INTRODUCTION OF NON-STANDARD EPICS CONTROLLERS

I. Badillo, M. Eguiraun, M. del Campo, I. Arredondo, D. Piso, ESS-Bilbao Consortium, Spain
J. Jugo, University of Basque Country UPV-EHU, Spain

Abstract

Although EPICS is a mature software framework, the study and validation of new configurations of EPICS systems is very valuable, since new ideas open its evolution and improvement. Therefore, the goal of the present work is to introduce new technologies under EPICS control structures and test different configurations with innovative hardware in this kind of applications. Specifically, it is intended to validate the use of non-standard EPICS controllers. This paper presents a testbench using LabVIEW together with EPICS. LabVIEW eases and speeds up the development of control structures, avoids the hardware dependent developing costs and offers almost absolute compatibility with a high variety of hardware devices used in control and data acquisition. To validate its use in a real environment, is mandatory to make a study facing this solution and EPICS standard methodology, specifically CODAC system used in ITER. With such objective in mind, a testbench is implemented running both configurations and its results are compared. Following this scheme, as conclusions, the next step must be to implement exactly the same hardware-level structure for both approaches to improve the comparison.

INTRODUCTION

EPICS is one of the most important and widely used control systems oriented to large scientific facilities. This is a Big Physics standard based on a middleware approach, used worldwide on large scientific plants such as particle accelerators and telescopes and and it can be described as a set of open source tools, libraries and applications to create soft real-time distributed control systems. Nowadays, it offers solutions for most of the control needs and is compatible with a large variety of hardware platforms and operating systems (Linux, Windows, VxWorks...) available in the market.

Summarizing, EPICS is a mature software framework. The study and validation of new configurations valid for EPICS systems is always a very valuable research, since new ideas open its evolution and improvement. In this sense, the main objective of the work sustaining this paper is the study of the inclusion of non-standard control structures in EPICS networks. Here, the first steps towards this direction are presented. Two main schemes are considered: the standard architecture of EPICS systems using classical IOCs under Linux machines, and a system which integrates LabVIEW and EPICS. The validation of the proposed system is made by comparing their reliability to the classical scheme. The hardware setup differs as in the LabVIEW approach I/O signals are software generated instead of using real DAQ. However, from a EPICS level view, both systems are completely analog... A testbench has been implemented for such comparing process.

The classic EPICS methodology shows a set of distributed Linux machines implementing IOCs. They are responsible for communicating EPICS with the control system tools and devices (motors, sensors, data acquisition systems, switches etc.). This approach requires the development of drivers (or equivalent mechanisms as EPICS devices) for each new device, which are the interface between EPICS records (set of process variables) and hardware (or 3rd party software). These drivers can be split into two parts: device support and driver. The first one is the interface for records and hardware independent. The second one provides low level software access to the hardware. The development of these drivers requires C language notions, EPICS knowledge (records, scan methods, Channel Access) and experience with I/O hardware (I/O registers, buses, interrupts). That means an extra effort every time a new device is added to the control structure.

The second architecture consists of using LabVIEW together with EPICS. This approach avoids the hardware dependent developing costs of the previous architecture. National Instruments (NI) hardware and software offer support for a high variety of devices and cards, and therefore, there is no need to write specific drivers. Moreover, the use of LabVIEW simplifies the development of the control structures. NI provides an EPICS server integrated in the LabVIEW solution which is based on the LabVIEW DSC module, and runs on the real time system in the PXI controller. The Real Time controller publishes EPICS PVs taking data from LabVIEW's Shared Variables. This scheme shows interesting advantages, but, before adopting this method of working, it must be validated. To do so, it is proposed to perform two parallel implementations in the testbench: the first one following the classic methodology and the other one using LabVIEW.

In the next section a thorough explanation of the two experimental setups for the validation of each proposed scheme is included, and some comments about the implementation. The last section is dedicated to the future plans for this experiment and conclusions.

EXPERIMENTAL SETUP

The experimental setup that has been developed implements a reduced local controller, which includes data acquisition, sequencing and control. The typical signals involved in a local controller are present in the emulated system. These signals will be the Process Variables (PVs) of the EPICS system. The generation of these signals can be performed by other devices such as signal generators, but,

ISBN 978-3-95450-124-3

in order to have a high level of flexibility within a single system, an FPGA in the first solution as *hardware in the loop*, and software based signal emulation on the second approach have been chosen for this task. Different type of signals are generated (digital, analog), periodically changing their values as well as acting against external events in the case of the classic IOC approach.. The complete set of signals consists of 1960 PVs: 400 binary inputs, 400 analog inputs and 400 calculation type PVs each approach.

