

THE UPGRADE OF PULSED MAGNET CONTROL SYSTEM USING PXIe DEVICES AT KEK LINAC

D. Wang*, M. Satoh
KEK, Ibaraki 305-0801, Japan

Abstract

The pulsed magnet control system (PMCS) at KEK electron positron injector LINAC operates at every 20 ms to achieve simultaneous injection for four rings, a 2.5 GeV Photon Factory (PF), a 6.5 GeV PF-AR, a 4 GeV SuperKEKB lower energy ring (LER) and a 7 GeV SuperKEKB high energy ring (HER). The system consists of a control server that operates on the Windows 8.1 platform, in conjunction with a PXIe chassis equipped with a DAC, an ADC, and an event timing module. The PXIe DAC board responds to the trigger signal which is generated from the event receiver and sets the current of the pulsed magnet. The readback value of the magnet current is retrieved from the ADC module. Despite its long success, the PMCS has several problems in use. One is the discontinued support of Windows 8.1. Another key concern is the unsatisfactory long-term stability. To solve the problems, an upgraded system using real-time Linux to communicate with PXIe modules is adopted. The EPICS driver for PXIe devices is developed to integrate with the LINAC control system. The development of the new Linux-based PMCS is introduced in this work.

INTRODUCTION

The injector LINAC at KEK is responsible for the injection of 4 target rings which consist of a 7 GeV electron high energy ring (HER), a 4 GeV positron low energy ring (LER), a 2.5 GeV Photon Factory (PF) and a 6.5 GeV PF-AR ring, as shown in Fig. 1. LINAC is responsible for performing a simultaneous top-up injections into four target storage rings and a DR using the method called pulse-to-pulse modulation (PPM) [1].

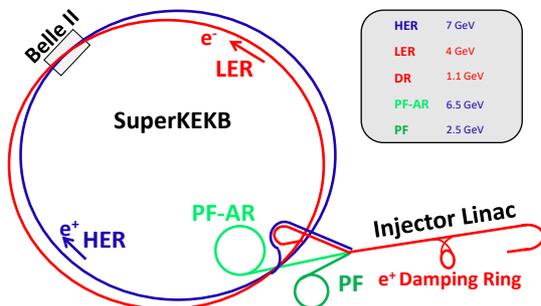


Figure 1: A schematic view of LINAC, SuperKEKB, and PF/PF-AR.

In order to meet the PPM requirements, a total of 16 pulsed magnet control units have been deployed across the 600-meter LINAC since 2017. Upon receiving a specific event

code, indicative of a particular beam mode, these control units are activated to adjust the magnet current. By using this configuration, the magnetic field can undergo pulse-to-pulse modifications within 20 ms. This rapid response ensures that the beam profile is optimized to the requirements of each destination ring.

PULSED MAGNET CONTROL SYSTEM

Hardware

Figure 2 shows the rack of one pulsed magnet control unit. Each unit consists of a homemade server and a National Instruments (NI) PXIe-1082 chassis fitted with four modules, a controller control module (NI PXIe-8381), an event receiver (EVR) board (MRF PXI-EVR-230), a DAC board (NI PXI-6733), and an ADC board (NI PXIe-6356) [2, 3].

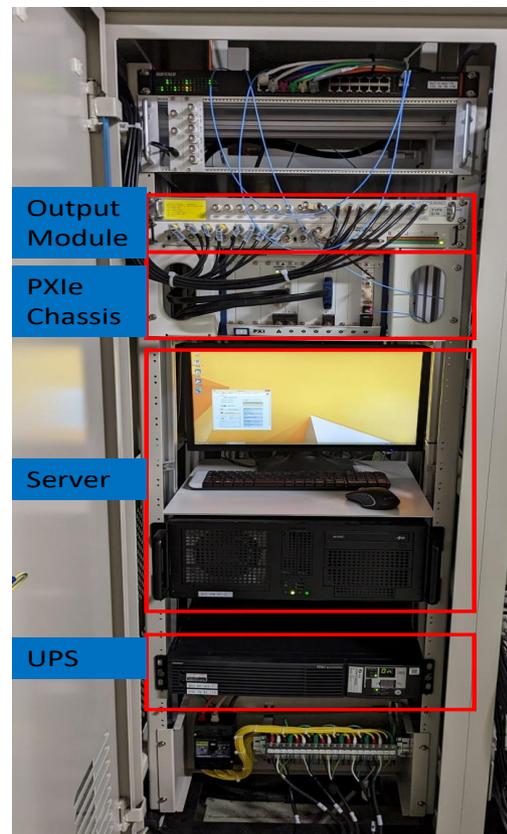


Figure 2: Rack of a pulsed magnet control unit.

