LNLS CONTROL SYSTEM

J. G. Franco, R. H. A. Farias, L. Jahnel, J. R. Piton, P.F. Tavares, LNLS, Campinas, Brazil

Abstract

The LNLS synchrotron radiation source control system is presented. The system uses homemade, inexpensive and simple low-level hardware based on 8-bit microprocessors and has been used both for the storage ring and beam line controls. The system architecture allows easy design of customised general small applications. This system has been successfully used even in industrial plants with very low failure rate. A review of the performance of the storage ring control system indicates the critical points to be modified for the project of a future booster synchrotron, such as improving the ramping time and data transfer rate.

1 INTRODUCTION

LNLS is a materials science research facility built around a synchrotron radiation source based on a 1.37 GeV electron storage ring. In this paper we describe the low and high level software and hardware comprising the machine control system, critical points found, beam line controls and customised small applications. The system has been in operation since December 1995, when the commissioning of the injector started with the preliminary version of the high level software. System performance and reliability have improved steadily since then and have now reached the design specifications, demonstrating that the initial conceptual design was basically sound.

2 LOW-LEVEL HARDWARE AND SOFTWARE

The Control System is divided into three levels. The equipment level is the lowest one, in direct contact with the equipments being controlled. It is formed by homemade *Local Controllers* (LOCO) based on the Z80 microprocessor. A LOCO crate is composed of a serial and a CPU card in addition to control cards attached directly to the equipments. The intermediate level, called distribution level, is formed by a *Concentrator*, which is composed of several serial cards. Each of these serial cards is connected to a number of LOCOs forming a control system network segment. Each LOCO is called a node of the control system network. All data coming from the local controllers are condensed in a Dual-Port Memory device (DPM) in each serial card. The maximum number of nodes in each segment is 13 and the maximum number of control cards in each node is 18. Finally, a low-cost PC connected to the concentrator represents the User Interface Level. The communication between the concentrator and the PC is carried out via a PCLOCO card installed in one of the ISA bus slots of the PC and is directly connected to the DPM of the several nets connected to the concentrator rack.

2.1 Local Controller

Local controllers manage the interface between the Control System and the equipments in the facility. The control interfaces in their lowest level —at the equipment level— have also been developed at LNLS, using TTL digital signals and analog signals in the ranges of 0V to +10V and -10V to +10V, with 12- or 16-bit resolution. The software for the local controller is written in assembly language and it is standard for all controllers, regardless of the specific equipments controlled. Thus, all controllers run strictly the same software, which makes the general maintenance simpler. This standardisation required the implementation of algorithms that allow the Local Controller's CPU to automatically recognise the cards being controlled and to follow the appropriate management procedures. Distributed databases are thereupon not necessary. The card recognition procedure is carried out during the boot operation by means of CPU calls to pre-established input addresses, which return an indication of whether or not a card exists as well as its type. The CPU then allocates the memory necessary for the card and calls the specific management routines. The allocated memory is mirrored in the serial communication card responsible for the final data exchange with the higher levels of the control system. Once the boot is completed, the CPU starts the data transmission from the equipment control cards to the serial card and periodically checks the serial card for the existence of fresh data to be sent to the equipments. It must be remarked that the local controllers do not have any information about the equipments that are being controlled but only about the cards that are being used. This increases the flexibility as regards modifications that may be eventually necessary in the control system, since it requires only updating a database in the control computer. On the other hand, the accidental insertion or removal of a card from a Local Controller would cause data transfer to the higher-level control software to fail. Algorithms and diagnostic programs are used to detect and correct this fault.

2.2 Communications

A 2 Mbps serial card is responsible for autonomously transmitting data between the Local Controllers (equipment level) and the Concentrators (distribution level). The CPU processing time is then employed exclusively in the control of the equipments. Data transfers within the Local Controller use a Dual-Port memory device. The serial card collects the data to be sent to the higher levels of the control system and makes those received from the higher levels available to the CPU. Data are periodically collected by the concentrator, which requests data from each of the nodes in the control system network individually, thus preventing collision events in the transfer process. In the concentrator, data from each node are stored at pre-established addresses of a DPM and can then be accessed by the PCLOCO card. From the high level software point of view acting upon an equipment means reading or writing a byte at a specific I/O position of the PC port.

