ISBN: 978-3-95450-262-2 ISSN: 2673-5350 doi: 10.18429/JACoW-IBIC2025-MOPCO18

SOFTWARE UPGRADE OF BEAM DIAGNOSTICS READOUT SYSTEM BASED ON PXIe HARDWARE*

Z. H. Xu[1,2], F. Li[1,2], W. L. Huang[1,2], L. Zeng[1,2], R. Qiu[1,2],
M. A. Rehman[1,2], B. Zhang[1,2], R. J. Yang†,[1,2]
[1]Institute of High Energy Physics, Chinese Academy of Sciences (CAS), Beijing, China
[2]China Neutron Spallation Source, Dongguan, China

Abstract

In the beam diagnostics system of the China Spallation Neutron Source accelerator, multiple National Instruments PXIe multifunction DAQ modules were utilized for readout system development. The original software architecture, implemented with LabVIEW and DSC modules on the Windows system, introduced substantial challenges in EPICS integration, including closed-source limitations, high licensing costs, and suboptimal real-time performance. This paper details a comprehensive software upgrade methodology that preserves the existing NI PXIe hardware infrastructure. The upgraded system implements standard EPICS Input/Output Controllers developed in C language under the Linux system, integrating signal acquisition and front-end electronics control functionalities within EPICS IOC. By developing a Linux kernel-level driver, and implementing real-time optimizations, the new architecture achieves significant performance improvements, including reduced latency and higher sampling rates. This re-engineering approach enhances the readout system's stability, reliability, and flexibility of EPICS data interaction, providing a viable open-source alternative to the traditional LabVIEW/Windows stack for accelerator instrumentation systems.

INTRODUCTION

The beam diagnostics system at CSNS relies heavily on NI PXIe hardware for precise signal acquisition and processing. The original readout software was built on the LabVIEW and Windows platform, utilizing NI-DSC modules for limited EPICS integration. While LabVIEW offers rapid graphical development, its closed nature, coupled with the Windows operating system's limitations in real-time and distributed control scenarios, posed challenges for long-term maintenance, scalability, and integration within the broader EPICS-based control system framework at CSNS and future projects like CSNS-II. These challenges necessitated a software upgrade that retained the proven PXIe hardware investment while migrating to a more open, robust, and performant software ecosystem based on Linux and native EPICS.

*Work Supported by National Natural Science Foundation of China (NO. 12305166) and the Natural Science Foundation of Guangdong Province, China, (NO. 2024A1515010016)
†yangrenjun@ihep.ac.cn

OLD SOFTWARE ARCHITECTURE

In the Beam Diagnostics System of the China Spallation Neutron Source [1], there are over a dozen PXI-based data acquisition hardware systems, including those for beam current measurement, beam position measurement, and beam profile measurement, among others.

The specific distribution is shown in Fig. 1.

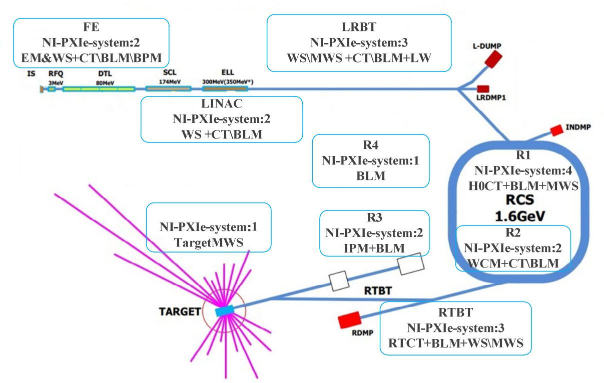

Figure 1: Distribution of PXI beam diagnostics equipment.

Software architecture is based on the PXI platform and operating on the Windows system, the data acquisition software architecture is developed using LabVIEW. The system performs functions such as analog input signal reading and digital I/O control. After data acquisition is completed, data sharing and Process Variable publishing are achieved through the NI-DSC (Data Acquisition and Control) module integrated with EPICS functionality [2]. The Control System Studio Operator Interface serves as the client for monitoring and operation.The software architecture diagram is shown in the Fig. 2.

