

## REMOTE CONTROL OF THE LEP POWER CONVERTERS

J.G. Pett

CERN, SL Division, 1211 Geneva 23, Switzerland

### Abstract

A major break with traditional methods of remote control for power converters was necessary for the LEP accelerator. An overview of the complete system, emphasising the important design objectives and overall strategy, is given. Being based on a minimum set of programmable modules, the vital role of the real time software and its development are discussed. The flexibility and cost effectiveness of the chosen methods are reviewed in the light of the initial months of LEP operation.

### A New Approach to Remote Control

The methods of interconnecting equipment prior to the start of the LEP project were based on the available technology (principally CAMAC) and on the fact that accelerators were rather compact machines. Experience gained over many years with operating machines such as the ISR, led rather naturally to a 'wish' list of improvements on the then current methods.

The LEP Project therefore needed a new method of approach which provided answers to both the vastly increased size of the site and solutions to the already known problems.

Because of the vast site a totally new method of serial communication was necessary in order to keep cabling costs to a minimum. Additionally, greatly improved diagnostic methods had to be foreseen in order to keep the mean time to repair within acceptable limits. This also dictated that a high degree of uniformity, both in method and hardware realisation should be aimed for, such that support activities, spares etc could be optimised.

Evolution of accelerators has always occurred, frequently requiring extensive changes or modifications to power converter systems. LEP was designed from the start in two major construction phases, thus requiring a remote control method which facilitated such evolution. It was also considered mandatory to choose a method which enabled a clear division of responsibilities to be established between the Controls and Power Converter groups, not only for the construction phases but also for the maintenance and support activities.

### Overview

The solutions chosen for the control system in general have been described extensively in the past (see Ref. 1). Briefly, they consist of control room work stations connected to an extensive network based on time-division multiplexing (TDM) and the IBM token ring. The interconnections to the power converter equipment uses the MIL 1553 and a specially developed serial timing system. Both these serial communications mechanisms

are connected in a single multi-drop cable to each power converter. This latter point helped to define a clear separation of responsibilities between the controls and power converter groups. Messages, consisting of a number of fields, are exchanged between the power converters and an "intelligent gateway" computer (termed a Process Control Assembly or PCA) positioned in the same areas as the converters. Apart from the header fields, containing 1553 communications details, there are three fields of interest for power converter control. They are the Family-member, Action and Data fields, where the first provides a virtual address, the second indicates the function to perform and the last provides any associated data.

In total, 780 power converters are controlled at the moment but this number will increase by some 10% during the second phase of LEP construction. Additionally, a considerable number of converters will have to be rearranged and connected to different loads.

### Basic Design Features

Probably the biggest departure from earlier methods was the idea of using embedded microprocessors (8 bit, 6809 types) in each power converter (Ref. 2). This stemmed from the need to provide an extremely uniform user interface as well as a high degree of parallel processing to counteract the use of "slower" serial communication methods. This led naturally to a close examination of the possibilities offered by the use of such "local intelligence".

In order to realise a minimum overall cost, a decision was taken to use an identical remote control hardware for all converters. Such a choice allows considerable production economies to be made because of the high manufacturing volume, but also gives additional benefits namely :

- Higher reliability due to the use of volume production testing methods.
- Much easier installation and subsequent maintenance.

A consequence of the above choices was that all individual differences between the converter types (some 27 in all) and their physical realisations had to be taken care of by the microprocessor software, programmable logic or option straps. A somewhat unexpected by-product of these decisions has been the ease with which last minute changes and improvements could be incorporated.

For each function which needed to be remote-controlled, a direct interface to the microprocessor bus was designed, based on a commercial standard bus. This enabled the entire functionality to be controlled by software placed in EPROMS. A careful choice was made between doing absolutely everything in software and providing a limited amount of additional hardware. Such

a choice was arrived at by combining the development of the base level software with that of the individual hardware modules. All vital protection mechanisms for the power part of the converter were none the less executed in dedicated hardware.

### The Realtime Software Design

The design of the real-time software proved to be a tedious and time-consuming activity, since the power converter environment was largely unknown and more "hostile" in some cases than expected. After the development of the module test software, as outlined above, a serious study of the entire methodology of power converter control was undertaken. This helped enormously in defining a logical and uniform approach. A top-down design method helped to define the major software blocks and the data flows etc.

