ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA155

PLC INTEGRATION IN EPICS ENVIRONMENT: COMPARISON BETWEEN OPC SERVER AND DIRECT DRIVER SOLUTIONS

L. Antoniazzi, A. Baldo, M. Giacchini, M. Montis, INFN-LNL, Legnaro, Italy

Abstract

In the IFMIF EVEDA project [1], INFN-LNL Laboratory has been involved in the design and construction of a normal conducting Radio Frequency Quadrupole (RFQ) used to bunch and accelerate a 125 mA steady beam to 5 MeV. The EPICS based control system [2] has been entirely developed in house using different hardware solutions: PLC for tasks where security is the most critical feature, VME system where the acquisition speed rate is crucial, common hardware when only integration is required without any particular feature in terms of security. Integration of PLCs into EPICS environment was originally accomplished through OPC DA server [3, 4] hosted in a Windows embedded industrial PC. Due to the issues analyzed in injector LCS, LNL proposed to migrate to the usage of EPICS Direct Driver solution based on s7plc [5]. The driver itself is suitable for direct communication between EPICS and PLCs, but it doesn't take care of data update and synchronization in case of communication failure. As consequence LNL team designed a dedicated method based on state machine language to manage and verify data integrity between the two environments, also in case of connection lost or failure.

INTRODUCTION

The main objective for IFMIF (International Fusion Materials Irradiation Facility) is the construction of a linear accelerator for neutron irradiation effects on materials that will be used to realize future fusion reactors.

The IFMIF facility will provide an accelerator-based neutron source that produces, using deuterium-lithium nuclear reactions, a large neutron flux with a spectrum similar to that expected at the first wall of a fusion reactor (Fig.1). The main components of the apparatus for the neutron beam production are therefore the following:

- the generation system of deuterons, consisting of two linear accelerators in parallel each producing a current of 125mA beam and made up of an ion source (INJ), a low energy beam transport (LEBT), a radio frequency quadrupole (RFQ), a medium energy beam transport (MEBT), superconducting cavities and high energy beam transport (HEBT);
- the lithium target and the associated circuit for the evacuation of the produced power;
- test cell where are arranged the samples of the materials to be tested.

Because of the complexity of the project, its implementation requires a preliminary step related to the validation of the prototypes. For this reason, IFMIF-EVEDA (Engineering Validation Engineering Design Activities) involves the construction of prototypes of each of the components mentioned above. Prototype of the accelerator facility named LIPAc (Linear IFMIF Prototype Accelerator) is currently being commissioned at QST site, Rokkasho, Japan [6]. In this scenario the Italian contribution is related to the construction of the RFQ system in charge of INFN-LNL.

During 2016 and the first half of 2017 the RFQ and its ancillaries, including Local Control System (LCS), were assembled at the Rokkasho site in Japan, currently the hole apparatus are in use for the RF conditioning [7].

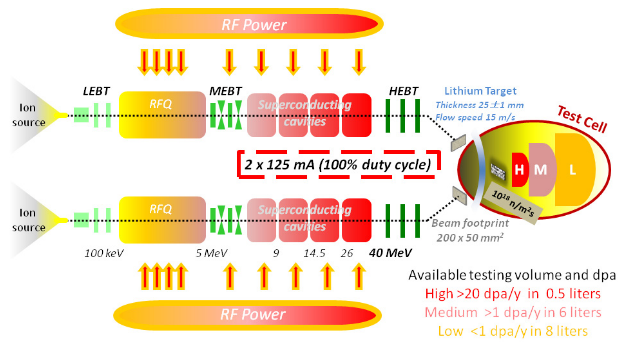

Figure 1: Schematics of IFMIF facility [1]

IFMIF EVEDA RFQ LCS

The RFQ Local Control System (LCS) architecture is designed to optimize the reliability, robustness, availability, safety and performance minimizing the costs related to its purchase and maintenance. Following this philosophy and the IFMIF-EVEDA guidelines, the control system network is composed by two different kinds of hosts:

- physical hosts for critical control system tasks;
- virtual hosts where no particular functional tasks or hardware is required.

