The Tesla Test Facility Injector Controls

F. Gougnaud, J.F. Gournay, Y. Lussignol
CEA-Saclay DAPNIA/SIG 91191 Gif/Yvette France

1. INTRODUCTION

The goal of the Tesla Test Facility (TTF) project is to demonstrate the feasibility of a large superconducting positron electron collider. The collaboration includes several European and American laboratories. TTF is a linear accelerator with four cryomodules of eight cavities. The aim is to reach 500 Mev with a gradient of 15 MV/m. TTF will be installed at DESY. The Injector has been built by LAL (Orsay) and DAPNIA (Saclay) including its controls. There was an agreement between us and DESY in February 1993 to use EPICS for the injector controls [1]. This paper will present the architecture and the hardware of the system, the EPICS drivers, the EPICS tools used and some specific tools we have developed.

2. INJECTOR OVERVIEW

The TTF Injector comprises an electron gun with a high voltage system, a subharmonic buncher, two beam transport lines with magnetic lenses, triplets, dipoles and steerers. It also includes a superconducting capture cavity and a klystron modulator [2]. Different beam diagnostics are used : beam position monitors, beam pulse intensity monitors, a SEM-grid beam profiler, view screens and an optical transition radiation monitor.

3. HARDWARE

There are 4 dedicated VME crates (IOC in EPICS vocabulary) : one for the gun, one for the beam lines, one for the beam diagnostics and one for the capture cavity and the modulator. The CPUs in the VME crates are Motorola MVME162LX 222 (MC68040, 4MB DRAM). Most of the other VME boards are standard industrial products. These include:

A/D boards ADAS ICV150 (12 or 14 bit)
Fast A/D board OMNIBYTE COMET
D/A boards ADAS ICV712 (12 bit)
D/A boards ADAS ICV701 (16 bit)
Binary Input /Output boards ADAS ICV196
Stepping motor boards ADAS ICV914.

We chose ADAS boards because we have had good experience during the last years with these boards [3]. Furthermore, the manufacturer has a complete set of hardware to interface the VME boards to the equipment (isolated inputs/outputs, relay outputs, motor amplifier boards).

For SEM-grid and beam position monitors, the boards were designed by LAL. A programmable timing system designed by Fermilab, comprising a rep-rate generator and IP timer modules [4], delivers synchronization to the gun, the capture cavity and the diagnostics. Lastly, we use an ESRF-designed VME board for video switching [5].

4. EPICS DRIVERS

We have received and installed EPICS during the summer of 1993. After that, the first important step was to develop a set of drivers for all the VME boards we planned to use. There were no major difficulties in writing them, thanks to the EPICS manuals and to similar drivers which came with EPICS. In addition, discussions and E-mail exchanges with the original EPICS developers made the work easier. Initially, the standard structure consisting of a device and a driver layer was adopted as advised in the documentation [6]. But this structure was not always justified and we only kept it when necessary. Although this was probably obvious to the main collaborating laboratories, we didn't get the information and then wasted some time in reorganizing our drivers accordingly. This shows that even if a world-wide software sharing policy is very constructive, it may be difficult for a small, remote team to stay in phase with the major laboratories involved in the life of the product.

Most of the drivers were easily written except for the stepping motor which involves asynchronous processing. This type of driver should not block the database during the processing. Consequently, the driver puts the commands in a message queue. A readback task reads this queue, processes the commands on the VME board and allows the driver to hand the control back quickly to the database scanning tasks. The readback task updates the motor data in the database using a callback routine mechanism.

The free circulation of the EPICS source code is also a great advantage. For instance, the driver for the COMET board was first written at Los Alamos. It was then extensively modified at DESY and finally we made some minor enhancements to improve the software-event processing.

This gives some concern about the future life of our developments. As it is unlikely these drivers will be incorporated in the standard EPICS we will have to do this task each time we install a new release.