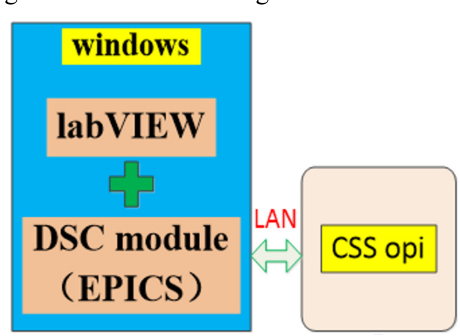

Figure 2: Software architecture.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-262-2 ISSN: 2673-5350 doi: 10.18429/JACoW-IBIC2025-MOPCO18

© Content from this work may be used under the terms of the CC BY 4.0 licence (© 2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

SOFTWARE UPGRADE METHODOLOGY

The software upgrade will be carried out in two phases. The first phase is to replace the NI-DSC module. Involves replacing the NI-DSC module with a standard soft IOC to handle data publishing and sharing. The data acquisition and control software will continue to be developed in LabVIEW on Windows. MSYS2 will be installed on Windows to emulate a Linux environment, within which EPICS Base and necessary support modules will be installed to set up a standard EPICS IOC and deploy the required PVs. The calab software will then be used in LabVIEW to establish a connection with the IOC, enabling data binding between LabVIEW controls and the IOC process variables. The first phase software architecture is shown in the Fig. 3.

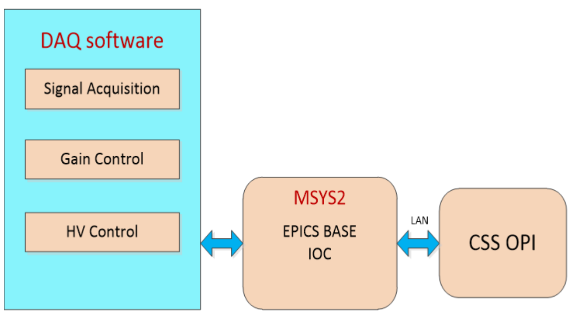

Figure 3: The first phase software architecture.

The second phase is to install EPICS base completely on the linux system and develop a standard hard ioc to implement data acquisition and control functions.

The core objective of the upgrade was to replace the LabVIEW/Windows software layer while fully utilizing the existing NI PXIe hardware (e.g., PXIe-6363 modules, PXIe-1085 chassis). The new software architecture is built on three main pillars: a native Linux driver for NI PXIe hardware, the EPICS framework for control and data distribution, and real-time optimizations for deterministic performance.The second software system architecture is shown in Fig. 4.

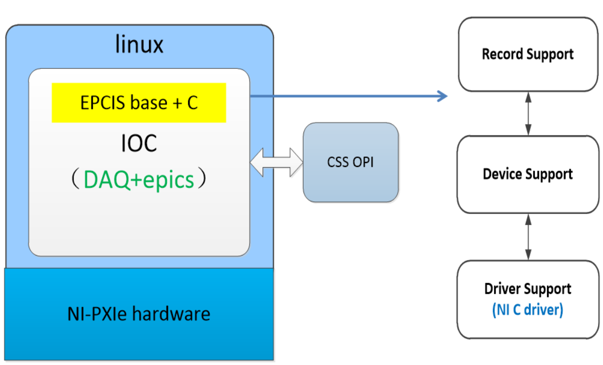

Figure 4: The second phase software architecture.

Linux Driver Install

The crucial first step is to develop or use the driver provided by NI that is compatible with the Linux kernel to interface directly with NI PXIe hardware, because the official labview driver cannot be directly used for Linux, and the device support of EPICS ioc is developed in c language. Therefore, it is necessary to install the C language driver for the device on linux and ensure that it can be used normally.

To use ni-daqmx on Linux for C development, first add NI's repository and install the ni-daqmx and ni-daqmx-devel packages. This driver supports modern PXI, USB. It enables analog/digital I/O, counter/timer operations, and multi-device synchronization through a high-performance, task-based API. The key advantage is a stable, cross-platform C API that is identical to Windows, allowing for easy code porting and integration into the Linux system for embedded, or high-performance data acquisition systems [3].