The architecture realizes the 3-layer structure and each layer defines a proper hosts group (equipment directly connected to the apparatus, control devices, Human-Machine Interface) while the EPICS framework provides the interface between them (Fig. 2). In the final stage LCS is integrated in the LIPAc Central Control System (CCS). The upper software layer of the CCS realizes the common layer where all the different LCS will communicate and share information without any additional effort [8].

The RFQ system is a complex apparatus composed by many kinds of subsystems (radio frequency, vacuum, water cooling, etc.) developed using different hardware solutions. As a consequence, every part of this structure must be properly integrated to obtain the desired degree of control. Following these criteria, the system was designed and realized using these assumptions:

- PLC hardware is chosen in tasks where security is the most critical feature;
- VME system is used where the acquisition speed rate is crucial;

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA155

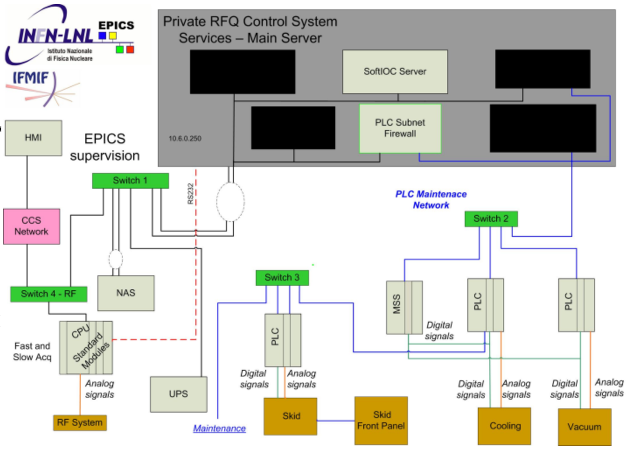

Figure 2: IFMIF EVEDA RFQ LCS architecture.

The main functionalities of LCS are:

- Fast acquisition system for the RFQ cavity power;
- Vacuum system;
- Cooling system;
- Machine Protection System (MPS)

The fast acquisition is based on VxWorks real time OS which run over a VME architecture. The most important signals about RF power are sampled with a maximum rate of 1MEvents/s.

Vacuum, Cooling and MPS functionalities are entirely realized by SIEMENS® S7-300 PLCs and modular safety system (MSS). The integration of these PLCs into the EPICS control network was originally accomplished by an OPC (OLE [Object Linking and Embedding] for Process Control [3]) DA (Data Access) server hosted by a dedicated Windows embedded industrial PC, as foreseen by the project guidelines. The entire system architecture and functionalities were verified during the RFQ high power test executed at LNL in the middle of 2015.

In a next stage, due to reliability and maintainability issues detected both in the injector and RFQ LCS, LNL and project coordination agreed to adopt the S7plc EPICS driver, originally developed by SLS (Swiss Light Source) [5]. Migration from the already developed systems, to a so different communication method, required several changes on both sides as well as the introduction of common rules to manage the variable exchange on the communication channel.

OPC DA USING EXPERIENCE

Use of OPC DA server as communication layer between S7 protocol and EPICS is a common and convenient solution, simply to implement, which provides several benefit in particular for PLC code development. In fact, the complete memory addressing leaves the developer free to organize the code in the most convenient way, on the other hand the identification of the exchanged variables could be quite complicated due to the possibly spread of them in PLC code. Other advantages are related to the configurable variable access rights provided by OPC (read, write and read/write), bidirectional read/writes access and multiclient architecture applicable at different layer (PLC, OPC server or EPICS) permits a easy integration without any effort on variable sharing. Thanks to this features each change in variable values is automatically transmitted to the rest of the system, the drawback of this feature is the possible overwriting of the PLC data in case of IOC reboot. Under the network aspect, the OPC performs a convenient separation between PLC sub-network and the rest of control network.

