

# IRIO Technology: Developing Applications for Advanced DAQ Systems Using FPGAs

M. Ruiz, Member, IEEE, S. Esquembri, A. Carpeño, J. Nieto, A. Bustos, E. Bernal, D. Sanz, E. Barrera

**Abstract**—IRIO tools are a set of software modules simplifying the development of advanced data acquisition systems (DAQs) using FPGA-based devices. In particular IRIO provides all the integration chain for the development of applications for EPICS middleware. The simplification arises because IRIO defines three main elements: a data acquisition and processing architecture for the FPGA, a software layer interfacing this implementation and an EPICS devices support implemented with `asynDriver` integrating all in EPICS. IRIO uses RIO technology from National Instruments and LabVIEW for FPGA development. The tools have been integrated and tested in ITER codac core system for fast controllers and in one prototype of the ION Source in ESS-Bilbao. IRIO software is distributed under the GPL V2 license.

## I. INTRODUCTION

THE implementation of FPGA-based data acquisition systems in large physics experiments is widely used nowadays [1]. The combination of analog and digital front-ends with FPGAs in industrial form factors such as PXIe, uTCA, or ATCA, allows the implementation of intelligent data acquisition systems (IDAQs) with processing capabilities that run with deterministic behaviour [2]. Additionally, FPGAs provide users the option to change continuously the functionality of their applications using a well-known development cycle and using VHDL-based tools [3]. One of most innovative development cycle for advanced instrumentation systems in PXIe platform is the combination of LabVIEW for FPGA and FlexRIO/cRIO/RIO hardware because the development time is reduced notably [4]. Furthermore, advanced data acquisition applications have to be integrated with other high level applications which are in charge of the configuration and supervision of the whole control system. EPICS [5] is one of the most widespread solution in large research facilities such as APS (Advanced Photon Source), ALS (Advanced Light Source), ITER (International Thermonuclear Experiment Reactor) [6], ESS (European Spallation Source), etc. The typical PXIe hardware/software platform supporting FlexRIO/RIO/cRIO is shown in Figure 1. EPICS is running in a Linux-based computer connected to a PXIe chassis allocating the FPGA-based DAQ devices with the customized hardware applications. The manufacturer provides the kernel driver support for the user to interact with the devices.

Manuscript received May 30, 2016. This work was supported in part by the Spanish Minister of Economy and Competitiveness under the project ENE2015-64914-C3-3-R.

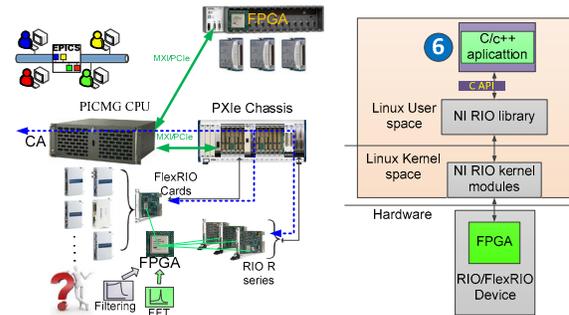


Fig. 1: Architecture of a PXIe system with RIO devices.

IRIO software tools have been designed with the idea of simplifying the development of advanced data acquisition applications and their integration with EPICS. This integration is possible by developing the corresponding EPICS device supports. A challenging point here is the development of a unique EPICS device support that is supporting a user-defined hardware based in a FPGA. If the user changes the design in the FPGA following a predefined rules the EPICS device support will recognize the resources and connect the EPICS PVs to the aforementioned resources. IRIO tools meet all these requirements providing the user with these elements (see Figure 2):

- A collection of LabVIEW for FPGA projects with templates supporting different applications like, analog acquisition, image acquisition from camera-link, digital I/O, etc. The templates are designed using a set of predefined and documented rules. All designs implemented with these rules will be usable by EPICS users.
- A library (IRIO Library) in charge of discovering the resources implemented by the user in the FPGA and providing functions to access to them. IRIO library needs the NI-RIO Linux Device Driver provided by the manufacturer. IRIO library supports data acquisition from FPGA-based devices to host memory and to GPU memory. This last functionality has been included modifying the NI-RIO Linux Device Driver and enable users to develop highly optimized image acquisition and processing applications using FPGA and GPUS.

All authors are with the Instrumentation and Applied Acoustic Research Group, Universidad Politécnica de Madrid, Campus Sur UPM, Madrid 28031 (telephone: +34-913364696, e-mail: mariano.ruiz@upm.es).

