAQ boards allowed us to perform different kind of experiments with accelerator hardware and to develop control algorithms.

In the second stage, PCs communicate via CAN-bus. DeviceNet high level protocol for CAN-bus was chosen from the three most widely used protocols (CAL/CAN-open, DeviceNet and SDS) which are used over CAN-bus in industrial and automotive applications. DeviceNet compliant protocol stack and different CAN-bus device drivers were developed. [2].

3.2 Software of top level

As was mentioned above, PC-N1 ran under non-real-time OS Linux 2.2.x.

Linux is a dynamically developing, modern, POSIX-compatible UNIX-like OS. It is freely available together with its source codes and well-documented. Our experience showed that a wide set of application software and tools, and a high level of reliability and supportability could be achieved with OS Linux in desktop, network and server applications [5]. The structure of top level software is shown in Figure 2.

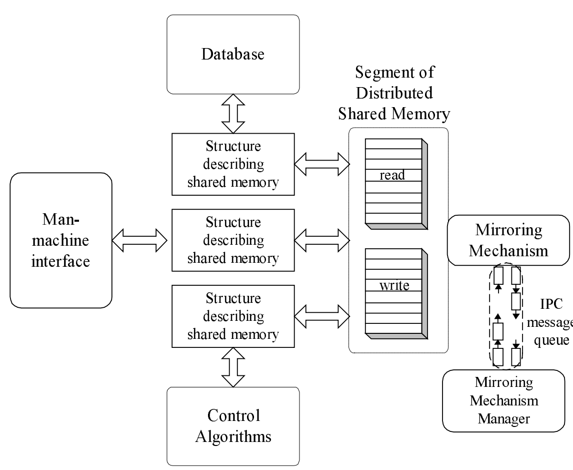

Figure. 2. Structure of top level software.

Man-machine interface was developed with the help of LessTiff library for X-Windows system under Linux.

Modules of application software (man-machine interface, control algorithms, database, etc.) could watch and control an accelerator through a segment of DSM. Every module of application software accessed the segment of DSM through an individually described structure. Data from particular blocks of DSM belonged to other components of CS (for example, modules of front-end level) were reflected in the segment of DSM in top level (Figure.3).

3.3 Mirroring mechanism

Segments of DSM, together with mirroring mechanism, were used to reflect data describing current state of an accelerator and necessary for operating of application software. Mirroring mechanism lied hidden from application software. It was implemented in software with Ethernet interlayer. It allowed different hardware and software components to be used at front-end level during first and second stages of CS development without any software changes in top level.

Software components accessing the segment of DSM, and not responsible for mirroring, might know nothing about inter-level communication construction and were not concerned with the appearance of data. This approach ensured rather clear application program interface which simplified work of programmers and made possible the independent development of parts of application software as mirroring algorithm for different types of hardware.

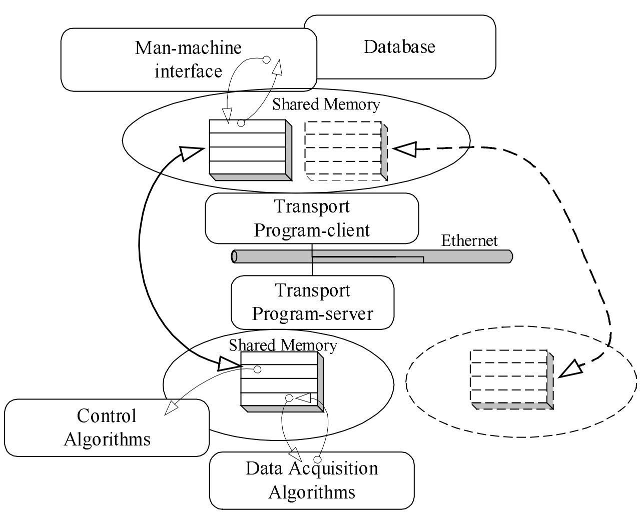

Figure 3. Structure of Distributed Shared Memory.

Mirroring mechanism implemented on top level might be modified dynamically by the Mirroring Mechanism Manager – a special-purpose software process. The process controlled the software module responsible for transportation and replication of different segments of shared memory. Physical properties of measuring parameters and type of communication were known in the initial stage of CS development and defined order and period of replication of different DSM blocks.

4 CONCLUSIONS

New software and hardware components usable in the design of new, modern control systems of an accelerator was developed. Based on the described components in the above, a CS for small linacs was constructed.

A two stages approach - DAQ boards together with LabView used during the initial “research” stage, and embedded controllers compatible with DeviceNet high protocol connected via CAN-bus used in the final stage - looks attractive and useful.

CAN-bus-ISA adapter and new members of “Smart Device” embedded controllers allow construction of an effective CS and perform general control tasks.

The DSM approach allowed us to develop application software for CS more quickly and reliable due to simple API. Future upgrading of the system is expected to be simplified.

REFERENCES

[1] A.S. Alimov, D.I. Ermakov, B.S. Ishkhanov, E.A. Knapp, et al. A compact industrial high-current continuous wave electron linac .// Proceedings of the 1999 Particle Accelerator Conference, p. 2555 (New York, USA, 1999.)

[2] A. Chepurnov, D. Komissarov, F. Nedeoglo, Device Net implementation under Linux for use in control system of a particle accelerator // These proceedings

[3] A.S. Chepurnov, A.A. Dorochin, K.A. Gudkov, V.E.Mnuskin, A.V. Shumakov, Family of Smart Devices on the base of DSP for Accelerator Control,// Proceedings of Int. Conf. on Accelerator and Large Experimental Physics Control Systems, W2B-d (Chicago, Illinois USA,1995).

[4] A.S. Chepurnov, A.S. Alimov, A.A. Dorokhin, B.S. Ishkhanov, V.M. Lipgart, V.E. Mnuskin, S.A. Kosterin, V.I. Shvedunov, A.V. Shumakov, Gradual Upgrading of the Accelerator Control// Proceedings of the Int. Workshop on Controls for Small and Medium-Scale Accelerators, p.169 (KEK, Tsukuba, Japan Nov.1996).

[5] D. Komissarov, F. Nedeoglo, A. Chepurnov, Linux and RT-Linux for accelerator control - pros and cons, application and positive experience. // These proceedings.