LabVIEW EPICS PROGRAM FOR MEASURING BINP HLS OF PAL-XFEL*

Hyojin Choi†, Kwang Won Seo, Youngjin Suh, Kyehwan Gil, Seung Hwan Kim, Heung-Sik Kang,
Department of Accelerator, PAL-XFEL, Pohang, Korea

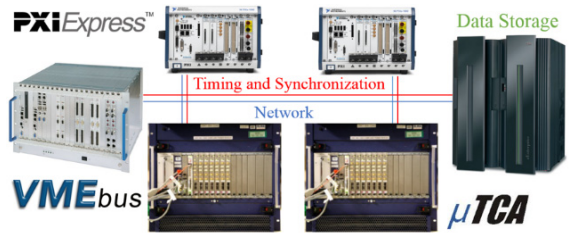
Abstract

In PAL-XFEL, a 4th generation light source, the HLS (Ultrasonic-type Hydrostatic Levelling System) developed at BINP (Budker Institute of Nuclear Physics) in Russia was installed and operated in all parts of PAL-XFEL in order to maintain observations of the vertical change building floor by the ground sinking and uplifting. For this, a HLS measuring program was written using NI LabVIEW and an EPICS IOC Server was built using the CA Lab which has been developed at BESSY (Berlin Electron Storage Ring Society for Synchrotron Radiation) in Germany. The CA Lab was improved and verified in order to confirm that it could support EPICS BASE libraries V3.14.12, and EPICS CA Client and that the EPICS IOC Server could be easily constructed by CA Lab in a 64-bit LabVIEW. This made Multi-core CPU (Multi-core Processor / Multi-thread Program) resource of 64bit Computer System (64bit Hardware PC, 64bit Windows OS, 64bit LabVIEW Multi-thread Programming) to be 100 percent utilized. This study proposes a configuration process for the HLS measuring program algorithm and a building process for the EPICS IOC Server by using CA Lab.

INTRODUCTION

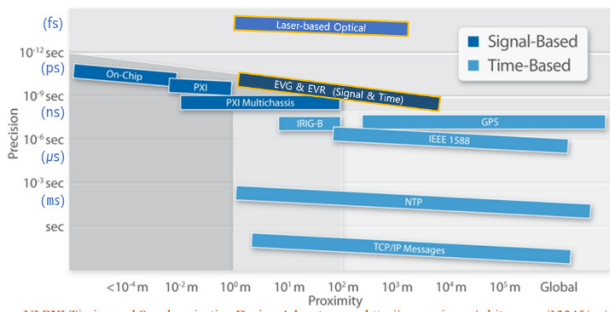
As devices created due to big science and big technology have been placed far away from control systems in terms of space and geography, advanced technology, including that for a distributed control system has become necessary (7.7 Control and Timing [1]). Various dispersed control platforms have been installed for PAL-XFEL like Figure 1. Owing to recent rapid developments in information technology (IT), such as advanced robot technology, Internet of things (IoT), big data and artificial intelligence, it is easy to establish such a new system through buying a distributed control system device developed by diverse companies. Engineers in many accelerator research institutes around the world have been engaged in efforts to advance effective technological developments by sharing technology through the open platforms of control systems and combining their experience and knowledge [2].

As shown in Figure 2, in order to synchronize an ordinary distributed control system, the time-based (time-stamp) synchronization of the entire big scientific device should be effected, first of all. After this, the signal-based synchronization of a module inside each device or the device should be achieved. Regarding RF reference of FEL and pump-probe experimental devices of beamline, a jitter and drift need a synchronization system of the sub-10 rms femtosecond level [3], and a laser-based optical synchronization system for the highest precision of synchronization has been developed to be applied to them (7.8 Femtosecond precision optical synchronization [1])[4].



① Establish a distributed control system.
② Design a high-performance and high-bandwidth network system suitable for a distributed control system and establish it.
③ Install a storage and backup system that can perform high-speed processing of large data in order to store large measurement control data without data loss.
④ Install a timing and synchronization system to get synchronized measurement data.