During the RFQ power test at LNL a Windows embedded industrial PC (Siemens IPC427) hosted the Simatic Net OPC server, IOC application and device support for the OPC server itself to monitor and control about 1100 PVs (760 related to vacuum system and 340 related to cooling system) organized as one unique group with a refresh rate of 1s. The device support module used is the one developed at Helmutz Zentrum in Berlin (HZB) [4]. Even though OPC DA solution was used during RFQ power test, we had several issues related to variable overwriting during IOC reboot; in particular cases we observed a mismatch in data communication and information update between the IOC and the PLC: while the Graphical User Interface (GUI) at EPICS level provided a normal behaviour (no data disconnections), at IOC level the PVs were freezed. In this condition we found an irregular CPU working load (with spike of 100%) on the OPC server.

APPROACH TO S7PLC DRIVER

The S7plc driver is based on send receive over TCP/IP communication which essentially connect EPICS output PVs with PLC receive buffer and PLC output buffer with EPICS input PVs; by this, it is evident that bidirectional access to the variables is no more possible, as for example a RESET command set true by EPICS which is executed and set back to false by PLC. This aspect implies several changes on both sides to modify the logics which required bidirectional access.

For these reasons moving from OPC DA server to a TCP/IP send/receive based communication several issues were considered for PLC implementation:

- convenient variable naming and organization became crucial to identify the variables to be exchanged;
- link the variable to the send receive buffer in accordance to the connection status;
- definition of common policies on data writing access right between PLC and upper layer.
- support bidirectional write access at list for particular set of variables.

The already adopted standardization of the PLC code of RFQ LCS, derived by [9], simplified the first 2 point and it will be discussed in detail later, while the main assumption adopted for variables writing access right was that PLC process control data must contain at each moment the correct values. In particular, during connection establishment, PLC reboot or IOC reboot the values in PLC memory have to be maintained and updated into the IOC variables; only after the execution of this stage command and configuration variables could be transferred from the EPICS IOC to the correspondent variables in the PLC memory.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA155

