

# Laboratori Nazionali di Frascati

LNF-93/039 (P)

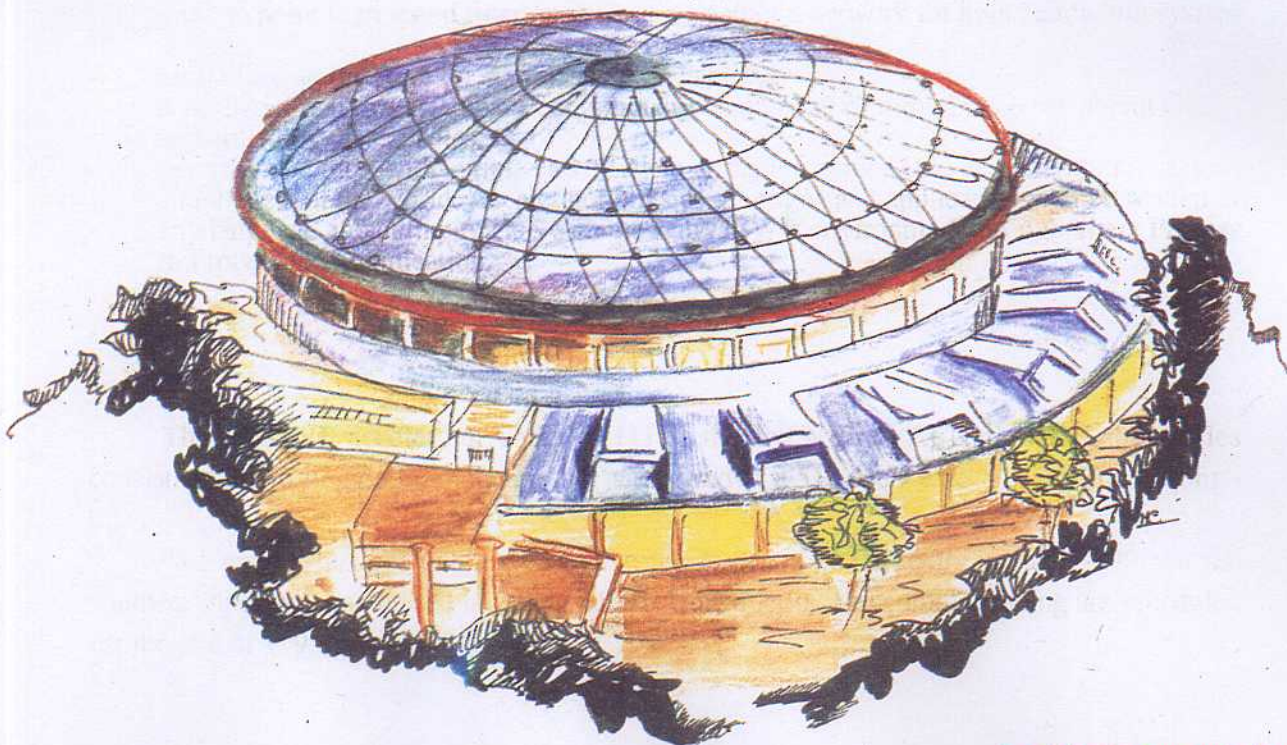
27 Luglio 1993

G. Di Pirro, C. Milardi, A. Stecchi, L. Trasatti:

**DANTE: CONTROL SYSTEM FOR DAΦNE BASED ON PERSONAL COMPUTERS AND HIGH LEVEL TOOLS**

PACS.: 29.50.+v

Presented at the  
RT93, June 8-11th, 1993, Vancouver, Canada



Servizio Documentazione  
dei Laboratori Nazionali di Frascati  
P.O. Box, 13 - 00044 Frascati (Italy)

**DANTE: CONTROL SYSTEM FOR DAΦNE BASED ON PERSONAL  
COMPUTERS AND HIGH LEVEL TOOLS**

G. Di Pirro, C. Milardi, A. Stecchi, L. Trasatti

INFN, Laboratori Nazionali di Frascati, P. O. Box 13, 00044 Frascati (Rm) ITALY

***Abstract***

The DANTE (DAΦNE New Tools Environment) control system has been fully designed and is now being integrated. Several innovative features have been inserted in the design:

- the use of Macintosh Personal Computers throughout the whole system, including VME CPU's built on purpose using Apple CPU boards gives access to an enormous quantity of high level software, where new powerful tools become available day by day. LabVIEW<sup>®</sup> by National Instruments has been chosen as development environment.
- point to point high speed fiber optic links instead of a network for high bandwidth system update;
- centralized architecture for simplicity and reliability;
- centralized real time Database continuously updated at runtime by the peripheral CPU's and polled by the consoles;
- open architecture, with simple and efficient communications between internal processors and the rest of the world. Accelerator physics programs and applications can be written in any language and communicate with the LabVIEW<sup>®</sup> environment through direct Process to Process Communication.