Figure 1: Construction of a distributed control system.



NI PXI Timing and Synchronization Design Advantages http://www.ni.com/white-paper/13345/en/

Figure 2: Signal-based and time-based synchronization architecture performance.

Timing and synchronization technologies correlate events in time, which is necessary to perform coordinated activities. For software to orchestrate these coordinated activities, the program needs to be synchronized and include a concept of time. As shown in Figure 3, hardware and application programs can be accurately synchronized only when the timing structure is within all systems.

In order to operate big scientific devices stably for a long time, measurement control devices that operate individually also should be highly reliable and available to be operated for a long time without problems occurring. For this purpose, stable and reliable hardware platforms and operating systems (OS) should be selected.

*Work supported by Ministry of the Science, ICT and Future Planning
†choihyo@postech.ac.kr

ISBN 978-3-95450-148-9

With regard to the persons taking charge of these devices, it is important for them to perform the application programming in an effective and rational manner. Only when they understand the principles and concepts of the physical, chemical, instrumental and electric movements of controlled systems, can they think of the best algorithms and write effective (optimum) application programs.

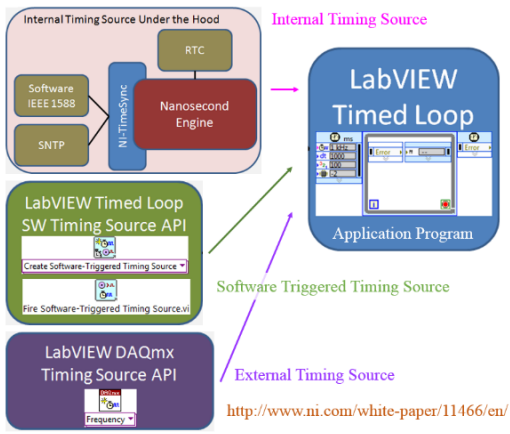


Figure 3: Timing and synchronization in NI LabVIEW.

NI PXI PLATFORM

PCI eXtensions for Instrumentation (PXI) is based on industry-standard computer buses and permits flexibility in building equipment. Since it was developed in 1997, PXI has been managed by PXI Systems Alliance (PXISA). Regarding PXI Architecture, refer to the PXI Hardware Specification file issued by PXISA [5]. With regard to PXI, various kinds of high-performance and input-output modules have been produced, so people can choose the right modules for their purposes. If fast feedback control is necessary, a real-time system should be established with NI FlexRIO custom instruments using FPGA. The real-time system using FPGA should be remotely connected to a network and replaced with an improved algorithm. Figure 4 show the NI PXI platform.

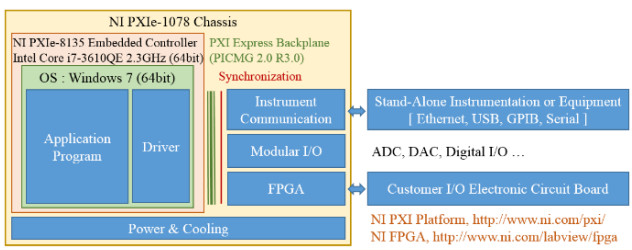


Figure 4: Construction of a PXI platform.

NI PXI Multi-core Controller

Figure 5 shows the NI PXIe-8135 block diagram and architecture of Intel Core i7. A programmer should review the specifications of hardware programs and controllers in use and then write the best and most effective program algorithm for the purposes at hand. The Intel Core i7-3610QE processor, a multi-core processor, applied to the PXIe-8135 controller has four cores and each core can make two threads work [6]. As shown in Figure 6, operational states of a multi-core processor can be checked through the Windows task manager.