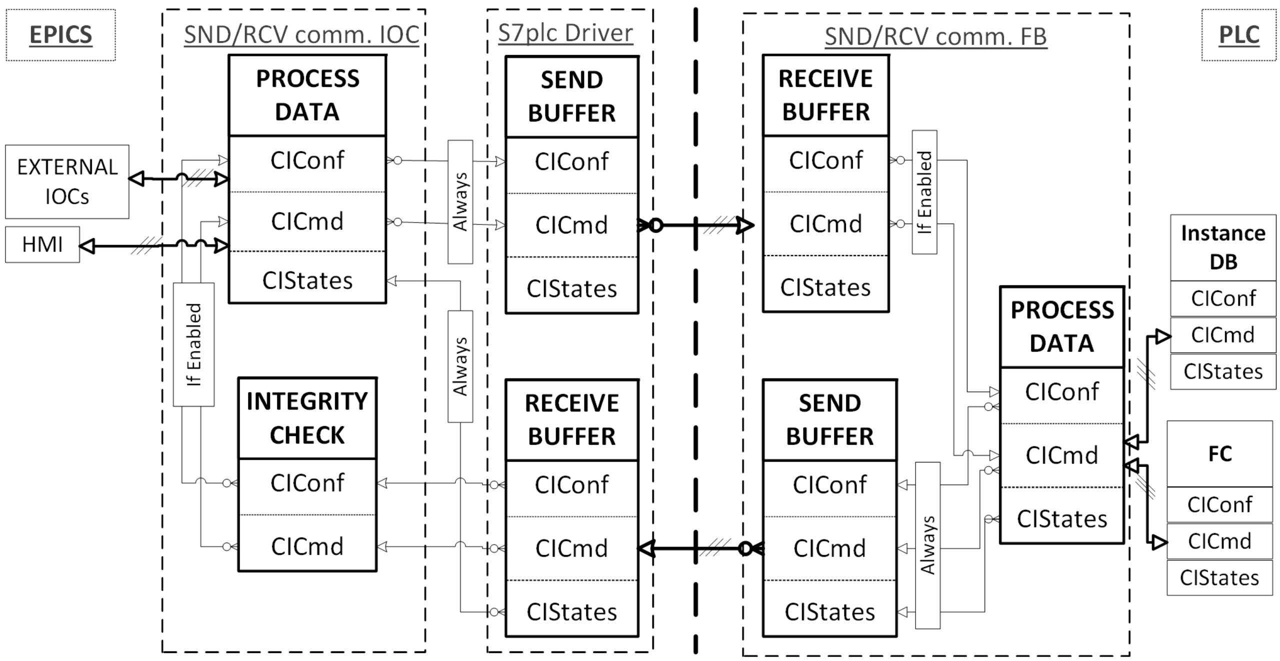

Figure 3: Data exchange between EPICS and PLC.

Maintaining the possibility of modify configuration parameters on both sides gave us several benefits:

- reduced modification to the PLC code already developed,
- allows maintenance operation both by PLC or EPICS experts;
- match the cooling system particular requirements needed for its operations.

From network point of view, in order to maintain the separation between PLC and EPICS network a firewall was introduced.

PLC CODE ORGANIZATION

The PLC code standardization adopted by the RFQ LCS [9], based on ITER standard, widely use UDT (User Data Type) to organize the input/output data of the PLC functions and function blocks (FC and FB) in 5 main groups:

1. PIIn (Peripheral Interface Input);
2. PIOut (Peripheral Interface Output);
3. CIConf (Control Interface Configuration);
4. CICmd (Control Interface Commands);
5. CIStates (Control Interface States).

While Peripheral I/O concern physical I/O, control interface UDTs are dedicated to data exchange between functions in the same PLC, between different PLCs or with third party control systems; configuration and commands are typically inputs for PLC while states are output provided to the upper layer. The UDT organization uniforms the naming and aggregates variables with similar scope giving evident advantages to link them with the memory Data Block (DB) dedicated for the send/receive communication.

Figure 3 shows how the variables are transferred between PLC and EPICS, highlighting the conditions which manage the writing rights on CIConf and CICmd variables. Note that bold arrows in Fig. 3 are related to the set of three variable groups. For the PLC part a specific FB was developed to manage the data between send and receive buffers and the process data in a consistent way, in accordance to the write access rules, taking into account the connection status. Process data DB was organized using structures which collect the homonymous UDTs directly linked to the application FB or FC. The PLC's operator or the automatic routine in the PLC can use the instruments provided by the developed communication function to take the writing right on CICmd and CIConf variables when needed.

Communication Set-up

PLC communication set up is executed as in [5], but in our case data channel initialization and timing for send receive are in charge of the EPICS IOC, while PLC sent data only after the receiving. When there is no connection, connection time-out or in case of connection initialization, PLC has writing rights on CIConf and CICmd variables.

EPICS ALGORITHM FOR S7PLC DRIVER

As anticipated, in our configuration, EPICS application is in charge of initialize the communication channel and manage sent receive timing. To administrate the initialization procedure and further commands, which allow to modify online the variables writing access right, the developed IOC includes a specific integrity check procedure. As showed in Fig. 3. In particular, while CIStates PVs provided by the S7plc driver are directly the ones shared with the rest of EPICS environment, the CIConf and CICmd PVs are copied into the correspondent PVs connected to the process control only if permitted by the integrity check procedure. The integrity check procedure

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA155

is based on a state machine running on the IOC (Fig. 4). Transitions between the various status depend by communication state, timer and principally by a subset of PVs dedicated to communication management, located at the header of the send/receive buffer exchanged between IOC and PLC, to coordinate each other.

For the RFQ LCS one IOC for each subsystem (vacuum and cooling) was developed. The shared variables were further organized in different connection sockets, this to allow the set up different refresh rate, or switch on/off a variable group if needed. Moreover, generation of the substitution file has been automated with a python script which extract information from PLC send/receive buffer DBs, avoiding as much as possible errors in this task.