**I. DAΦNE**

The DAΦNE accelerator complex [1] of the INFN Frascati National Laboratories consists of a two ring colliding beam Φ-Factory and of a 510 MeV e<sup>+</sup>/e<sup>-</sup> injector for topping-up.

The project has been approved by the INFN Board of Directors in June 1990 and the engineering design has started in January 1991. Construction and commissioning are scheduled for the end of 1995.

The complex consists of (see Fig. 1):

- a commercial LINAC, with built in control system;
- an accumulator damping ring for injection and topping up;
- a system of transfer lines;
- the two major rings.

Response times required by the different parts of the complex vary widely: the refresh time for the transfer lines can be of the order of a few seconds, while the main rings will require updates at a frequency of at least 10 Hz.

From the point of view of the peripheral hardware, the accelerator complex consists of about 1000 devices, which will be controlled by 5 consoles and about 50 peripheral CPUs.

## II. SYSTEM STRUCTURE

Fig. 2 shows the general architecture of the control system. Three levels are defined:

PARADISE (PARAllel DISPlay Environment) is the operator interface level. Several consoles, built on Macintosh personal computers, communicate with the rest of the system through high speed DMA buses and fiber optic links.

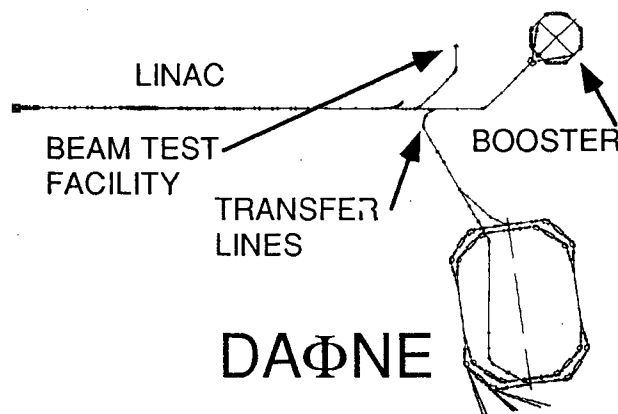


FIG. 1— DAΦNE accelerator complex

PURGATORY (Primary Unit for Readout and GATing Of Real time Yonder) is the second and central level of the system. It contains a message dispatcher (CARON) that forwards commands from the consoles to the appropriate peripheral CPUs and a set of READERS that poll the peripherals for meaningful events and update through VSB a central memory which contains the machine status and represents the prototype of the machine database. This memory can be accessed through VME from the consoles. This arrangement makes the whole system asynchronous: the consoles do not need to interrogate the periphery to find out the status of a device, but find it updated automatically in a fast memory. The VME–VSB arrangement insures that no bottlenecks are generated by very loquacious devices.

HELL (Hardware Environment at Low Level) is the third level of the system and it is constituted by many (about 50) VME crates distributed around the machines. Each crate is equipped with at least one CPU which performs control and readout of the related elements in the machine. Only significant changes in the parameters are transferred to the Purgatory, thus

hiding useless information from the central processor.

Several innovative features have been inserted in the design to improve reliability and ease the debugging phase. We shall describe the most important ones.

All the communications of the system are controlled by the purgatory. Point to point links have replaced the use of a network, to make the system modular and easy to expand.

We shall not use interrupts, and will replace them with polling mechanisms. This may increase slightly the necessary computing power, but it will dramatically reduce the debugging and maintenance of the system. The failure of a peripheral unit will be diagnosed and isolated very efficiently.