5. USE OF EPICS TOOLS

Once the drivers were written, the displays for the operator interface were designed rapidly. We used MEDM (an EPICS tool that provides a graphical user interface) and KM (a tool assigning the SUNDIALS knobs to variables). Synoptic displays show the whole machine with diagnostics and magnets (Fig. 1). Steerers and lenses can be controlled with the SUNDIALS. The response times are very satisfactory. MEDM has all the general functions necessary for a control system and is used for most of the displays: the RF capture cavity, the diagnostics, the timing... For the RF capture cavity, with its cold tuning system, the display shows the very complex automation of the RF loops (Fig. 2). This was the first application delivered in June 1994. For the beam profiles, a MEDM display plots the raw data, the average smoothed data and shows additional information computed from the data.

SNL provides a simple tool for programming sequential operations based on the state transition diagram concept. It was used for different applications, such as to program the automation of the RF loops and for the calculations of the centre of gravity and the different widths of profiles. SNL was used for the standardization of user code in the IOCs and for simplicity in the database access method.

To build the databases of the different IOCs, the EPICS Database Configuration tool (dct) is used. Dct is a basic character-oriented tool which mainly lacks the ability to represent the links between the records (which could be rather complex) in a nice way. Nevertheless, this tool is adequate because we have split our databases in logical pieces small enough to be handled in this way.

