# THE SPARC CONTROL SYSTEM

F. Anelli, M. Bellaveglia, E. Chiadroni, L. Cultrera, G. Di Pirro, D. Filippetto, S.Fioravanti, E.Pace, INFN LNF, Italy
A. Cianchi, INFN Roma 2, Italy
V. Surrenti, M. Quattromini ENEA- Frascati, Italy

*Abstract*

We describe the control system operation for the ne injector project built at the Laboratori Nazionali di Frascati INFN (SPARC). The injector started the operation in the autumn of the 2007 and the control systems has been full operating since the start of commissioning and integrate all tools to help the whole machine operation from the gun until the undulator. The SPARC control system must follow all evolution in the machine installation. To allow us a rapid develop of the control system we have made some commercial choices: Labview as developing system; Gigabit Ethernet as interconnection bus with a simple TCP/IP protocol and mainly standard PC as front-end CPU and console. We developed control applications for all machine elements and diagnostic tools. We also developed some tools to help the operation such as an electronic logbook full integrated in the console windows and an automatic process to store all information.

## SPARC

The SPARC (Sorgente Pulsata e Amplificata di Radiazione Coerente, Self-Amplified Pulsed Coherent Radiation Source) (fig.1) project is to promote an R&D activity oriented to the development of a high brightness photo injector to drive SASE-FEL experiments at 500 nm and higher harmonics generation. Proposed by the research institutions ENEA, INFN, CNR with collaboration of Universita` di Roma Tor Vergata and INFM-ST, it has been funded in 2003 by the Italian Government. The machine is installed at Laboratori Nazionali di Frascati (LNF-INFN). It is composed by an RF gun driven by a Ti:Sa laser to produce 10-ps flat top pulses on the photocathode, injecting into three SLAC accelerating and 6 undulator sections.

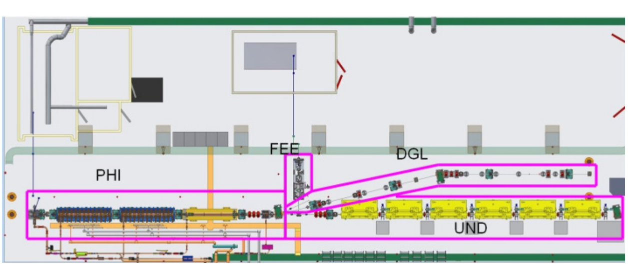


Figure 1: SPARC.

## CONTROL SYSTEM DESCRIPTION

The control system should guarantee and simplify machine operation. In general the main operations in an accelerator control system are: data taking, display of information, analysis, command execution and storage.

The simplest and functional control system has distributed processors on a classic three levels architecture.

- *First level*: at this level we find the console with its human interface to allow the operator to control the machine, a logbook to share information within the collaboration, a database to store all information coming from the machine and a web tools to help the management of the control system and to share some information outside the collaboration;
- *Second level* at this level we find the front-end CPU that executes commands and handle all the information about the status of the machine available at the first level. Meanwhile it automatically saves data from its various elements in two ways: on value changes and/or at fixed time intervals;
- *Third level* is the acquisition hardware where we find an appropriate acquisition board or the secondary field bus to acquire data from the real element.

The interconnection bus between the levels is a Gigabit Ethernet LAN.

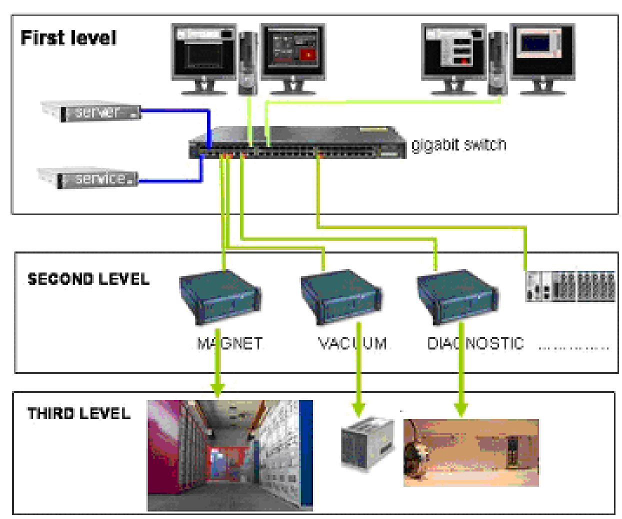


Figure 2: Control system structure.