- An EPICS device support implemented with ASYN driver methodology [7]. The device support includes the templates with the EPICS records allowing the connection with the resources implemented in the FPGA. This device support is integrated with ITER SDD tools that automatizes the code generation of EPICS IOCs.
- An areaDetector [8] driver for image acquisition systems using FlexRIO and cameralink standard.
- A library with C++ classes to integrate RIO devices in ITER nominal device support [9] [10]. This allows the user the implementation of customized EPICS device support for a specific hardware implementation in the FPGA. The use of ITER nominal device support approach also allows to use other applications such as archiving.

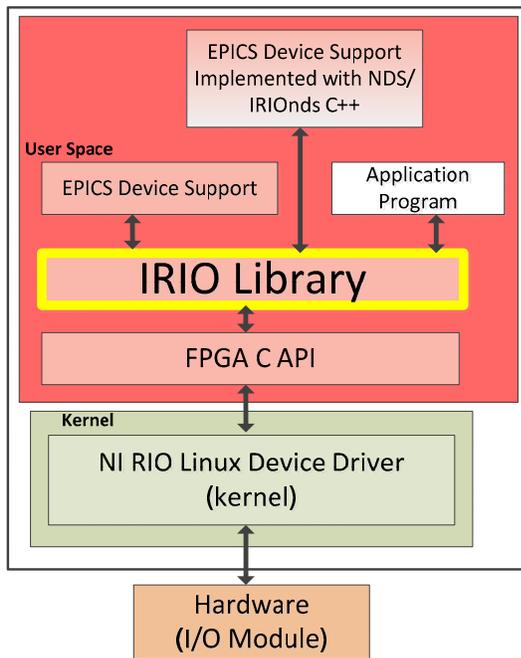


Fig. 2: Software layers in IRIO tools.

Using these tools the development cycle is summarized in Fig 3. The user develops the hardware application for the FPGA using the template and the LabVIEW development tools and later moves the bitfile obtained to a Linux platform in order to develop standalone applications using IRIO library of EPICs based applications with IRIO EPICS Device Support or NDS/AreaDetector approach. The details and features of the different modules are presented in the following paragraphs.

## II. IRIO LABVIEW TEMPLATES

Implementation of analog and image data acquisition and processing (IDAQ) using FPGAs can be very diverse depending in the final application [11] [12]. The multitude of possible FPGA implementations that the user can accomplish and how this is interfaced with the host complicate the management software implementation of the IDAQ. In order to simplify and standardize this interface a set of templates have

been implemented. All of them meet a set of predefined rules named design rules. IRIO project includes templates for LabVIEW/FPGA simplifying the implementation of analog, digital and image data acquisition systems –using CameraLink standard–.

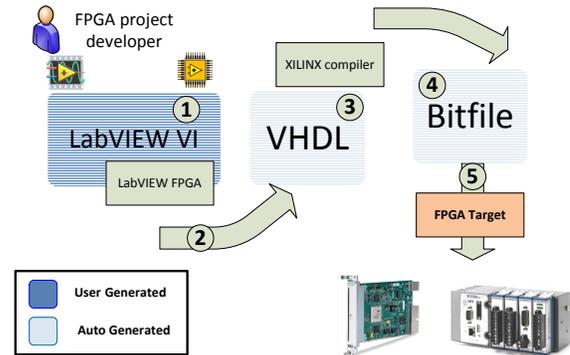


Fig. 3:

Once the user has implemented its application code taking the templates as starting point, the code is translated to VHDL, by LabVIEW/FPGA, and compiled by XILINX tools, to get the resulting bitfile (see Fig 3). LabVIEW/FPGA tools generate an output file containing not only the bitfile but a description of the I/O terminals in the main hierarchy of the design. This information is used by another software tools called “LabVIEW FPGA C API generator” in charge of generating a C language header file with a list of enumerated data types identifying (using a label) the resources implemented in the design. A library provided by the manufacturer is able to Access the physical resources using this enumerated data types. The “design-rules” have standardized terminal identification according to their functionality. Finally, indicate that any change in the hardware design will require a new compilation to obtain a new bitfile and thus a new compilation of the C application using the FPGA because the header file will be different.