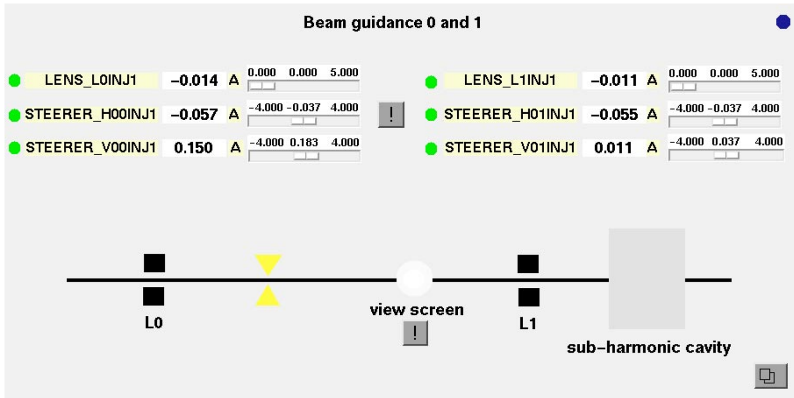

Fig. 1 A Synoptic Display for the Injector

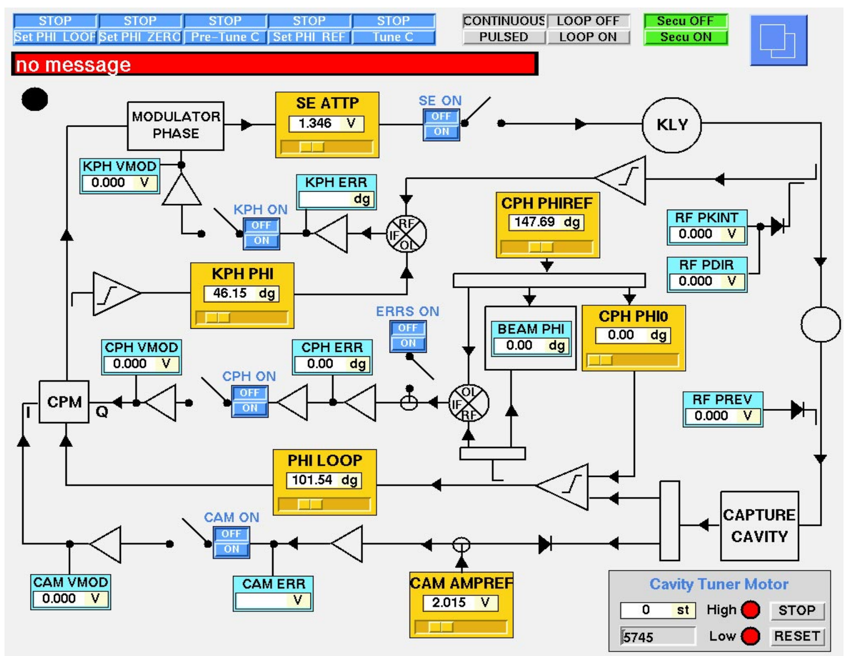

Fig. 2 Superconducting Capture Cavity Control Screen

6. NOMENCLATURE CONVENTIONS

To name parameters, the following convention is used : *device_location_property*. In general, the device name identifies the equipment (steerer, dipole etc.). For complex equipment like the gun or the capture cavity, the device name is, in fact, the name of a subsystem (high voltage for the gun or phase loop for the cavity). The meaning of the location field is obvious. In general, it consists of a number followed by the name of a logical sector of the machine. For complex devices, the location name is the name of these devices (gun or capture cavity). The property field refers to the physical parameter that is controlled like "intensity" or "voltage" or "status"... This property is followed by two lower case characters following the EPICS convention: ai for "analog input", ao for "analog output", bi for "binary input", bo for "analog output". All the characters are upper case except the last two. For instance, we have: LENS_1INJ1_Iai (readback value), LENS_1INJ1_ONbo (on/off control), KPHLOOP_CAPCAV_PHIao (phase control for the capture cavity).

7. SPECIFIC TTF TOOLS

At DAPNIA we have developed some additionnal tools on the IOP side. The goal was to replace some existing tools (parameter page, BURT GUI) by tools which are more convenient for our own use, or to fulfill some specific needs (virtual oscilloscope, journal printing tool, error messages display).

The parameter page program is a tool which allows the user to control accelerator devices (e.g. power supplies). Its specifications are inspired by a similar tool used at Fermilab [7].

The configuration of a page is defined in a simple ASCII file with the names of the devices. In this file, it is also possible to specify pointers to a "next" or "previous" file in order to rapidly switch between pages. Finally, SUNDIALS knobs number can also be specified to automatically assign the knobs to the parameters. Each line displays the information for one device. The notion of device is implemented in a very simple way : all channels belonging to the same device are identified with the DESC field of the database. The parameter page has some additional features :

- push buttons to save a set of parameters and to restore them

- undo push button to return to the value before the last change

- "init" buttons to set the database alarm limits around the actual readback values. The intervals used to compute the limits can be set individually for each parameter

- trend plots for each readback value.

This tool needs to access the databases located in the IOCs. This is done in the EPICS world through the so-called Channel Access facility. A small library was developed which enables this communication with a maximum efficiency : support of asynchronous operations and monitors, cache to avoid repeated connections with the channels, support of complex get functions (retrieval in one call of all information concerning a channel). A rather similar library was provided by APS but it came after our own development.

This tool is based on OnX, a GUI builder developed at LAL [8]. OnX is easy to use for developing complex user interfaces, but it has limited performance for updating such screens with dynamic numerical values and trend plots. Nevertheless, the extended functionalities and the nice look and feel (Fig. 3) will surely make this tool very useful for the control of the machine.