2.3 Spy Crate

To minimise the computation power necessary for monitoring some parameters at the user interface level, a spy crate was introduced in the system. For each serial card in the concentrator, there is another one in the spy crate, with a special software module, listening to all transactions in the serial lines. Similar to the concentrator, this data is stored in a DPM and another PC can access the data, sharing some tasks with the main PC. In order to ensure the integrity of the serial communication, the serial cards in the spy crate are not allowed to place any kind of messages in the serial lines.

3 HIGH-LEVEL SOFTWARE

The high-level software layer implements the user interface and machine physics algorithms, and manages static and dynamic machine parameter databases. The user interface is implemented on Intel PCs running Windows95® with the Pascal-like object-oriented visual-design environment DELPHI®. The choice of the development environment was dictated by time constraints regarding both performance requirements and programming ease, allied to the user-friendly interface provided by Windows applications. The user interface comprises a series of modules (Pascal units defining a window or *Form*) which allow the operator to access any equipment in the facility. All these modules share a common dynamic database (implemented as two vectors of bytes in memory - one for the values sent to the local controllers and the other for values read from the local controllers), which contains all information defining the status of the machine. This dynamic database is created, initialised and maintained by an independent application (the so called SERVER) which is permanently running in the background and which communicates with the various user-interface modules via DDE (Dynamic Data Exchange) links. Thus, several independent interface modules (even modules belonging to different applications and running in different machines) can access the same dynamic database via the server. It is the server’s task to interact with the lower levels of the control system software and hardware. It permanently reads the status of all equipments in the system and answers to requests from its clients to send new settings to the local controllers or to prepare special communication protocols (such as in preparing to dump a ramping curve before ramping the machine to high energy). The DDE Server is a fairly short application written with performance in mind and contains no knowledge of the inner workings of the various equipments connected to the local controllers. In contrast, the client modules must access various static databases describing detailed properties of the controlled equipments such as conversion constants, calibration curves, etc. The strategy of dividing the pure user-interface tasks from the actual communications and control tasks into two independent concurrent applications proved very useful because it allowed the many client modules to be divided into various applications. At present, there are over 20 client applications. The three main applications (LINAC, Transport Line, and Storage Ring) contain most of the control windows for individual equipments in the three subsystems. The remaining applications perform the following tasks: (a) continuously monitor the interlock system, (b) standardise (cycle) the magnets, (c) preparation and management of the ramping process, (d) routine checking of all control system network nodes and equipment prior to machine operation, (e) on line detection of equipments faults, orbit distortion and beam loss, (f) permanent register and display of all analog machine readings, (g) management (saving, loading, printing) of machine configurations, (h) control of the RF Storage Ring System, (i) orbit correction and correction matrix measurement, (j) machine status displays on beam line monitors, (k) machine supervision with automatic operator call (via a commercial pager system) in the case of machine faults; this module allows the machine to be run for user operation without the presence of an operator and (l) automatic machine shutdown at the end of a shift. Some client modules act on groups of equipments performing simultaneous coupled changes in various machine settings, according to predefined beam dynamics procedures. It is possible, for example, to use a pair of correctors to adjust independently position and angle of the beam at the injection point. Pairs of quadrupoles in the transport line can also be adjusted in order to match empirically the vertical dispersion function of the line. All client modules make heavy use of static databases, which are implemented in the form of Paradox® 5.0 tables, directly accessible from DELPHI® programs. Although this implies a certain performance penalty since table searches must be performed at run-time to set up equipment parameters for the interface modules, the ability to edit these tables on-line (without the need to recompile the code) has proved to be extremely useful.