The key advantages of leveraging ni-daqmx on a Linux platform are its robust performance, seamless ecosystem integration, and significant cost efficiency. It delivers a high-performance, stable environment ideal for demanding, low-latency acquisition. A major benefit is its cross-platform API consistency, which allows C code to be ported effortlessly between Windows, Linux, and real-time targets, drastically reducing development time. Linux itself enables reliable, headless operation for embedded systems and unlocks the power of a free, open-source toolchain and scripting ecosystems. Offering a highly cost-effective solution for scalable deployments.

EPICS IOC Development

The development of an EPICS ioc for National Instruments PXI hardware involves creating a structured software architecture that seamlessly integrates the deterministic, high-performance acquisition capabilities of the PXI platform with the standardized network-based control and monitoring provided by EPICS. The primary focus is on embedding the data acquisition logic directly within the ioc process, ensuring low latency and direct hardware control [4].

The implementation can be logically divided into three critical layers:

Driver Support is the foundation, providing a direct interface to the PXI hardware.

ni-daqmx API: This layer is built exclusively upon the ni-daqmx C API. It is responsible for all low-level communication with the DAQ card.

Hardware Configuration: It encapsulates functions to create, configure, and manage DAQmx tasks. This includes setting up sampling rates, voltage ranges, triggers, and clock sources.

Data Buffer Management: It implements the routines for initializing data buffers, handling continuous or finite acquisitions, and managing the flow of data from the hardware to the application. This often involves sophisticated techniques like circular buffering.

Callback Functions: For continuous operations, it sets up DAQmx callback functions to receive notifications from the driver when new data is available. This is crucial for real-time performance [5].

The architecture of DAQ and EPICS ioc is shown in Fig. 5.

ISBN: 978-3-95450-262-2 ISSN: 2673-5350 doi: 10.18429/JACoW-IBIC2025-MOPCO18

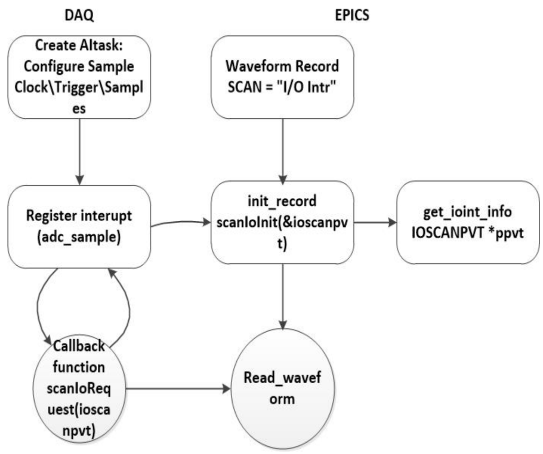

Figure 5: The architecture of DAQ and EPICS IOC.

SOFTWARE TEST

In the wire scanner profile measurement system of the CSNS-II injection region, a second-stage software architecture has been adopted, namely a Linux-based system with dedicated drivers and a hard real-time EPICS IOC, to achieve beam profile monitoring. The injection region is equipped with three wire scanners, comprising a total of 136 wires, while also requiring control over the gain switching of the front-end analog electronics. Accordingly, the data acquisition system is constructed using an NI-PXIe chassis housing a PXI controller and NI-PXIe-6358 multifunction cards, as illustrated in Fig. 6.

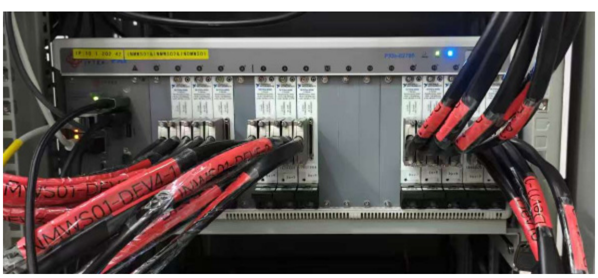

Figure 6: Hardware of the acquisition system.