The established units facilitates the independent control and monitoring of the output currents for as many as 8 power supplies. It offers a resolution of 16 bits and operates with a sampling rate of 1 MSa/s. To counteract potential instabilities that may arise from power disruptions or external

* di.wang@kek.jp

signal interference, an Uninterruptible Power Supply (UPS) system is integrated. This ensures the provision of continuous power. All components of this system are compactly accommodated within a standardized rack.

As depicted previously, every individual unit is capable of managing up to 8 pulsed magnets. In the context of beam operation, each of these pulsed magnets can respond to 12 unique beam modes. The resultant magnet current value is determined upon the specific injection beam mode in use.

The trigger signal comes from the event timing system. The event generator (EVG) sends pulse magnet trigger signal to all downstream event receivers (EVRs) every 20 ms. Besides that, the data buffer functionality inherent to the event timing system is utilized to transmit extra information which includes beam mode, timestamp and pulse ID.

Software

On the software front, as shown in Fig. 3, a LabVIEW application governs the process of data acquisition and provides connectivity through EPICS Channel Access (CA) [4]. The source code of LabVIEW is managed by git and can be easily dispatched to all 16 control units.



Figure 3: A LabVIEW programs runs on a Windows 8.1 server.

Due to a variety of reasons, a resolution has been reached to enhance the current system through an upgrading process. Various justifications for this decision are enumerated as follows.

- Windows 8.1 is discontinued.
- The trigger dropping rate is high [4].
- System restart is required occasionally [4].
- The compatibility and performance issue of NI network shared variable.

NEW SYSTEM

PXIe Modules Under Linux

Figure 4 shows the structure of the new PMCS. The initial step of the upgrade process involves the preparation of the

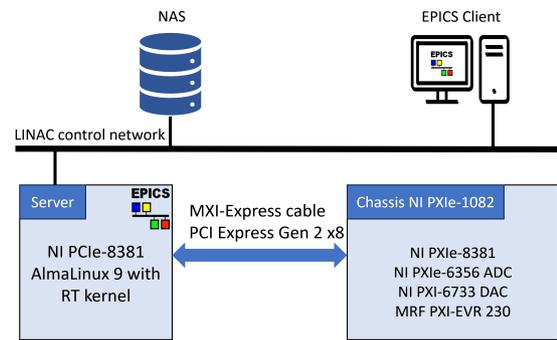


Figure 4: Diagram of the PMCS.

AlmaLinux distribution, after which an evaluation of the viability of the operation is conducted through the testing of drivers and access libraries.

In the case of NI modules, the server establishes a connection with the chassis using an NI PCIe-8381 module via an MXI-Express cable, which can transmit at a rate of 40 Gbps in each direction simultaneously. The Linux kernel driver is provided within the NI-DAQmx package. This package offers comprehensive assistance for devices oriented towards data acquisition and signal conditioning. The data acquired via the ADC can be effectively stored within an EPICS waveform record, accomplished through the utilization of the callback function afforded by NI-DAQmx.

The mrfioc2 module from the EPICS community is used as a low-level driver of the PXI EVR module [5]. Additionally, it is noteworthy that a minor modification of the mrfioc2 module is necessary to facilitate the reception of the data buffer from the EVR, owing to a compatibility issue related to CPU endianness.

Trigger System

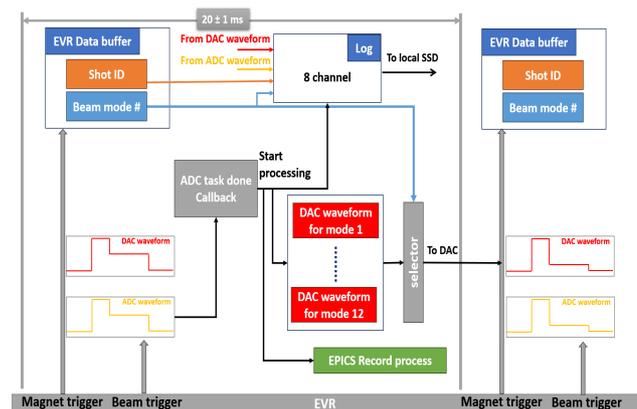


Figure 5: The data flow and trigger system.

As shown in Fig. 5, every 20 ms, the PMCS is triggered by a magnet trigger event. The current beam mode and shot ID are extracted from the data buffer of EVR. Meanwhile, the DAC outputs a waveform spanning a duration of 4 ms, directed towards the pulse driver. Upon completion

of waveform acquisition by the ADC, an interrupt is generated, subsequently prompting the initiation of the logging system to store data originating from 8 distinct channels. Concurrently, the output waveform of the subsequent pulse is determined by the specific shot ID. The callback function also triggers the initiation of the record processing procedure. Finally, after writing a new waveform of the next pulse to the memory, the DAC enters a standby state and prepares for the forthcoming trigger event.

The synchronization mechanism between the DAC and ADC is established by configuring the trigger mode to operate in an external trigger. In this setup, the DAC is triggered through an external signal. The ADC performs a continuous process of data sampling, although the generation of the sample clock is synchronized to the arrival of the external trigger signal.