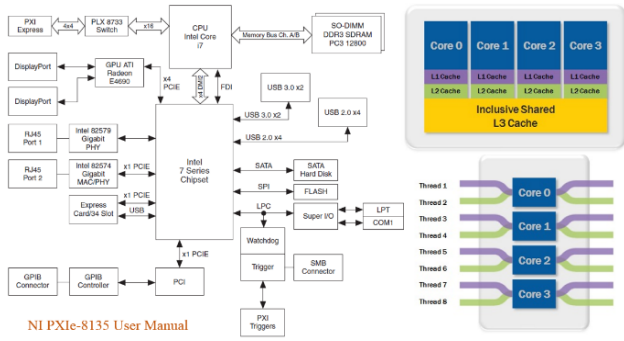


Figure 5: NI PXIe-8135 block diagram and Intel Core i7 architecture.

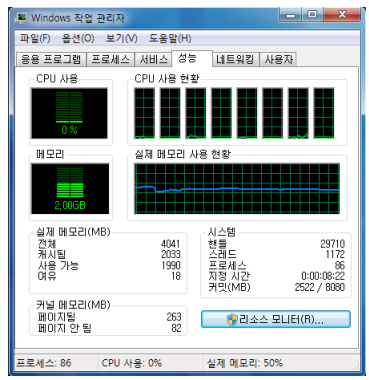


Figure 6: Windows task manager.

Multi-core Programming with LabVIEW

Many programmers are enthusiastic about the performance of multi-core but they sometimes can't utilize it properly. As shown in Figure 7, applications written with LabVIEW can choose a processor of multi-core in a simple way. There are various program languages that we can choose and each program language has its strong points and weak points as well as distinguishing features. The advantages of the LabVIEW program are that it is convenient and immediate. Anyone can learn about it quickly and easily control various kinds of diagnostic control hardware systems [7].

ISBN 978-3-95450-148-9

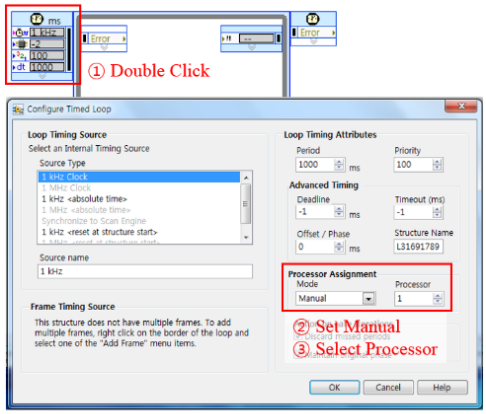


Figure 7: How to choose multi-core processors.

BESSY CA LAB

There are diverse ways to materialize EPICS communication in LabVIEW, as shown in Figure 8. CA Lab that satisfies the conditions of the EPICS BASE library (V3.14.12) and can be used in LabVIEW was developed and distributed by BEESY in Germany. Programmers can download suitable Ca Lab for their LabVIEW version (32bit or 64bit) and OS type (Windows or Linux) and use it. Ca Lab is a 64bit EPICS library that can use functions of 64bit CPU, 64bit OS and 64bit LabVIEW perfectly and BESSY Ca Lab provides ways to use it and a file of examples. Figure 9 shows the concept of movements of Ca Lab [8].

- LabVIEW® DIM Interface ⇔ EPICS - DIM Interface
- LabVIEW® Shared Memory Interface to EPICS IOC by SNS
- LabVIEW® ActiveX CA by Kay Uwe Kasimir, ORNL
- LabVIEW® Data Logging and Supervisory Control Module, NI
- LabVIEW® native EPICS implementation by SNS
- LabVIEW® CA Lab Channel Access implementation by BESSY

https://wiki.gsi.de/Epics/ConnectingLabVIEWandEPICS

Figure 8: Connecting LabVIEW and EPICS.

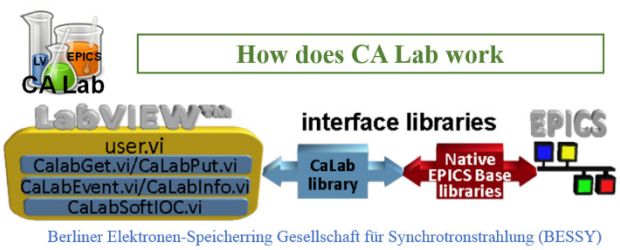


Figure 9: Concept of Ca Lab movements.

