DESIGN AND DEVELOPMENT OF A NEW CONTROL SYSTEM FOR THE ELETTRA LINAC

L. Barbina, D. Bulfone, S. Filippi *, M. Lonza, R. Marizza, P. Michelini, A. Tomicich
Sincrotrone Trieste, Padriciano 99, 34012 Trieste, Italy

ABSTRACT

The injection system of ELETTRA relies on a high energy linac which has been bought as a turn-key machine. The low level of reliability and flexibility of the original control system has led to the decision to completely redesign and integrate it with the ELETTRA control system which has proven high efficiency in the past years of operation. This paper describes the new software and hardware architecture developed.

1 INTRODUCTION

Although in the past three years the ELETTRA linac has provided electron beams with constantly increasing reliability, the improvement of the efficiency remains one of the most important goals. For this reason a complete revision of some sub-systems like the timing system, the diagnostics and the vacuum system has been commissioned [1].

The ELETTRA linac consists of a 100 MeV pre-injector [2] built with traditional technology and seven SLED accelerating sections [3]. All the equipment, including the modulators and klystrons, are located in the eighty meter long "klystron room" where also the control system is placed.

The existing control system mixes its monitoring and control features together with the interlock functionality. Being based on the control system software the protection of sophisticated and expensive machine components does not provide the requested level of reliability.

A simple revision of the existing control system is not enough to improve the linac efficiency as both the system architecture and the hardware interface are to be modified. It was finally decided to rebuild the system with different technologies and integrate it with the ELETTRA control system [4].

2 SYSTEM ARCHITECTURE AND HARDWARE INTERFACE

The high level of electromagnetic noise present in the klystron room forces us to keep the signal cables as short as possible in order to avoid interferences.

A distributed architecture where the input/output interfaces are close to the sources of the signals satisfies the requirement (figure 1). The machine has been divided into several "plants", i.e. the gun, the pre-injector and the seven SLED accelerating sections, each controlled by one Equipment Controller (EC), made of a VMEbus crate with several input/output modules and a CPU board with a Motorola 68040 microprocessor. This modularity enables us to independently operate each of the plants and allows a smooth installation of the new control system.

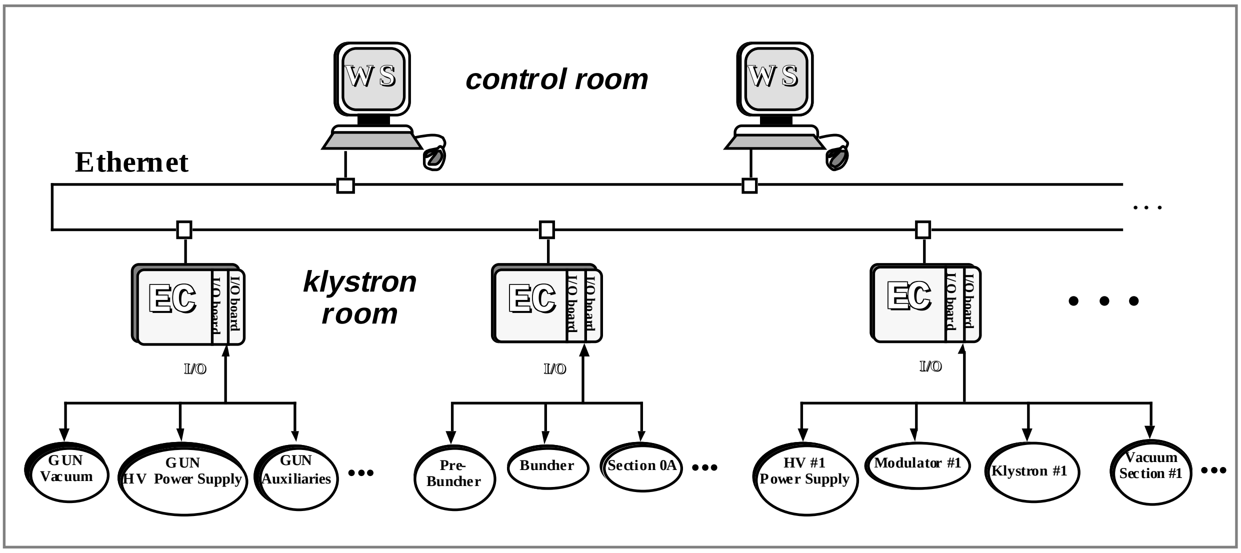

Figure 1: Linac control system architecture.

* Hosted by Sincrotrone Trieste for the thesis work.