An outline diagram of the software blocks is shown in Fig. 1. This diagram shows the major data flows and identifies the principal tasks. Messages are either received via the CR1553 block, itself consisting of three interdependent tasks, or via a local terminal input mechanism. The message is then parsed by the CHK task which also ensures that the requested action is known, is accompanied by the required data, that the action can be executed in the present state of the converter and that the data is within limits.

If the action requires a change of state of the

converter or of any associated data then a parametrised call is made to the PUT block. If, on the other hand, data values are requested, then the GET block is called. This block passes the data to the DIR (direct output to device) block which ensures that the reply message is routed to the originating device.

Any errors or failures, such as alarms, are processed by the ERR block which also calls the DIR block. Many failure mechanisms as well as the entire internal timing functions are coordinated by the TIMER block.

Associated with the PUT, GET, DIR and CR1553 blocks are a number of driver routines, a special example being the ADC task shown in Fig 1. Most of these driver routines are activated either by direct calls from the above mentioned blocks or by interrupts generated by converter hardware.

It was clear that an interrupt driven mechanism would be needed to cope with the large number of input/output (I/O) devices (more than 20) and that a real-time executive would be needed to correctly schedule the treatment of these interrupts.

One of these I/O devices, namely the reference function generator, had to operate independently from the rest of the equipment and is therefore not shown in Fig. 1. This complex device, which was based on a 16 bit digital to analog converter (DAC) driven by a dedicated microprocessor, had to generate the analog reference signal to the power converter in precise synchronism with