## III. IRIO LIBRARY

IRIO library plays an essential role in the applications implementation. It detects the resources included by the developer in the FPGA following the design rules. IRIO Library allows accessing I/O elements implemented in the FPGA and managing streaming channels implemented with DMA. IRIO library is in top of the Linux device driver and its API provided by the manufacturer. IRIO classifies the different types of applications implemented in the FPGA by mean of “profiles” linked to cRIO, FlexRIO and R-Series hardware platforms. When using IRIO the first step is the initialization process calling `irio_initdriver` with bitfile and headerfile as main parameters. One important point here is that the header file is parsed by the library. IRIO library detects the hardware platform and the profile implemented. For each profile the library searches the terminals as the design rules define creating the data structures for later access using the API provided. If the user changes its FPGA design because there is an error in any of the functionalities implemented it will be necessary to recompile the design to obtain a new bitfile and header file without recompiling the C application. Figure 4 shows a simple

example of how to use the IRIO library in a C program. The program initializes the library identifying the resources, puts the FPGA in the running state and finally performs a data acquisition start and access to a digital port before releasing the resources.

```

irio_initdriver("test",
"S/N","Model","bitfile","version","pathtobi
tfile","pathtoheaderfile",&DrvPvt);
p_DrvPvt: "Data structure with all the
resources found".
irio_setFPGAstart(&DrvPvt, 1, &status)
//FPGA to run!!!
irio_setDAQStartStop(&DrvPvt, 1, &status)
//data acquisition starts!!
irio_setAuxDO(&DrvPvt, 1, &status)
//Boolean set to "1"
irio_close (&DrvPvt)

```

Fig. 4: Excerpt of a C program using IRIO Library

Table I summarizes the profiles implemented and supported in IRIO library. They are classified according to the platform. These profiles allow the implementation of analog/digital signal, and images using DMA, as well as analog and digital simple samples using I/O. There are two different versions for DMA movement, one for DMA to host computer memory and another for NVIDIA GPU memory.

TABLE I. PLATFORMS AND PROFILES

Platform	Profile	Functionality
cRIO	DMA Analog Data acquisition and I/O	I/O
R Series	DMA Analog Data acquisition and I/O	I/O
FlexRIO	DMA to HOST Analog/Digital and I/O	
FlexRIO	DMA to HOST Image Data acquisition and I/O	
FlexRIO	DMA to GPU Analog/Digital and I/O	
FlexRIO	DMA to GPU Image Data acquisition and I/O	

The final conclusion is that IRIO allows the identification and the access to all the resources implemented for the different profiles, therefore the user have to focus in the implementation of the final application. Figure 4 summarizes the concept of how IRIO maps the resources to high level applications.

#### IV. IRIO EPICS DEVICE SUPPORT

One of the most important objectives that can be achieved using IRIO software tools is the integration of these advanced data acquisition applications in EPICS if the user follow the design rules. All implemented resources are going to be detected and mapped as a set of EPICS PVs. The implementation of this functionality is based on the driver development mechanism provided by EPICS and known as EPICS device support. The most used mechanism nowadays is asynDriver. IRIO EPICS Device support has been implemented using asynDriver. The development of an EPICS IOC integrating a RIO device will be detailed later

The main features of IRIO EPICS device support implementation are the following:

- Templates with the description of the EPICS records, organized by functionality. This means that there are records for multifunction data acquisition (analog and digital input/output) and images. For example, in the first group, exists templates with records (PVs) managing the data flow for a set of channels acquired using the same sampling rate. IRIO EPICS device support has templates for all the profiles supported in IRIO.
- IRIO EPICS device driver uses records connected with asynlayer using the standard device support provide by this. All PVs are connected using the standard interfaces defined in asyndriver.
- Interfaces used for the different functionalities are; asynInt32, asynOctet, asynFloat64, asynFloat32Array, asynInt8Array..
- Waveform records with interrupt SCAN are supported for analog/digital data acquisition and images.
- Records for Analog and digital channels support EPICS periodic or interrupt scan method
- Identification and management of all resources implemented in the FPGA using IRIO library. The driver launches all the threads in charge of data acquisition and PVs publishing. User has not to take care of low level details when using this EPICS device driver.

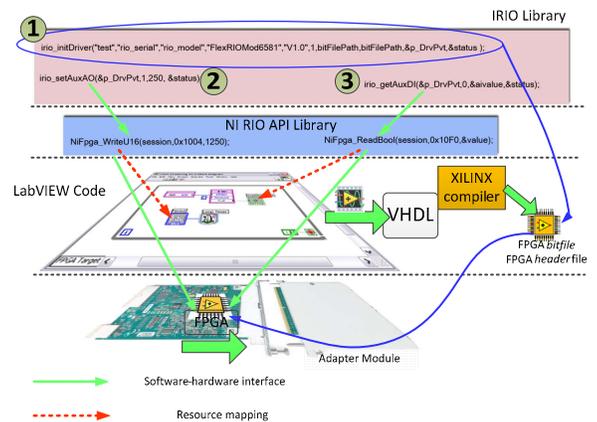


Fig. 5: Resource mapping using IRIO Library

The steps needed to use the IRIO EPICS device driver are described below:

- Considering that the user has implemented the FPGA part, it only need to add FPGHA resources to EPICS project. These are included adding the following sentences in the Makefile available in the EPICS app folder:

```

IRIO_FPGA_DIR = <folder>
IRIO_FPGA += <headerfile>.h
IRIO_FPGA += <bitfile name>

```

- Addition in the st.cmd file of the driver initialization calling the “nirioinit” function implemented and exported in the driver. This function requires as parameters the serial number of the RIO device to be used and the bitfile.