### Hardware

First of all we decided that each distributed CPU controls only a certain type of elements. This simplifies the number and type of acquisition board assigned to the front-end processor. We want to use the right processor in depending on the element to control. We decide to implement in our system PXI bus, industrial personal computer and real time processors.

At console level we need the maximum flexibility in terms of number of screens and possible remote connections. Also at this level, we plan to use small form factor PCs with at least 4 monitors each.

A storage facility for the whole system software and data is foreseen.

### *Software Environment*

In order to reduce the time of development of the SPARC control system, we decided to use well known Rapid Application Development (RAD) software. Labview became the natural choice for the following reasons:

- in the collaboration laboratories the use of National Instrument software is very popular (we can say it is a "standard");
- Labview is used as development software in the DAFNE [3] control system. This choice allows us to re-use, when possible, already existing software;
- Labview is considered as reference software by many hardware manufacturers that write interface drivers in Labview.

Other software such as Matlab (whose integration with LabView is very well established), Mathematica or self-made will be introduced and integrated in the control system in order to help the online and offline analysis of the beam during the machine operation.

## ELEMENT

The control system allow to control all machine elements from the Laser until undulator and their diagnostic instruments.

In table 1 is described the kind of elements under control and their own interface.

Table 1: Element Controlled

| Element | Number | Interface |
|---|---|---|
| RF Modulator | 2 | TCP/IP |
| RF Low Level | 1 | PXI Digitizer |
| Vacuum Pump | 30 | DAC, ADC, IO, Serial |
| Vacuumetr | 12 | Serial |
| MagnetPS | 50 | Serial, ModBus |
| Flag | 24 | Serial Motor |
| Camera | 24 | IEE1384, GigaEth |
| BP M | 12 | Bergoz ADC |
| BCM | 2 | Bergoz DVM |
| Faraday Cup | 1 | High speed digitizer |
| Laser Photodiode | 1 | High Speed Digitizer |
| Filter wheel | 3 | Serial motor |

### *Element Abstraction*

To control an element means control its hardware interface.. For example to control the magnet mean control its power supply.

In this case we make an abstraction of the power supply to allow include the main characteristic of the element and we define a class.

We have two kind of cluster, see fig., one contains all information that is necessary to control the power supply called static part it contain the kind of interface, the eventually conversion factor and some parameter such as min max value and so on. The second cluster contain all dynamic information such as the current, the voltage, the status and the error.

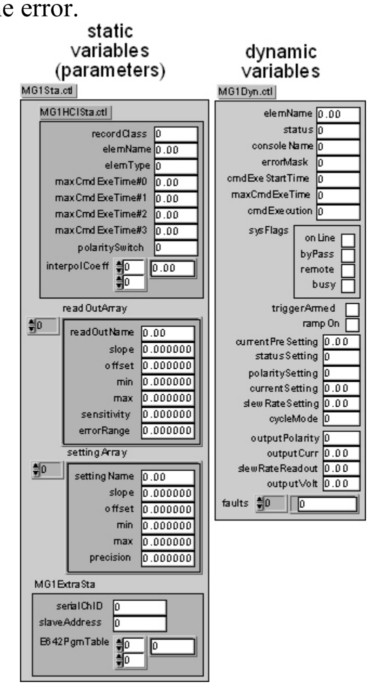


Figure 3: Magnet clusters.

### *Communication Transfer*

To transfer the information we decide to transfer the whole cluster this because the transfer time is similar to transfer a single information (the normal dimension of a dynamic cluster is normally less than 1500 byte). The choice of data is performed by the console software. If we need to transfer a big amount of data, ex. image or waveform, we developed a specific data tape. All the data are transferred in binary code

### *Front-End Software*

In the developing of the control system we decide to develop software's skeleton that contain always the same part this reduce the debugging time (fig. 4.)

This software contains the follow part.