3.1 Beam Optics Modules

Standard beam optics simulation tools such as the machine optics design code MAD as well as in-house developed orbit corrections codes are installed in a remote UNIX workstation. These beam optics applications can be accessed from the control room PC's via front-end modules developed in DELPHI®. The main capabilities of these front-end modules are: the direct interface with the main modules of the control system, which allows on-line acquisition of machine parameters to be fed to simulation programs; the user-friendly interface which eases the manipulations of input files; post-processing tools which permit fast analysis of the outputs as well as feeding new machine settings to the control system. The connection between the PC-based codes and the workstation is accomplished via a LAN running the TCP-IP protocol. Orbit correction can also be accomplish locally (i.e. without having to access the remote UNIX workstation for the correction calculation) by means of fixed corrector matrices with assume the standard operation mode. Local correction is faster and allowed the implementation of an automatic correction scheme applied every 20 seconds. Each correction is done in 100 steps in order to reduce perturbations to the orbit during the correction.

3.2 Multi-user Operation

Although only one PC (the *local* computer) is directly connected to the lower levels of the system (through the control network concentrator), machine parameters can be adjusted and read by any other PC connected to the Lab's Local Area Network. The communication between these *remote* clients and the main operation computer is done using TCP-IP protocol controls native to the DELPHI® environment. All transactions take place between the DDE server module running in the main (local) computer and DDE Server modules running in the remote computers. The Server modules running in the local as well as in the remote computers are identical (only configuration parameters are different). The client modules running in the local and remote computers are also identical, such that, as far as the client modules are concerned, there is no difference between remote and local operation. This strategy made the migration from the original single user system to the present multi-user environment easy since no client modules had to be rewritten. Multi-user operation has proven extremely useful during maintenance shifts, allowing distributed access to various machine subsystems.

4 BEAM LINE CONTROLS AND INDUSTRIAL PLANTS

As shown above, the architecture of the LNLS control system allows the control of many physically distributed equipments, as in the LNLS synchrotron light source. However, it can also be easily configured and customised to control localised small systems, such as the automation of a beam line, experimental setups in a development laboratory or even dedicated systems in industrial plants. In this case, the main control PC connects to a crate containing the control cards, rather than to a concentrator crate containing serial cards that connect to the local controllers. For each control interface a DELPHI® component was created and the users can easily construct their own applications. The common software used on most beam line stations, 3-Windcm, was developed in DELPHI® too. It controls interfaces to perform specific operations such as energy selection, energy scanning, motor positioning and device monitoring. It allows a range of customisable options adapted to the particular hardware configuration of each beam line. At start, the program checks if every previously defined card is present and working properly. In addition to the basic functions, the user can add his own routines to specific points of the scanning algorithms, or add any independent routine. These routines are called "scripts" since they can access public objects from the main software and use its macro functions, but they also are written in DELPHI®. They are compiled into independent DLL files (dynamic libraries) which are rapidly selected and changed at run time. The LNLS applied similar systems to control metallurgical processes in an industrial environment. These systems have been running for 6 years with little maintenance. The user reported a few failures normally associated with the controlled equipment and hard industrial environment such as dust, oil, soot and high temperature.

5 SYSTEM UPGRADES

For the booster control system project, it was necessary to verify and detect the critical points of the storage ring system, since the basic system will be the same, but the response time of the booster is more critical. The first point detected is the CPU speed, that will be increased to 8 MHz or, if necessary, 16 MHz. This speed that may seem slow by present standards, is good enough in our system to control the booster ramping time (most critical operation), due to the very specific software and hardware. This modification is simple and the same hardware can be used. Another critical point detected is the serial communication. At this moment the serial card can work up to 2 Mbps, but a new design is under development and will work up to 8 Mbps. This implementation is not very easy because modifications will be necessary at the physical layer and communications protocol. In this design we are moving from the Z-80 to 87C520, 8051 micro-controller family with special implementations from Dallas Semiconductor. The booster control system will use only cards with 16-bit resolution converters. These cards are under tests in the storage ring and are showing good performance.

6 CONCLUSIONS

The storage ring control system has been in operation for approximately 4 years. This system has good performance, low maintenance (better than 0.15% of the operation time this year and mostly associated with the crate power supplies) and good modularity. It is relatively cheap. A crate to control a power supply with 16-bit resolution and the implementations to control the booster can be assembled for about US$ 700.00. The present configuration of the control system includes a 200 MHz Pentium II PC, with 128 MB of RAM and 10 GB of hard-disk, a concentrator connected to 9 networks making up approximately 350 control boards and 3400 control points. There are no critical problems in the present system and it can be adapted easily to the booster.