Another major concern is the grounding of the plants which could induce additional noise on the signal lines. For this reason all the input/output signals of the ECs will be isolated from the VMEbus and with respect to each other.

Special care will be taken in the treatment of the signals before the acquisition: conditioning and filtering will guarantee noise rejection while keeping good dynamics of the signals.

The control system has to manage about 500 input and 200 output signals. Beside the "static" inputs, some impulsive signals must be acquired; they are few microsecond long pulses with a repetition rate of 10 Hz, which is the injection frequency of the linac. Special sample-and-hold boards will be adopted for this purpose. They will be triggered by a delayed 10 Hz timing signal and will read only the value of the interesting part of the pulse.

At the presentation level UNIX workstations with X11/Motif graphic software are used as operator consoles. Workstations and ECs are connected by a fibre-optic Ethernet link to avoid problems of grounding and noise.

3 THE INTERLOCK SYSTEM

The interlock actions for the machine protection will be assured by a mixed system made up of hard-wired devices and Programmable Logic Controllers (PLC). The interlock logic is structured in four hierarchical levels and defines a sequence of enabling signals for the klystron filaments, high voltage, radio frequency, etc.

Most of the inputs are digital signals but in some cases analog signals can generate interlock events whenever a given threshold is exceeded; dedicated comparator boards are foreseen for this purpose.

4 THE LOW LEVEL SOFTWARE FRAMEWORK

A new-concept low level software framework has been developed for the linac control system. The first goal was to ease the programming work and to simplify the system configuration and maintenance by enhancing the modularization of the software elements. The second was to improve the system performance especially in the high-level communication part.

The design is based on the concept of distributed data-base where all the input data of the machine are stored in local data-bases resident on the ECs memory which are continuously refreshed by reading the field.

A client read-request from a workstation is served by a *service request manager* (figure 2) with a simple query on the Local Data-Base (LDB) instead of directly accessing the field. Working with this dynamic copy of the field has several advantages:

- reduced request wait time in a client-server architecture, since the data retrieval is very fast; this also allows to pick up multiple value structures
- automatic monitoring of the devices, which are continuously polled
- to avoid several accesses to the field when different clients request the same data.

The LDB is updated by some *poller processes* that communicate with the *device controllers* by means of *logical device drivers* and a shared memory; *device controllers* can access the input/output hardware through *physical device drivers*.

In case of a write-request by a workstation client, the *service request manager* directly talks with the *device controllers*.

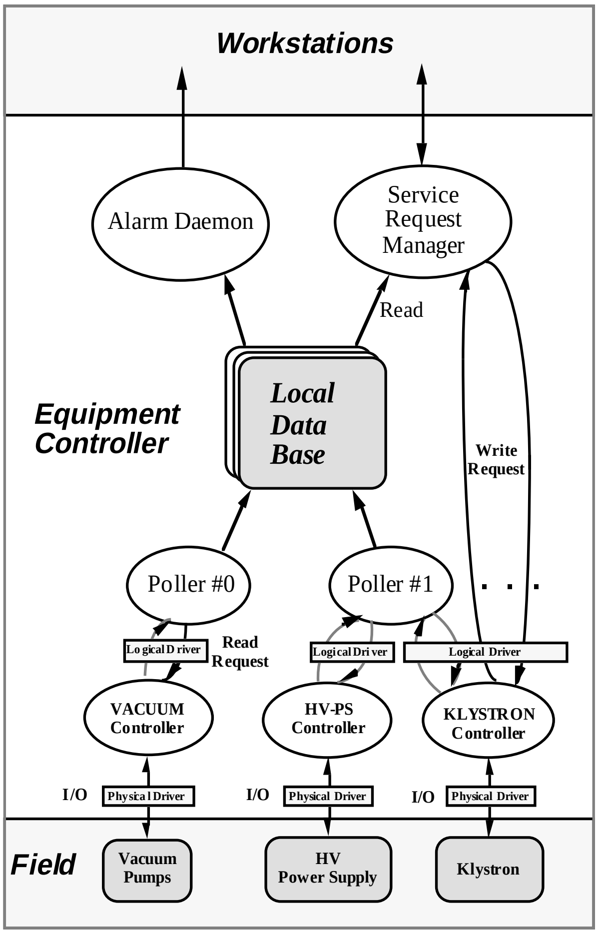

Figure 2: The low level software framework.

Each *device controller* is dedicated to a piece of equipment like a power supply or an amplifier, and is in charge of doing all the control tasks which concern that device. The *poller*, the *service request manager* or any other dedicated process can ask the *device controller* to do an action by sending a command request which can consist of a simple input/output access or a more complex action like beginning of a ramp or starting a loop. A sort of multi-tasking scheduler runs inside the *device controller* process and allows to execute several tasks while performing the normal command server work (figure 3).