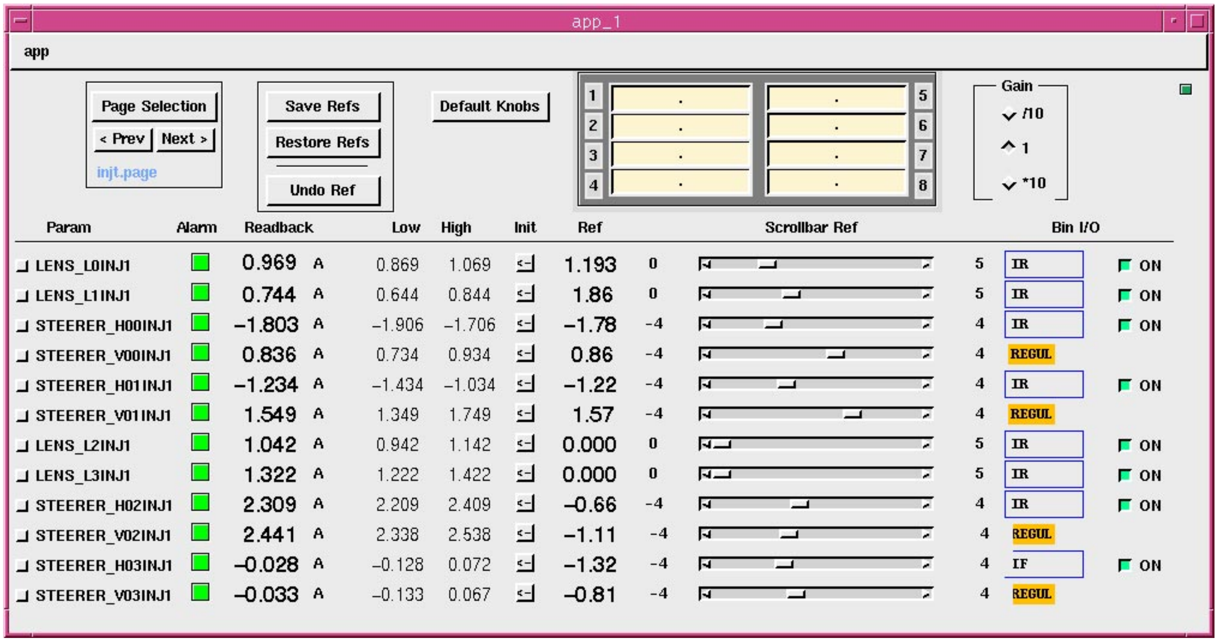

Fig. 3 Parameter Page Display

Some other tools were also developed :

A virtual oscilloscope used to display simultaneously the four channels of the COMET board. It has the basic functions which exist on a real oscilloscope : continuous sampling or one-shot mode, vertical and horizontal scale setting. For the injector, the COMET board is used for the beam pulse intensity measurements. This tool uses our in-house developed channel access library and is also based on OnX.

A journal printing tool : such a journal is composed of basic keywords and parameters known from the database. These keywords are used to define formatting instructions. When a request is made for printing a journal, the parameters are read from the IOCs, the journal is composed and sent to troff for formatting. The printout is used for the log book of the machine. The user interface is based on Tcl/Tk.

A front-end GUI for the backup and restore facility (BURT). This facility is used to save a set of parameters in a file and to restore this setup when needed. This program is extremely important during the startup and the commissioning phases. One such user interface already exists but it does not completely correspond to our requirements. In particular, it lacks the most common selection procedure we use : selection by the comments that are attached to the setups rather than selection by filenames. Furthermore, our interface has the minimal functionalities required, hence it is very simple to use.

A tool which displays the error messages of the system. EPICS provides a UNIX system-wide error server. All the incoming error messages go into a file but no tool is available to display these errors in real-time. Our tool shows the incoming error messages in a scrolled list. It is possible to enter a given string of characters in order to select messages containing this string (useful to check a given IOC for example). This tool was made with Tcl/Tk.

8. CONCLUSION

At the end of 1995, the TTF Injector control system runs for most of the components of the machine. The performance and reliability are very satisfactory. This system was developed with limited manpower, approximatively 3 man-years. The use of EPICS is indeed largely responsible for this good result. Our experience in developing control systems has convinced us for a long time of the benefit of standardization (the use of the graphical environment SL-GMS for several projects) and software sharing (the use of a database inspired from the SLAC SLC database for 3 projects). However, with EPICS, one goes a big step further with the power of the database, the clean interface between this database and the clients through Channel Access and the large set of powerful tools available.

ACKNOWLEDGEMENTS

We acknowledge the work done by M. Bernard, M. Mencik and M. Taurigna for the gun and the diagnostics.We would like also to thank all the members of the EPICS community who have helped us in some way to design our system and especially B. Dalesio from LANL and M. Kraimer from ANL.

REFERENCES

[1] M. Clausen. Controls for the Tesla test facility status and future plans (ICALEPCS'93, Berlin, October 1993).
[2] D.A. Edwards (Editor) TESLA TEST FACILITY LINAC Design Report (DESY, March 1995).
[3] J.F. Gournay et al. Le Système de Contrôle du Projet MACSE (Windays, Paris, April 1991).
[4] M. Shea et al. IP 8-Channel Timer Module (Fermilab, September 1994).
[5] J. Cerrai. Notice Multiplexeur Vidéo MUXV (ESRF, Grenoble, October 1990).
[6] M. R. Kraimer. EPICS Input/Output Controller. Application Developer's Guide.
[7] J. G. Smedinghoff. The Fermilab Accelerator Control System Parameter Program and Plotting Facilities (2nd International Workshop on Accelerator Control Systems, Los Alamos, October 1985).
[8] G. Barrand. OnX : a tool to build interactive graphical applications (LAL Orsay).