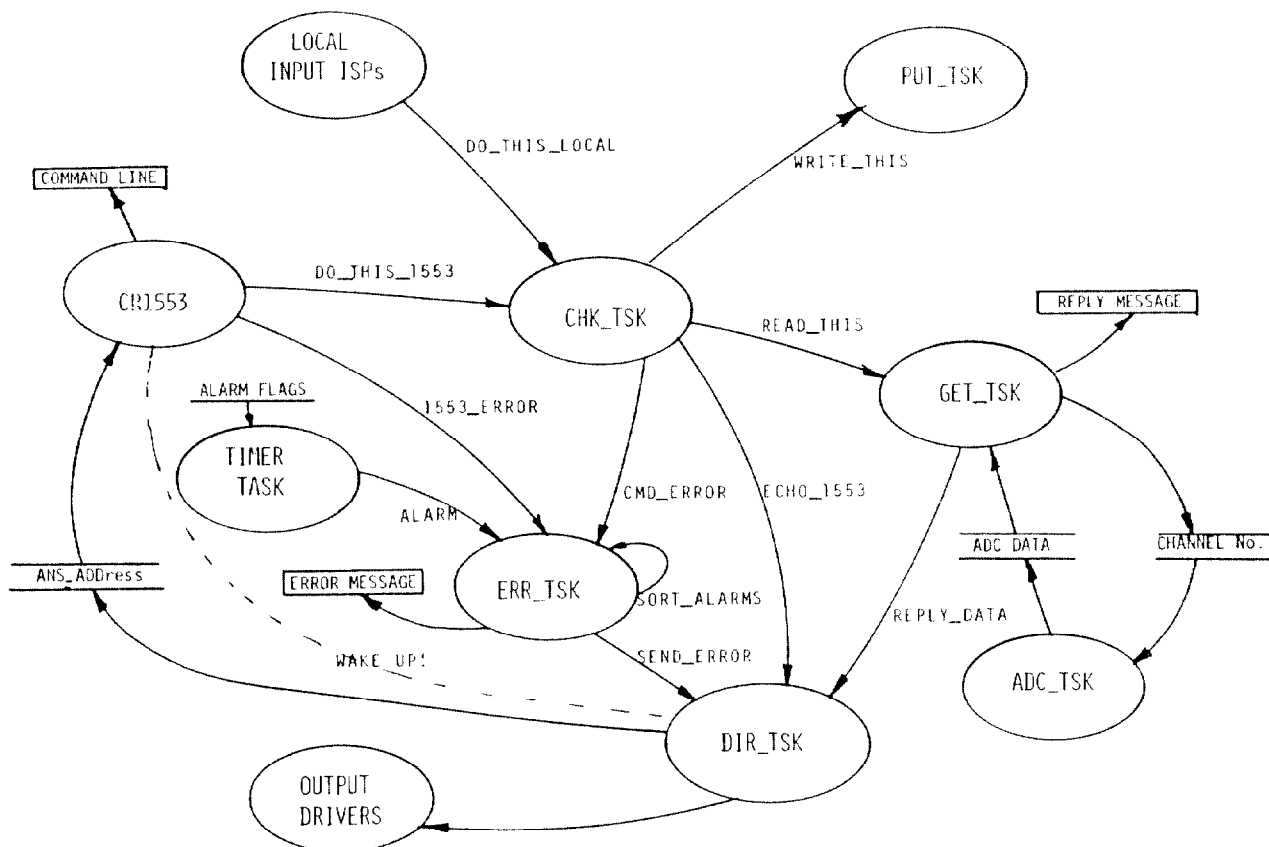


Fig. 1 Outline Diagram of the Software Blocks

the machine timing signals. The solution adopted was to generate the required function by a succession of linear segments of equal length (256 ms.) by using a binary integration method. Data transfer and control words were exchanged with the main microprocessor via an area of common memory. An eight-channel transient recorder, needed for diagnostic purposes, was also built in the same module. These two devices were entirely controlled by the on-board microprocessor. A considerable number of operating modes for the function generator were programmed on-board, in particular those for the quasi-exponential start and stop of the output function and compensation for the eddy-current effects in the LEP vacuum chamber.

By analysing the different physical and operational requirements of the different power converters it was possible to design more than 95% of common software modules. This was achieved by using a memory area for each converter type which contained specific data, error messages etc. This data was placed in a separate, electrically-erasable memory device (EEPROM), thus facilitating upgrading and use of common software. Typically such data consists of maximum and minimum output currents, slew rate, DAC type, power converter name as well as its manufacturing code and software version number.

#### Development

The software was developed using commercial packages, many of which are by now considered to be rather primitive. The majority of high-level functions were written in Pascal but 6809 Assembler was used extensively for the drivers and reference function generator. The real-time executive was specially adapted to facilitate calls from both Pascal as well as Assembler. The executive provided most of the important features needed for our purposes and since it did not use a time-slicing method ensured an orderly completion of tasks.

More than 60 kilo-bytes of code and 32 kilo-bytes of random access memory are used in each converter. The total development time, from initial hardware test software to the final installed version needed some 13 man-years of effort. A large part of this effort was spent in debugging the software in the environment of a real power converter. The use of a sophisticated logic analyser was indispensable for this work, since it enabled both waveform analysis as well as disassembly of code to be performed simultaneously. While the use of a number of power converter simulators proved to be essential during initial development, it was vital to test everything using the real converters. A number of bugs were only discovered during final testing of the complete accelerator, in spite of what seemed to be exhaustive proving tests.

Among the most difficult problems to locate and cure were those caused by radio-frequency interference (RFI), error handling, communication protocol violations, rounding errors, recursive calls and side effects due to minor program updates. Major problems were caused in the initial months of operation due to early and intermittent failures of PAL's and EPROM's. These

problems "crashed" the program only when certain parts of the code were exercised.

Feedback from users has been most useful in improving the software, in spite of the added task of "educating" them in the use of a new technology. As we are currently at Version 4 of the software, we have suffered from the usual problems of keeping track of patches, improvements and keeping the documentation up to date.

#### Conclusions

Initially, we had in mind to implement some twenty different actions (with parameters) whereas in practice there are now over seventy, some of which are compound actions. This "power converter command language" provides operators and maintenance staff with a uniform and user friendly interface.

Our experience to date has been very positive during the initial months of operation. Early failures were already low but have decreased to today's very low figure. Due to the large emphasis placed on diagnostics (both for hardware as well as software) the repair time has been kept to a minimum. No overall problems, due to the use of the new methods, have become apparent.

Flexibility has proved to be more than adequate and a number of important upgrades to specific converters have been implemented during operation. Changes are not that easy to make on a global scale because of the large number of EPROMS to exchange. This could be improved at a future date by using a more modern single circuit.

Costs are low for the overall functionality obtained because of the use of universal hardware which is configured by programming methods. Updates are very cheap and rapid to perform, being essentially manpower intensive. Equally, redeployment is greatly facilitated. The ability to upgrade algorithms translate into cost savings for the machine due to the possibility of using the same basic hardware to accomplish the new task but also because they can be done relatively rapidly.

#### References

1. P.G. Innocenti, The LEP Control System : Architecture, Features and Performance, CERN SPS/89-35 (ACC), August 1989.
2. J.G. Pett, Power Supply Control in LEP, CERN LEP Note N°385, 07.06.1982.