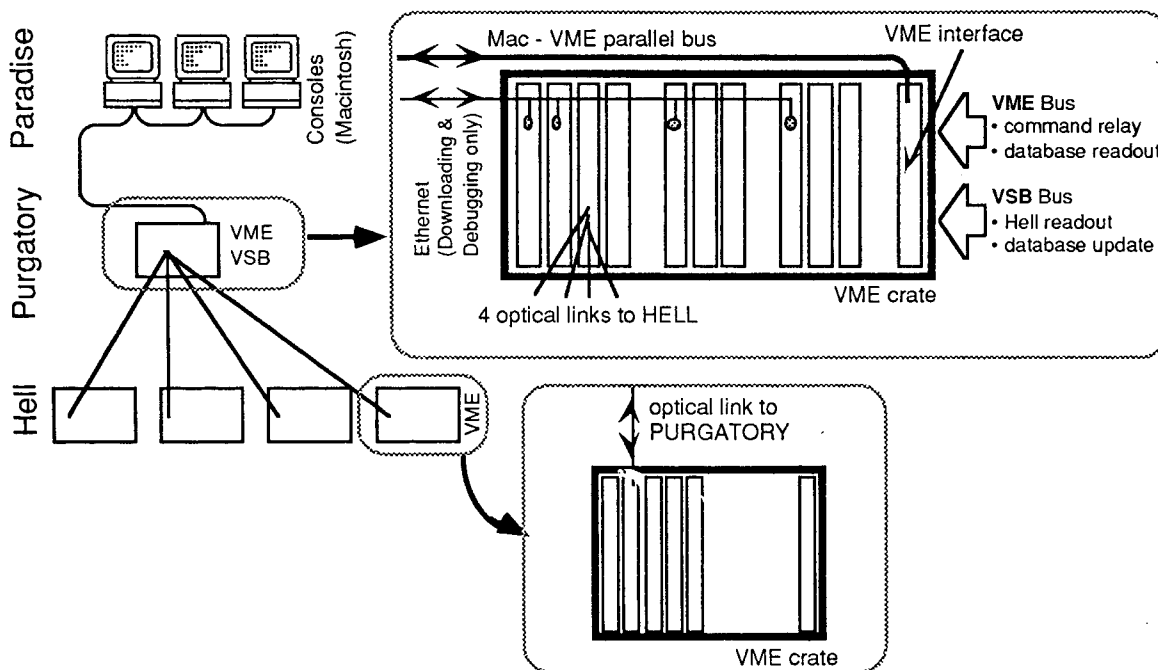


FIG. 2 – Control System Schematic Diagram

### III. SYSTEM SOFTWARE

We believe that software is a much more important problem than hardware. Therefore it was the development environment that decided our hardware components, and not vice versa. We spent a long time testing and evaluating languages and packages: FORTRAN, C, C++, Prototyper™(Smethers Barnes.), Hypercard (Apple Computer Inc), and so on. We finally decided to use LabVIEW®. This package has been on the market for a few years and is now well stabilized. A new, multiplatform version (LabVIEW 3) is in beta testing. It consists of a graphical programming language, plus a huge library of data acquisition and data analysis routines called VIs (Virtual Instruments). Compared to a standard C + Real Time Operating System the reduction in the number of needed manuals is dramatic. The software development time is also reduced by big factors, because of the many "standard" operations that are already provided (I/O, file handling, graphical interface, etc.).

We started by using it for the human interface, and then realized that it would be

extremely suitable also for the peripheral CPUs. We then decided to build a VME CPU based on a Macintosh logic board. The Mac LC III offers a very good price/performance ratio, and we had a commercial firm develop an interface to VME and VSB that also contains a 4 Mbyte triple ported RAM. The final object (DEVIL) fits into a two slot VME module. Commercial prototypes are available.

We now have a single programming environment both for the high level and for the low level machines. Furthermore the choice of a Macintosh computer makes available all the facilities of the Apple System 7 operating system. Using these it is very easy to implement a general Ethernet downloading and maintenance network.

The choice of a "Personal" computer is not usually treated with favor in our environment. It is true that "workstations" are at the moment more powerful in terms of MIPS than a Macintosh Quadra, but the distance is decreasing very rapidly, and the Power PC generation of both IBM and Apple machines will compare very favorably with existing higher priced machines. Furthermore, we believe that in an era of multiprocessor systems, power should be increased when needed by the addition of more CPUs, not of bigger ones. If the farmer's apple cart has become too heavy to be drawn by his ox, he gets a second ox, not a bigger one.