HLS IOC SERVER

Figure 10 shows the constitution of IOC server programs using PXI. A program bringing measurement data of HLS connected to a private network periodically reads the measurement of HLS with a time interval. Measurements of HLS are recorded in the shared memory which sends a wakeup signal to the analysis & display program after reading all HLS measurements. The analysis & display program that receives the wakeup signal then analyzes HLS measurements recorded in the shared memory, and displays the analysis result to the monitor and enters a wait condition, a sleep mode. When the Ca Lab IOC server program connected to the IOC network gets a request from data storage to store HLS data, the program sends HLS measurements data in the shared memory to data storage.

As shown in Figure 4, the NI PXIe-8135 controller has two Ethernet ports. IOC EPICS communication was connected to Port 2 (Intel82574) exclusively for servers and private communication for BINP HLS measurement was connected to Port 1 (Intel82579).

BINP HLS TEST ON PAL-XFEL

Figure 11 shows the method of using BINP HLS and the result of testing ULSE 2 sets which was borrowed from BINP to learn about the operation [9].

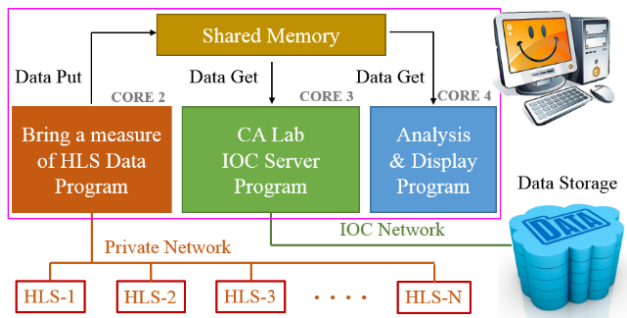


Figure 10: Multi-processor program.

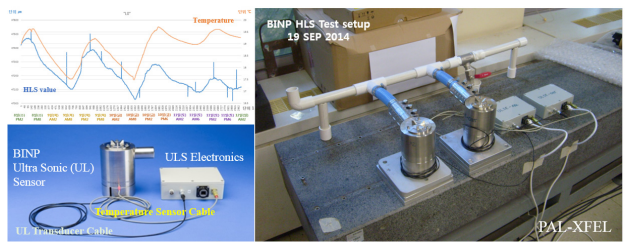


Figure 11: BINP HLS test on PAL-XFEL.

REFERENCES

[1] A.W. Chao, et al., "Handbook of Accelerator Physics and Engineering", 2nd Edition, Wold Scientific Pub Co Inc.

[2] D. Gurd, et al., "Accelerator Control and Global Networks – State of the Art", proc. of LINAC2004, FR201 (2004).

[3] S. Schulz, et al., "Femtosecond all-optical synchronization of an X-ray free-electron laser", proc. of Nature Communications (2015).

[4] S. Hunziker, et al., "Reference Distribution and Synchronization System for SwissFEL: Concept and First Results", proc. of IBIC2014, MOCZB2 (2014).

ISBN 978-3-95450-148-9

[5] PXI System Alliance (PXISA), http://www.pxisa.org/
[6] Top Eight Features of the Intel Core i7 Processors for Test, Measurement, and Control, http://www.ni.com/white-paper/11266/en/
[7] Advantage of Using LabVIEW in Academic Research, http://www,ni.com/white-paper/8534/en/
[8] BESSY CA Lab, http://www-csr.bessy.de/control/SoftDist/CA_Lab/
[9] H. Choi, et al., "The BINP HLS to Measure Vertical Changes on PAL-XFEL Buildings and Ground", proc. of FEL2015, MOP047 (2015).

ISBN 978-3-95450-148-9