As stated in the previous section, two different control systems are set up. The first one corresponds to the EPICS classical approach and the second one to the LabVIEW based solution. Both perform the same control actions and data acquisition, but the way of working is slightly different. The main settings of the test stand can be found in Table 1.

Table 1: Main Hardware and Software Settings of Both Test Stands

LabVIEW approach	*EPICS classical*
PXIe-1082 Chassis	PXI-1031 Chassis
PXIe-8108 Controller	PXI-8106 Controller
2.53 GHz Intel Core 2 Duo	2.16 GHz Intel Core 2 Duo
1 GB RAM	512 MB RAM
LabVIEW RT	Scientific Linux 6.1
LabVIEW 2010 SP1	EPICS R3.14.12.2

The following paragraphs describe each implementation:

- The first setup is taken as a reference. The IOC is running in a NI PXI-8106 embedded controller with Scientific Linux 6 in a NI PXI 1031 chassis.. A NI PXI-6259 DAQ card performs I/O tasks. As explained before, this device requires its own drivers, for both Linux (SL6) and EPICS. Both can be found in the ITER CODAC Core System v3.0 public version [1]. Once installed and configured, the PXI NI 6259 card is ready to be used. A plant, simulating a DC motor, is implemented as Hardware in the Loop (HIL) using NI PXI-7833R multifunction card, which includes a Virtex II FPGA. Taking advantage of the parallel computing offered by the FPGA, a complete set of signals is generated. The IOC database is defined including all the system PVs and their behavior. The number of PVs is increased sharing hardware inputs among them.

- The second hardware setup consists of a PXI chassis from National Instruments (PXIe-1082) with a PXIe8108 controller running LabVIEW Real-Time operating system. The control actions are defined in a LabVIEW (version 2010 SP1) program running in the PXI controller as well as the I/O signal set. These are written into shared variables. LabVIEW also acts as an EPICS server through the *EPICS IO Server* for LabVIEW. This publishes desired control signals to a EPICS network via Channel Access protocol. This method allows the developer to focus on the control without worrying about drivers, since these are provided with the NI hardware and software and the process is almost transparent for the user.

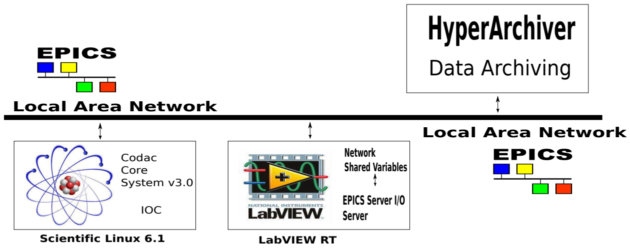

Figure 1: Overall system description over the network.

The overall system is implemented in a conventional Ethernet wired Local Area Network environment in which EPICS 3.14.12.2 is present.

In order to get reliable results concerning repeatability issues, all the processed data of both systems must be archived in a database for a later study. In the present work, a HyperArchiver instance is used, which has been developed at INFN/LNL (Italy) as an extended version of the RDB Channel Archiver, but using Hypertable as its main database. Hypertable is high performance distributed non relational database focused on large scale and intensive tasks and it is an alternative to MySQL, as it is a high performance, distributed and non-relational database. Moreover, it is distributed under GNU license. The version of Hypertable used in ESS Bilbao [2] and, therefore, in the current paper, is a modified version of the original one, used in INFN/LNL. The reason that lead the ESS Bilbao team to modify the original Hypertable was the necessity of managing efficiently arrays of variables. It has been also developed a python based GUI to make the data retrieval fast and reliable. The general networked system is described in Figure 1.

RESULTS

The LabVIEW based approach should be as stable as the classic IOC one, in order to validate it. Up to the moment, several test are been carried out in the laboratory and data measurements have been made after a few days running. Since the main concerning parameters in this experiment are those related to reliability and repeatability, the obtained data and conclusions must be regarded as preliminary.

The overall EPICS system handles up to 1960 PVs, 980 from each structure, similar to the ones presented in [3]. Most of the variables are periodically processed at 1 or 5 second rate. A group of 180 PVs have a passive scan, driven by events. The retrieved data from EPICS is periodically written into the database every 10 seconds.

