

EXPERIENCE IN USING BASED EMBEDDED CONTROLLERS WITH EPICS ENVIRONMENT FOR BEAM TRANSPORT IN SPES OFF-LINE TARGET PROTOTYPE

M.Montis, M.Giacchini, LNL INFN, Legnaro, Italy

Abstract

EPICS[1] was chosen as general framework to develop the control system of SPES facility under construction at LNL[2]. We report some experience in using some commercial devices based on Debian Linux to control the electrostatic deflectors installed on the beam line at the output of target chamber. We discuss this solution and compare it to other IOC implementations in use in the Target control system.

INTRODUCTION

SPES accelerator, under construction at the Legnaro National Laboratories (LNL), is on ISOL facility (Isotope Separation On-Line) dedicated to the production of radioactive ion beams with high energy and high degree of purity. In a structure like this one, the core of the project are the production target and the ionization ion system, which were carried out a study and design from scratch. This structure is currently in production for execution of experiments from the beginning of 2010.

Analyzing the characteristics required for inspection of this apparatus, it is possible to observe that:

- an heterogeneity of hardware and software solutions are required;
- we have to manage a distribute control system;
- it is mandatory a real-time processing for every control variable, in order to ensure a robust system.

According to specific requests for software control, optimization of costs that derives from using an opensource software and favouring cooperation between laboratories, EPICS was chosen as control system software for the SPES facility.

The control network architecture created to manage the off-line target prototype is a multi-level structure composed by machines dedicated to network management and all EPICS hosts that implement the real control system environment. For managing all the devices used for the ion beam production, we decided to use PLC solutions in every sub-system in which safety is mandatory and it is necessary to implement fault-tolerant solutions, while particular embedded systems equipped with EPICS software manage all the devices dedicated to the ion beam manipulation (ion beam acceleration and ion beam focus)[3]. Appropriate solutions have been implemented for the integration of PLC subsystem within the EPICS environment, which it will explain in the next section.

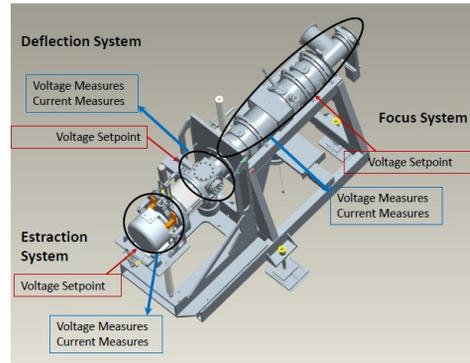


Figure 1: SPES Off-Line Target Prototype: systems and signals managed.

THE CONTROL SYSTEM

System Description

Managing the ion beam geometry, linearity and acceleration is made through the adjustment of the intensity of electrostatic fields generated by appropriate medium voltage power supplies (0-4kV) and high voltage power supplies (0-65kV). Controlling these power supplies it is possible to supervise the ion transport efficiency and optimize the emittance index. For this reason it is necessary to adopt appropriate control strategies to ensure a sufficient level of precision in the voltage settings, and security to all the devices managed.

We decided to base control algorithms on the following strategies:

- *voltage control*: initially we created a closed-loop system with PID regulators for controlling all the power supplies used in the deflection, focus and extraction systems. After this we changed the control strategy and we implemented a new control based in bisection algorithm because during the runs users had to usually change control parameters;
- *current monitoring*: the presence of any current could lead to the establishment of electrical discharges, resulting in the breakdown of components. For this reason we chose to implement a current monitoring strategy, with system shutting down when values exceeded a preset alarm threshold. Because of current values depends on power supplies work condition, we decided to define two different kind of thresholds:

1. one current threshold for voltage's transitory;