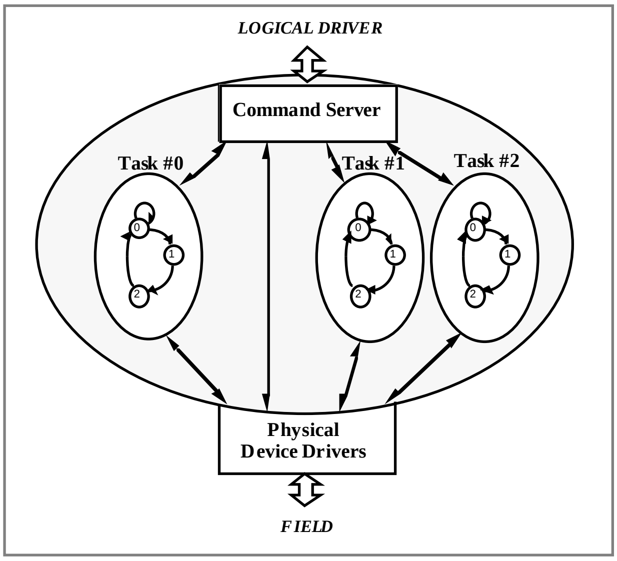

Figure 3: Device Controller structure.

Every single task inside the *device controller* operates as a "state machine" where the actions are written in C code and the transitions between states are defined by macros in order to ease the programmer's work.

Each *device controller* is written to manage a class of devices, e.g. the power supplies, but doesn't contain information about each specific device. This is supplied at run-time by a file of device dependent properties like hardware addresses, calibration parameters, maximum and minimum values, etc.

The *service request manager*, the *poller* and the *logical driver* are standardized modules; the only part to customize when developing a new control system is the *device controller* and the *physical drivers* for the specific equipment to control.

The configuration of the EC software is very simple and relies on three types of ASCII files: the startup file which loads the modules and launches the processes, the LDB configuration file that specifies the controller-poller association for each point in the data base and the refresh time, and the properties files of the device dependent parameters.

This new framework relies on the OS-9 operating system and makes use of standard real-time programming tools such as signals, alarms, semaphores, etc. The development of the back-bone of the system has just been completed and the first device controllers have been written.

5 THE ALARM SYSTEM

An EC *alarm daemon* can send warning messages to an alarm processor on a workstation when a particular event occurs on the continuously monitored variables. In order not to load the *device controllers* with additional accesses to the field, the *alarm daemon* takes the data from the LDB.

The alarm conditions are described by formulae in which the operands can be both constants or field variables (digital and analog) and the operators can be all the relational, logical and bitwise C operators. The formulae are defined by means of a context-free grammar, especially developed for this purpose, and supplied as strings in an ASCII configuration file. When the alarm system is started on the EC, a parser (developed with the LEX and YACC tools [5]) extracts the formulae together with other configuration parameters, like the monitoring repetition frequencies, from the ASCII file. The *alarm daemon* periodically reads the monitored variables, parses the formulae, evaluates the alarm conditions and, in case, sends some alarm messages to a workstation.

6 CONCLUSIONS

The project is in the design stage. The main difficulty for the substitution of the control system comes from the tight operation schedule of ELETTRA, which does not allow long shut-down periods.

To overcome this problem it was decided to start with the installation of the new control system on one of the linac plants (one of the accelerating sections) and disconnect it from the rest of the machine. Such a solution allows to continue the operation of the linac even if at the price of a lower injection energy. As soon as the installation on all the accelerating sections will be accomplished the pre-injector control system, requiring a longer shut-down, will be replaced.

REFERENCES

[1] G. D'Auria et al., "The ELETTRA Injector Linac Upgrading Program", this conference.

[2] C. Bourat et al., "The 100 MeV Preinjector for the Trieste Synchrotron", IEEE Particle Accelerator Conference Proceedings, Chicago, USA, March 1989.

[3] D. Tronc et al., "The Elettra 1.5 GeV Electron Injector", IEEE Particle Accelerator Conference Proceedings, San Francisco, USA, May 1991.

[4] D. Bulfone et al., "Controls in the Past Year of ELETTRA Operation", Proc. Int. Conf. on Accelerator and Large Experimental Physics Control Systems, Chicago, Illinois, 1995, to be published.

[5] T. Mason and D. Brown, "Lex and Yacc", Ed. Dale Dougherty, O'Reilly & Associates, Inc.