The extraction of the data from the database is made trough an Python based GUI as is shown in Figure 2, which

ISBN 978-3-95450-124-3

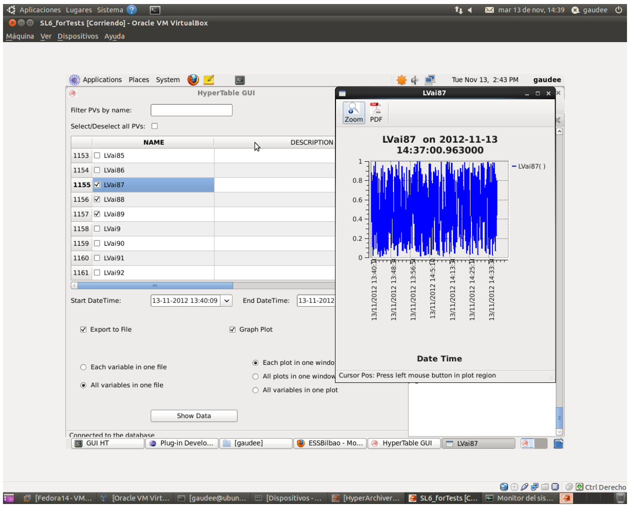

Figure 2: GUI for data retrieval.

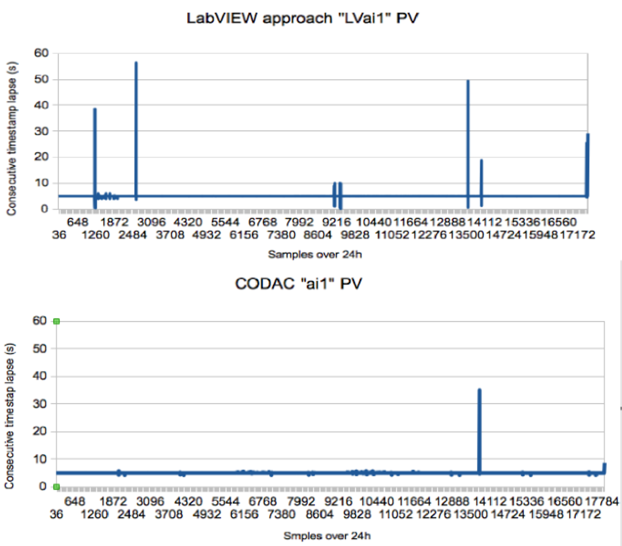

Figure 3: Data losses in both systems.

allows to find and select the desired PVs and choose the output format (which can be a graph plot or a .csv file) in any time interval.

The plots shown in Figure 3 are related to two particular PVs. Those are “ai1” from the CODAC based system and its analog variable called “LVai1” from LabVIEW based approach. The graphs correspond to consecutive timestamps lapses, which should be 5s long in both systems measured in a 24 hour period. Preliminary data analysis show similar data losses in both systems, but they seem to be consequence of the testbench implementation (archiving and network). However, those are not EPICS-related errors, which is working flawlessly in both approaches.So, the experiment environment must be tuned up.

FUTURE WORK

The results obtained from the preliminary test encourages us to follow working in the proposed direction. The overall testbench environment must be tuned up to obtain definitive results for the validation process of the proposed architecture.

The first step to follow is to set up an identical architecture for both approaches. That means to use the same PXI models as well as connecting the input signal set of both systems to the same source in order to have exactly the same reference signals.

Secondly, the test stand must be continuously operating the two configurations in parallel for a long time period (months) in order to get reliable results concerning repeatability issues.

Another interesting idea that is going to be studied is the inclusion of synchronization. At present, this experiment has three different unsynchronized clocks, and therefore three different timestamps: one from each system and a third one from the archiver. So, the use of a NTP server is planed in a near future.

Finally, it is planed to carry out a new testbench introducing an EPICS-based wireless link system maintaining the present two architectures. The main goal is to study the effect of the wireless link, in order to facilitate the replacement of as many cables as possible to benefit from the advantages of the wireless systems [4].

ACKNOWLEDGMENT

This work has been funded by ESS-Bilbao Consortium and GAUDEE research group of the University of the Basque Country UPV/EHU.

REFERENCES

[1] F. Di Maio, L. Abadie, C. Kim, K. Mahajan, P. Makijarvi, D. Stepanov, N. Utzel, A, Wallander, “The CODAC Software Distribution for the ITER Plant Systems”, Proceedings of ICALEPCS2011, Grenoble, France, 10-14 October 2011.

[2] M. del Campo, J. Jugo, M. Giacchini, L. Giovannini, “EPICS HyperArciver: Initial Tests at ESS Bilbao”, Procedings of IPAC2011, San Sebastian, Spain, 2011.

[3] G. Shen, M. Kraimer, “Server Development for NSLS-II Physics Applications and Performance Analysis”, Proceedings of PAC2011, New York, NY, USA, 2011.

[4] I. Badillo, M. Eguiraun, J. Jugo, “NFC like Wireless Technology for Monitoring Purposes in Scientific/Industrial Facilities”, Proceedings of ICALEPCS2011, Grenoble, France, 10-14 October 2011.

ISBN 978-3-95450-124-3