The acquisition software must perform several key functions, including sampling of 136 signal channels, control of the front-end analog electronics gain stages, and real-time data processing—all implemented within the IOC. A schematic of the front-end electronics is provided in Fig. 7.

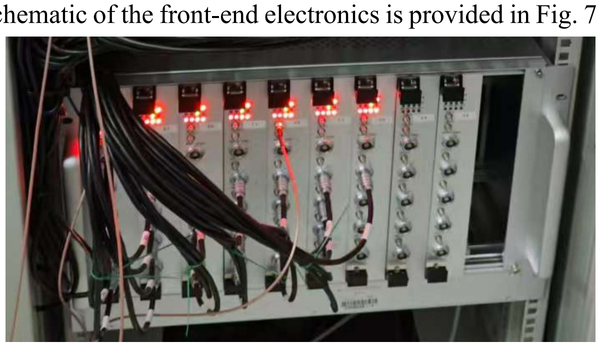

Figure 7: Front-end analog electronics.

After development, the IOC was deployed and tested in a laboratory setting. Signal generators were used to inject

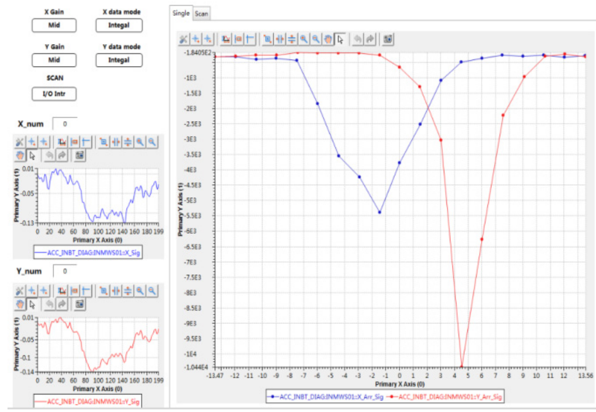

Figure 8: CSS OPI of wire scanner profile measurement system.

test signals individually into each channel of the front-end analog electronics. The IOC operated as intended, and the observed signals on the Control System Studio (CSS) Operator Interface (OPI) met all functional requirements. The CSS OPI is shown in Fig. 8.

CONCLUSION

As a critical component of the beam measurement system, the readout system is responsible for the acquisition and processing of detector signals as well as the control of front-end analog electronics. The stability of its software is particularly important. The legacy software, implemented with LabVIEW and the DSC module on the Windows system, could perform signal acquisition and data publishing. However, the Windows platform exhibited generally poor stability, often leading to system blue screens or reboots during long-term operation. Moreover, the PV publishing function realized by the DSC module was relatively closed, making it difficult to identify the root cause when PV failures occurred. Additionally, its IOC did not comply with the standard EPICS architecture, preventing seamless integration with other systems. In contrast, the new IOC developed based on Linux and EPICS base has completely resolved these issues. The high stability, open-source advantages, and rapid response of the Linux system are more conducive to readout system development. Furthermore, it enables seamless integration with the control system, significantly improving data publishing, control, and sharing capabilities.

REFERENCES

[1] W. Sheng *et al.*, "Introduction to the overall physics design of CSNS accelerators", *Chin. Phys. C*, vol. 33, pp. 1-3, 2009. doi:10.1088/16741137/33/S2/001

[2] Y. Enomoto, K. Furukawa, T. Natsui, H. Saotome, and M. Satoh, "Pulsed Magnet Control System Using COTS PXIe Devices and LabVIEW", in *Proc. ICALEPCS'19*, New York, NY, USA, 2020, pp. 946–949. doi:10.18429/JACOW-ICALEPCS2019-WECPR05

[3] D. Wang and M. Satoh, "The Upgrade of Pulsed Magnet Control System Using PXIe Devices at KEK LINAC", in *Proc. ICALEPCS'23*, Cape Town, South Africa, Oct. 2023,

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-262-2 ISSN: 2673-5350 doi: 10.18429/JACoW-IBIC2025-MOPCO18

pp. 635-638.
doi:10.18429/JACoW-ICALEPCS2023-TUPDP048

[4] EPICS, https://epics.anl.gov/

[5] National Instruments, https://ni.com/

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2024). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.