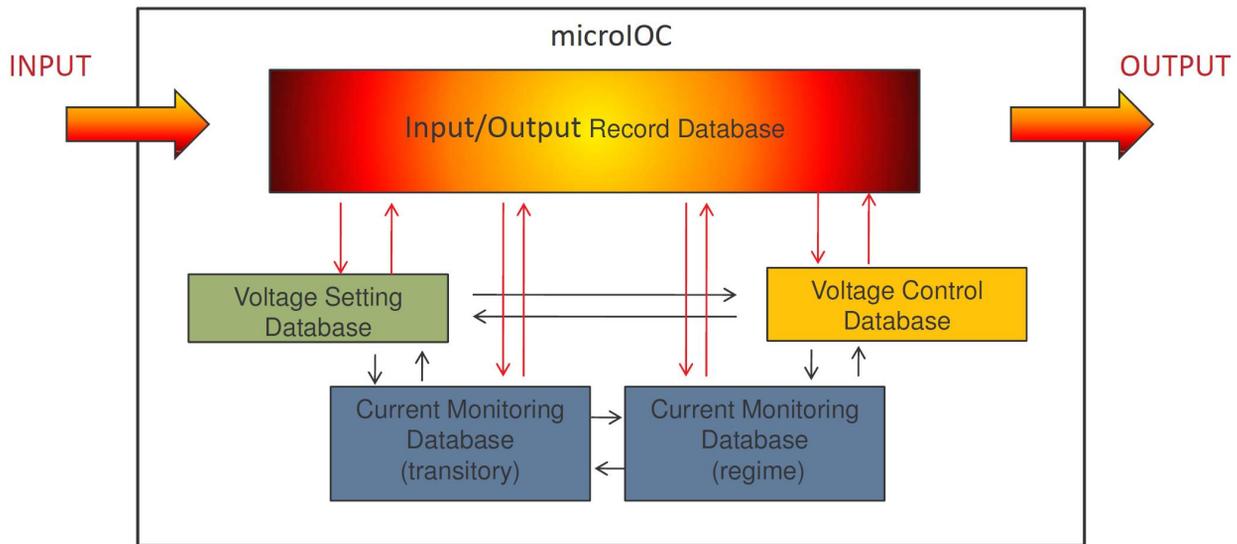


Figure 2: EPICS application developed: it is possible to see the connections between all the Record Instance Files defined and the signals from the field.

2. one current threshold when power supplies work in regime (constant voltage).

Appropriate EPICS Databases these strategy, and all the databases created for the power supplies remote control define the EPICS control system application used to manage the ion beam accelerator and focus systems for the SPES off-line target prototype.

Hardware Solution

While for the others sub-systems that constitute the SPES off-line target prototype we used computer general-purpose to control them, for the accelerator and optic sub-systems we decided to use particular embedded devices equipped with data acquisition cards and the software needed to define an EPICS Input/Output Controller (IOC).

The hardware chosen is an embedded computer (*microIOC*) based on the standard PC/104 and distributed by Cosylab, which allows for a robust and flexible remote control of devices connected to it, thanks to the absence of moving parts (such as hard drives and cooling fans) and a variety of communication interfaces available.

The Channel Access servers, used within the network for systems management of optical beam acceleration, are *microIOC* with the following characteristics:

- VGA connection;
- two ethernet connections, one in DHCP and one with static IP;
- two USB connections;

- one RS232 connection;
- one AD I/O board (ADIO 104-AIO12-8) used for data acquisition and signal processing. It is possible to manage:
 - 16 digital input;
 - 16 digital output;
 - 24 analog input (12bit resolution);
 - 12 analog output (12bit resolution).

Using this kind of boards we can control all the signals desired making an error of $\pm 1V$ for the medium voltage power supplies and $\pm 15V$ for the high voltage power supplies. While in the first case the error is acceptable, for the high voltage power supplies it is mandatory to decrease this value. One possibility suggested is to replace the ADIO board with another board having at least 16bit resolution for the analog input/output.

The operating system (a Debian based GNU/Linux custom distribution) and the EPICS software are stored in a Flash Card memory card. The EPICS environment is configured to initialize an IOC application at boot time. In this way the *microIOC*, when connected to the network, automatically creates the Channel Access Servers without further configuration by the user. The ADIO board uses a special *EPICS Driver Support* to interact with the control system environment and manage all the digital and analog signals.

During studies made and through the EPICS Community, we analyzed that the most used way to interface PLC systems with the EPICS environment is using particular hardware called OPC-Servers, that define a layer where the PLC signals are mapped in EPICS Process Variables. To optimize costs and minimize hardware complexity, we decided to implement it with a different communication interface using the ADIO cards provided by microIOC.

For monitoring the embedded systems and the EPICS environment, we used an appropriate machine equipped with Nagios[4] software; in this way it is possible to control both the hardware and the software status.

Software Solution and Application Developed

During code development we tried to find a database structure which would allow to minimize the number of Process Variables we have to manage and simplify the software maintenance. In this way it is possible to have a synthetic and readable code for anyone who have to monitor the SPES control system infrastructure. Following the project directives, the control strategy adopted and studies made with other EPICS applications we decided to develop a program structured in five different *Record Instance file*. Analyzing Figure 2 shows:

- one Database, using the EPICS Driver Support, defines the interface between the EPICS environment and the AD I/O board mounted on the microIOC. This Record Instance File allows to manage all the inputs coming from the field and the outputs used to control the devices desired, such as indicated in Figure 3;
- one Database manages the Records used into the Graphical User Interface for setting all the power supplies's control parameters;
- one Database monitors the currents provided by the power supplies when they work on regime;
- one Database monitors power supplies's currents during voltage transitories;
- one Database realizes the voltage control strategy (PID regulator and bisection algorithm).