- Template load for the instantiation of the PVs. For instance:

```
dbLoadTemplate("sample-rio-module.substitution")
dbLoadTemplate("sample-rio-dataacquisition.substitution")
dbLoadTemplate("sample -rio-wf.substitution")
```

- Compilation and execution of the IOC.

### V. IRIO FOR AREADETECTOR AND NOMINAL DEVICE SUPPORT

EPICS provides a set of software tools for the development of control and monitoring distributed data acquisition and control systems, facilitating the standardization of heterogeneous hardware access using the process variables. Nevertheless, EPICS does not standardize the PV names nor the mechanism for hardware access although asynDriver is a very widespread solution for this. There are several initiatives to standardize this step: Nominal Device Support (NDS) and areaDetector. IRIO software tools includes both solutions, an areaDetector software driver for FlexRIO cameraink-based applications and, a library with the C++ classes for the development of EPICS device support using NDS as ITER has defined. AreaDetector provides, the naming convention and the records for image and 2D detector, and a mechanism for the execution of different algorithms with the image acquired called plugins (this mechanism includes also the buffer management and the shared memory method among the plugins). NDS provides something similar, a set of PVs defined for DAQ and image systems, PVS for timing and the option to trigger plugins for processing or streaming to data archiving applications. .

### VI. RESULTS AND USE CASES

Actually, the different IRIO software modules are in use by ITER CODAC CORE SYSTEM. IRIO is distributed with GPL V2 license. The different ITER domestic agencies have these tools available for the integration of the applications in the fast controller using FlexRIO/cRIO for the different plant systems. These applications are in development with the help of SDD and CSS [14] included in CCS [15]. This high level tool simplifies the creation of IOCs because it generates automatically the EPICS Project (or unit) configuring the most important files in the IOC. Of course this avoid errors for the user. Additionally, ITER is developing also applications for diagnostics using Nominal Device approach. Examples of these are the image acquisition systems and the fission chamber use case. These have customized EPICS device support adding functionalities like timing from an IEEE1588 device and data archiving using ITER DAN API. Obviously, EPICS is not able

to acquire information at high sampling rates and additional solutions are required to implement this functionality [17]. Other development using IRIO have done for ESS-Bilbao with the development using RIO devices for ion source control system.

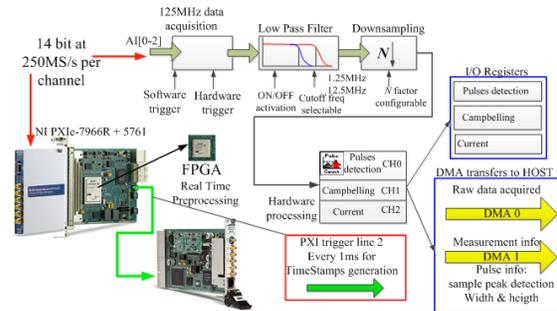


Fig. 6: Implementation of the pulse detection for the ITER fission chamber use case.

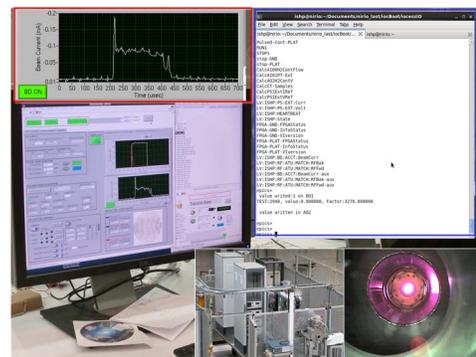


Fig. 7: EPICS IOCs implemented using RIO/FlexRIO using IRIO tools, working on a real case application for the ESS Bilbao ISHP