- Init: initializes all the variables for data communication and acquires information about the elements under controlled and perform any hardware initialization Ex. initialization of the serial ports.

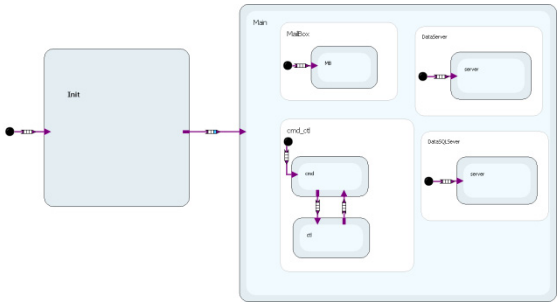


Figure 4: The front-end skeleton.

- Main: is constituted by a serious task running in parallel.
- Mailbox: acquires any commands from a console or from another computer puts them in a queue for execution.
- Cmd_Ctl: execute the command if present in the command queue. The second operation is the control operation. This operation consists in the continuous acquisition of the information from the hardware control and some automatic operation ex the phase feedback in the low level RF.
- DataServer: sends the clusters of the elements under control at the request of the higher level. This information, as mentioned, can be static, dynamic and large amount of data.
- DataSQLServer: automatically writes a SQL database (PostgreSQL) of the parameters considered most significant. Saving can be done either periodically or upon significant change.

### *Console Software*

Machine control is via a series of windows that lets you view and send commands to control elements.

The window can contain one class of element or more if it is necessaries.

In the control window software must contain always some part (fig.5)

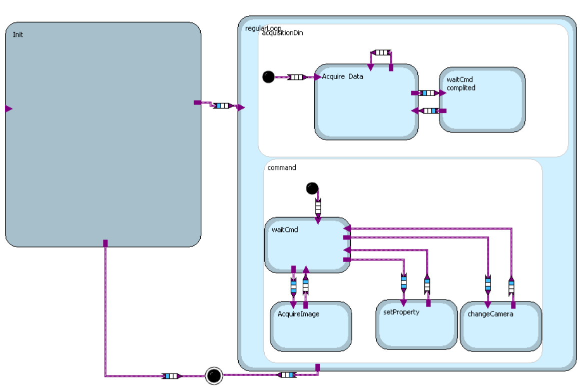


Figure 5: Diagram camera control.

The init part initialize the communication information find the TCP-IP address and port of the elements.

The main loop is divided in two part one is dedicated to acquire the information from the front-end and display or analysis the second is dedicate to send the appropriate command to the specific element.

## SERVICE PROGRAMS

The SPARC collaboration involves different national and international research institutions. Some services are necessary to allow all people to have the information available on the status of the machine and the progress of the work. The old system based on a paper logbook where the operator writes the data and glues pictures on it can be useful but cannot be available from remote researchers. We choose to develop an electronic logbook based on PostgresSQL [11]. These choices allow us to customize and integrate it in the control system.

### *Status Log Machine*

During the e-meter operation we start to study the possibility to have an automatic saving this mechanism can be useful in the maintenance of the machine and in the offline analysis.

So there is data acquisition system based on a database with a possibility to communicate via TCP/IP (PostgreSQL).

In the first operation we have each front-end processor running programs that send periodically all data of the controlled elements to the database. We have developed some different interface programs that can correlate the information

## ACKNOWLEDGEMENTS

We want to thank Luis Antonio Rossi and Serena Strabioli for their contribution in the hardware installation.

We are also grateful to all the SPARC staff for their continuous suggestions and encouragement for making a good and useful job.

## REFERENCES

[1] The SPARC project is financially supported by the EU Commission in the 6th FP, contract no. 011935 EUROFEL and contract no. RII3-CT-2003-506395 CARE.
[2] SPARC Project Team, Sparc Injector TDR http://www.lnf.infn.it/acceleratori/sparc/.
[3] G. Di Pirro et al. "DANTE: Control System for DAFNE based on Macintosh and LabView", Nuclear Instrument and Methods in Physics Research A 352 (1994) 455-475.
[4] LabVIEW, National Instruments Corporation, 11500 N Mopac Expwy, Austin, TX 78759-3504 USA (http://www.ni.com).