On the other hand, the amount of high quality software available on cheap machines is and will continue to be non comparable with "powerful" mainframes and workstations. After all, the only way to measure software reliability is still to count the number of errors that users detect during the first months of operation, and the number of users is inversely proportional to the cost of the machine.

Software is becoming bigger and bigger. If one considers that the effort involved in developing LabVIEW® is of the order of 100 man years, it is clear that we cannot any more start from scratch, particularly in a research environment, where manpower for software is typically very tight, while computing requirements are always very high.

#### IV. APPLICATION SOFTWARE

##### *Accelerator Physics software*

The control system of a modern accelerator requires a large amount of programs to interface the machine operation to the needs of the accelerator physicists.

They are necessary to optimize the machine working point and to run it efficiently.

The FORTRAN language has been chosen to write these applications due to its large diffusion among physicist. The project team can develop their own applications running through the control system. All the High Level Software programs run through the Control System human interface and permit to operate the machine in a truly interactive way.

As a starting point we are providing the necessary programs for the DaΦne accumulator. This helps to develop basic general tools like data exchange routines, and to test the application performances.

##### *Process to Process Communication*

An important aspect of Macintosh System 7 is Inter Application Communication (IAC), a new collection of features that helps application work together.

The IAC architecture provides the Program to Program Communications (PPC): a set of low-level routines for exchanging blocks of data between applications.

We have developed a set of interfaces (LVLibrary) based on the PPC for accessing the DANTE LabVIEW® environment from a standard standalone FORTRAN application running concurrently.

The advantage of using the PPC mechanism instead of a conventional file read/write is due to the speed that is typically of 4  $\mu$ s/byte.

The facility of a fast and easy link between Language System FORTRAN® and LabVIEW® can be useful for:

- sending data from a control window to an external routine that processes them and sends the results back;
- inserting a control process that can read and write values from and to the Control System.

It is also possible to insert a piece of FORTRAN (or C) code inside LabVIEW® as a subroutine (CIN).

## V. THE HARDWARE

VME has been chosen for both Purgatory and Hell. The ease of implementation of multiprocessing, the large amount of modules commercially available, the low cost make it the obvious choice in this moment.

Communications between levels are implemented through high speed parallel buses where distance and noise considerations allow it. On the other hand, since we must control a complex cluster of machines (LINAC, Booster Ring, Storage rings and transport lines between them), distances are greater than the 70 meters typically allowed by parallel buses. Therefore we are developing a VME module, OPLA (Optical Link Adapter), capable of 125 Mbit/s serial transmission on optical fibers. OPLA is a simple passive point to point optical link acting as a VME slave. It is capable of supporting the maximum throughput of a standard VME CPU. Four Tx/Rx channels are housed in a single width VME module (double width if we include the optical transceivers). The received data are stored in a 2048 word FIFO. We chose this solution, instead of automatic memory update through different crates, because it is simpler and more efficient in a message based system.

## VI. SECONDARY BUSES

Although VME is an obvious choice for a multiprocessor system, several situations occur where a sub-bus is less expensive than a set of direct I/O channels. For example, in the case of magnet power supplies, it is possible to control several of them using a digital I/O, an ADC and a DAC board. On the other hand, it is nowadays very easy to find power supplies that incorporate an RS/232 or IEEE 488 interface. This allows a strong simplification in the cabling system, which is in itself extremely expensive. We have decided to interface the magnet power supplies to our system using these sub-buses, namely RS/232 wherever speed is not important, e.g., the transfer lines magnets, and IEEE 488 where a high update speed is required.

## VII. STATUS

The central communication structure has been implemented and installed on a prototype of the DEVIL using the beta version of LabVIEW 3. We are now in the process of developing the control system interfaces to the separate pieces of equipment, in collaboration with the experts.

## VIII. SUMMARY

The DANTE control system is based on highly distributed hardware and software capabilities, with a strong accent on openness to other environments. Previous experience with smaller systems tells us that these are the most stringent requirements we will have to face, and that the human interface will be the most arduous problem to solve. The use of modern software techniques is a big help in this direction.

## IX. ACKNOWLEDGMENTS

We would like to thank the Accelerator Group of the LNF for continuing discussions and encouragement.

## X. REFERENCES

- [1] G. Vignola et al., talk presented at the San Francisco Particle Accelerator Conference, May 1993.