### REFERENCES

- [1] Abbasi, R.; Ackermann, M.; Adams, J. M. Ahlers; J. Ahrens, K Andeen, et al., "The IceCube data acquisition system: Signal capture, digitization, and timestamping," Nuclear Instruments & Methods in Physics Research section a-accelerators spectrometers detectors and associated equipment Volume: 601 Issue: 3 Pages: 294-316 Published: APR 1 2009.
- [2] M. Ruiz, J. Vega, R. Castro, D. Sanz, J.M. López, G. de Arcas, et al, "ITER Fast Plant System Controller prototype based on PXIe platform", Fusion Engineering and Design, Volume 87, Issue 12, December 2012, Pages 2030-2035, ISSN 0920-3796
- [3] R. Claus, "A new ATLAS muon CSC readout system with system on chip technology on ATCA platform", Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 824, 11 July 2016, Pages 272-274, ISSN 0168-9002
- [4] D. Sanz, M. Ruiz, R. Castro, J. Vega, J. M. Lopez, E. Barrera, N. Utzel and P. Makijarvi, "Implementation of intelligent data acquisition systems for fusion experiments using EPICS and FLEXRIO technology", IEEE Trans. Nucl. Sci., vol. 60, no. 5, pp. 3446-3453, 2013
- [5] Leo R. Dalesio, Jeffrey O. Hill, Martin Kraimer, Stephen Lewis, Douglas Murray, Stephan Hunt, et al. "The experimental physics and industrial control system architecture: past, present, and future". Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 352, Issue 1, 1994, Pages 179-184, ISSN 0168-9002
- [6] William Davis, Anders Wallander, Izuru Yonekawa, the IO-CT Control System Division, "Current status of ITER I&C system as integration begins," in Fusion engineering and Design, In Press, Corrected Proof, Available online 10 May 2016.
- [7] M.R. Kraimer, M. Rivers, E. Norum EPICS: ASYNCHRONOUS DRIVER SUPPORT. Proceedings of 10th ICALEPCS Int. Conf. on Accelerator & Large Expt. Physics Control System. 2005.

- [8] T. Madden, P. Jemian, S. Narayanan, A.R. Sandy, M. Sikorski, M. Sprung, et al, "Real-time compression of streaming X-ray photon correlation spectroscopy area-detector data". Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 649, Issue 1, 1 September 2011, Pages 237-239, ISSN 0168-9002
- [9] S. Simrock. "Standardization of EPICS Device Support for Data Acquisition". Proceedings of the Sixth Low-Level Radio Frequency Workshop 2013.
- [10] V.A. Isaev, N. Claesson, M. Plesko, K. Zagar. "EPICS Device Support". Proceedings of ICALEPS 2013. San Francisco. CA. USA
- [11] W. Zheng, R. Liu, M. Zhang, G. Zhuang, T. Yuan, "Design of FPGA based high-speed data acquisition and real-time data processing system on J-TEXT tokamak", Fusion Engineering and Design, Volume 89, Issue 5, May 2014, Pages 698-701, ISSN 0920-3796.
- [12] F. Carena, W. Carena, S. Chapeland, et al, "The ALICE data acquisition system", Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 741, 21 March 2014, Pages 130-162, ISSN 0168-9002, <http://dx.doi.org/10.1016/j.nima.2013.12.015>
- [13] Diego Sanz, Mariano Ruiz, Mikel Eguiraun, Iñigo Arredondo, Inari Badillo, Josu Jugo et al, "RIO EPICS device support application case study on an ion source control system (ISHP)", Fusion Engineering and Design, Volumes 96–97, October 2015, Pages 927-931, ISSN 0920-3796, <http://dx.doi.org/10.1016/j.fusengdes.2015.02.040>.
- [14] L. Abadie, K. Bandaru, G. Darcourt, H. Deshmukh, F. Di. Maio, A. Mariage. et al. SDD toolkit. "ITER CODAC platform for configuration and development" Proceedings of ICALEPS 2013. San Francisco. CA. USA
- [15] F. Di Maio, L. Abadie, C. Kim, K. Mahajan, P. Makijarvi, D. Stepanov, et al. THE CODAC SOFTWARE DISTRIBUTION FOR THE ITER PLANT SYSTEMS. Proceedings of ICALEPS 2011. Grenoble. France.
- [16] D. Sanz, M. Ruiz, R. Castro, J. Vega, M. Afif, M. Monroe et al. "Advanced Data Acquisition System Implementation for the ITER Neutron Diagnostic Use Case Using EPICS and FlexRIO Technology on a PXIe Platform". IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 63, NO. 2, APRIL 2016
- [17] R. Castro, L. Abadie, Y. Makushok, M. Ruiz, D. Sanz, J. Vega, et al., "Data archiving system implementation in ITER's CODAC Core System", Fusion Engineering and Design, Volumes 96–97, October 2015, Pages 751-755, ISSN 0920-3796
- [18] S. Esquemabri, M. Ruiz, E. Barrera, D. Sanz, A. Bustos, R. Castro et al "Implementation of an image acquisition and processing system based on FlexRIO, CameraLink and areaDetector". Fusion Engineering and Design to be published.