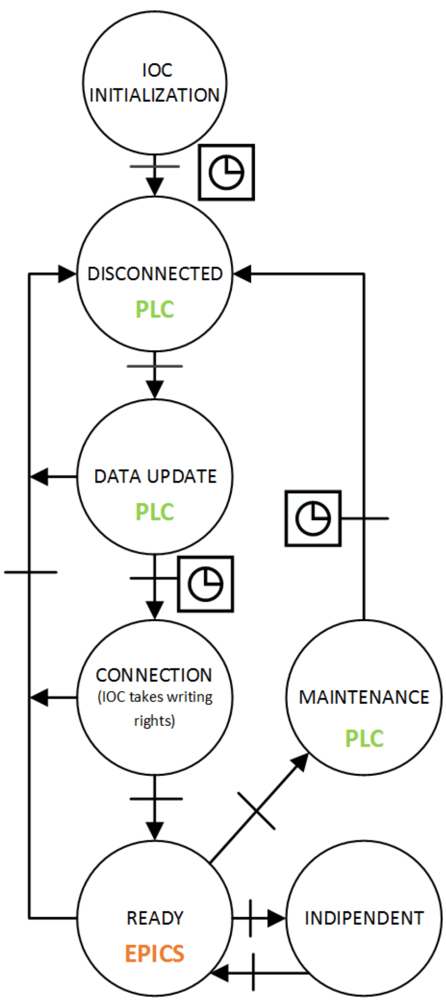

Figure 4: State machine implemented by the communication IOC

Table 1: PLC-EPICS Shared Variables

PLC	Sockets	Variables	Buffer Size
Vacuum	Fast	1361	2260 Byte
	Slow	770	2122 Byte
	Total	**2131**	**4382 Byte**
Cooling	Fast	657	2032 Byte
	Slow	380	998 Byte
	Skid Fast	363	1388 Byte
	Skid Slow	623	1592Byte
	Total	**2023**	**6010 Byte**

CONCLUSIONS

The migration from OPC DA to s7plc required several efforts on both side (EPICS and PLC), especially to keep bidirectional variable access, but it pays back us with:

- consistent data transfer and initialization procedure;
- well organized communication method on PLC side;
- possibility of define several connections respect the exchanged data functionality or scan rate;
- easy export of send/receive buffer DB to EPICS data base.

As report in Tab. 1, now in RFQ LCS the PVs exchanged each second via S7plc driver are: 2131 (4,3kB) on 2 connections for vacuum PLC and 2023 (6 kB) on 4 connections for cooling PLC, so we largely exceed the limit of 1800PVs suggested for the OPC DA driver. Grown up of exchanged PVs is partially due to progress of the system, but the main cause is the new communication strategy adopted, which require more memory and execution time to the PLC respect to OPC solution.

REFERENCES

[1] IFMIF, http://www.ifmif.org

[2] EPICS, http://www.aps.anl.gov/epics/

[3] OPC, www.opcfoundation.org

[4] OPC device support, www-csr.bessy.de/control/SoftDist/OPCsupport

[5] S7plc, http://Epics.web.psi.ch/software/s7plc/

[6] P. Cara *et al.*, "The Linear IFMIF Prototype Accelerator (LIPAc) design development under the european-japanese collaboration", in *Proc. 7th Int. Particle Accelerator Conf. (IPAC'16)*, BEXCO, Busan, Korea, May 2016, paper MOPOY057, pp. 985-988.

[7] E. Fagotti *et al.*, "Installation and low power test of IFMIF-EVEDA RFQ at Rokkasho site", in *Proc. 8th Int. Particle Accelerator Conf. (IPAC'17)*, Copenhagen, Denmark, May 2017, paper THPIK032, pp. 4162-4165.

[8] M. Montis *et al.*, "IFMIF EVEDA RFQ Local Control System integration into main control system", presented at the 16th International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS 2017), Barcelona, Spain, Oct 2017, paper THPH157, this conference.

[9] Evrard B., "PLC Software Engineering Handbook", ITER, 13067 St. Paul-lez-Durance, France. ITER 3QPL4H, Jan. 2013