Realizing a modular structure for the software allows us to modify it in function of the experiment's requests (concerning the control strategies) and the technical requests (for example hardware maintenance) without changing the entire application: for example we modified the control strategy switching between two different databases (as represented in Figure 4) keeping unaltered all the other record instance files, except for fews links used to forward the process chain.

From an informatic point of view, Databases are built up with couples of template/substitution files: through this EPICS environment feature, it is possible to modify all the Process Variables desired (number, type, attributes) and the

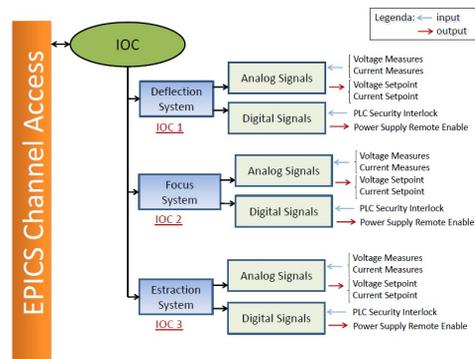


Figure 3: EPICS Control System Structure.

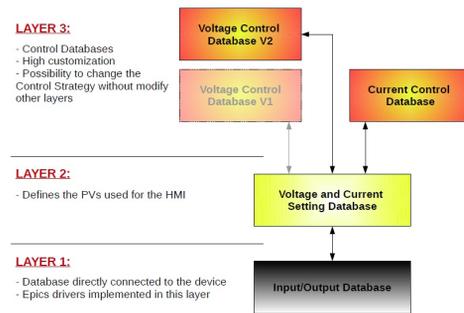


Figure 4: EPICS Application Database - Layers.

database architecture in a fast way editing a little set of text files; at the same time, if someone prefers use graphical editors (such as *VisualDCT*) to edit template files, the database structure will result easier to read and modify, because all the Records needed by the application will be instanced only at the IOC boot time. In this way both the debug phase and the maintenance are simplified; in particular this second feature is really important in a project like SPES, where all the works are developed and managed by different people.

For defining Process Variable names, we created a special Naming Convention that provides a descriptive name for every Record and, through it, every user can easy find and modify all the control parameters desired. This Naming Convention will be adopted for the entire facility's control system.

For every EPICS application developed, we created a specific Graphical User Interface (GUI) with Medm, an EPICS tool that provides an complete editor for creating and managing GUIs; an example of Medm GUI is visible in Figure 5. These interfaces are only used by software developers for managing the EPICS control system software, while specific GUIs developed in CSS are used by general users to supervise the SPES off-line target prototype and, in the future, all the SPES facility.

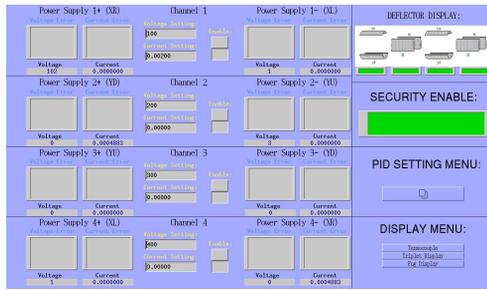


Figure 5: EPICS GUI developed for system maintenance.

REFERENCES

- [1] EPICS – Experimental Physics and Industrial Control System www.aps.anl.gov/epics/.
- [2] EPICS at Laboratori Nazionali di Legnaro <http://www.lnl.infn.it/~epics>
- [3] M. Montis, Master's degree thesis, Università degli Studi di Padova, a.a. 2009-2010 <http://www.lnl.infn.it/~epics/THESIS/TesiMaurizioMontis.pdf>.
- [4] NAL (Nagios Alarm Handler)(Not Alarm Handler) <http://www.lnl.infn.it/~epics/NAL.html>

CONCLUSION

Particular hardware and software solutions are adopted to design and develop the ion beam accelerator and focus systems in SPES off-line target prototype. On the hardware side, we decided to use specific embedded devices based on the PC/104 standard equipped with all the EPICS tools needed to connect them into the off-line target prototype's control system. During this production period we observed that this hardware solution guarantees good performances. This hardware allows to create a transparent layer between PLC systems and EPICS software too, minimizing possible incompatibilities between these two environments.

On the software side, the EPICS application structure adopted permits an easy management by users and a easy reuse of most of the code wrote for others applications and experiments, allowing a minimization of setting times. In general a modular approach for developing EPICS applications is desirable, especially in large and distributed environment like the SPES facility.