Real Time Configuration

```
[Wed Jul 12 11:25:59 2023] epics> mcoreThreadShowAll
[Wed Jul 12 11:26:10 2023]
[Wed Jul 12 11:26:10 2023]      NAME      EPICS ID      LWP ID      OSIPRI      OSSPRI      STATE      POLICY      CRUSPT
[Wed Jul 12 11:26:10 2023]      _main_      0x1d32060      7470      0      0      OK      ? ? ?
[Wed Jul 12 11:26:10 2023]      errLog      0x1045ae8      7472      10      10      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      EPICIFIFO   0x104d0b0      7473      90      89      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      PCII8R0000:0f:0c  0x1e1f300      7474      98      97      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      pmLog       0x209ac00      7479      10      10      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      taskWd     0x25f1400      7480      10      10      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      timerQueue  0x262cd0e      7481      70      69      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      cbLow      0x20660f0      7482      59      58      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      cbMedium   0x2614d10      7483      64      63      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      cbHigh     0x2615000      7484      71      70      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      dbCalInk   0x2615410      7485      50      50      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      timerQueue  0x2611c10      7486      58      57      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      scanInce   0x2944e50      7491      67      66      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      scan-10    0x2952a00      7492      65      64      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      scan-5     0x2952d00      7493      66      65      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      scan-2     0x2952f50      7494      67      66      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      scan-1     0x29531a0      7495      68      67      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      scan-0.5   0x29533f0      7496      69      68      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      scan-0.2   0x2953640      7497      70      69      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      scan-0.1   0x2953890      7498      71      70      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      CAS-TCP    0x2953c30      7499      16      16      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      CAS-UDP    0x295e640      7500      12      12      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      CAS-beacon 0x295e990      7501      14      14      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      ipToAscIProxy 0x7f88bc0f050  7502      10      10      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      timerQueue 0x7f88bc093f0  7503      52      51      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      CAC-UDP    0x7f88bc011050  7504      54      53      OK      FIFO 1-7
[Wed Jul 12 11:26:10 2023]      CAC-event  0x7f88bc0690e0  7505      52      51      OK      FIFO 1-7
```

Figure 6: A real-time IOC configuration with different priorities, FIFO scheduling policy, and defined CPU affinity.

The implementation of a real-time operating system plays a significant role in mitigating the concern of trigger drop occurrences. This brings the installation of a real-time kernel and the optimization of its policy settings to ensure a high-performance mode. With the help of the MCoreUtils module, the configuration of real-time parameters for EPICS Input/Output Controller (IOC) threads is also accomplished.

Adapting the MCoreUtils module allows for the adjustment of thread priorities, and scheduling policies (FIFO or Round-Robin) [6]. Additionally, optimizing the CPU affinity of IOC threads can enhance system performance, achieved through the segregation of CPU cores utilized by the house-keeping threads and the EPICS IOC threads. Figure 6 shows the result of our real-time IOC configuration.

Monitoring and Log System

Instead of employing EPICS channel access for logging purposes, an independent log thread is created. This approach serves to avoid potential performance complications caused by an excessive number of EPICS channel access clients. A ring buffer is established to accept log data from the ADC waveform every 20 ms and write them to a local SSD storage.

The log entries consist of three process variables associated with magnet currents, along with pulse information. The process variables include the magnet current's designated set value, its actual output value, and the value retrieved from ADC readback. All data from 8 channels is recorded within a log waveform, tagged with a timestamp sourced from the EVR, the shot ID of current pulse, and injection beam mode of current pulse. These appended attributes facilitate the subsequent diagnostic analysis in case of system failures.

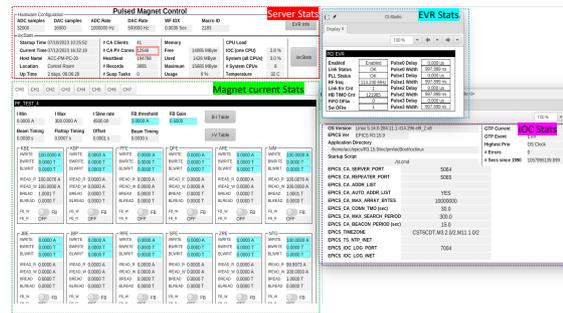


Figure 7: A monitoring OPI for PMCS.

An operator interface (OPI) panel is also provided to help operators monitor the IOC status and quickly find any anomalies. Figure 7 shows the interface.

Performance and Stability

The stability result measured in 2018 is reported in the paper [7]. Similarly, we undertook measurements of a test magnet and subsequently computed the relative stability utilizing statistical metrics including the mean value and standard deviation. The result is shown in Fig. 8. During the 8-hour experiment, the relative stability was 88 ppm, corresponding to the level with the precedent measurement.

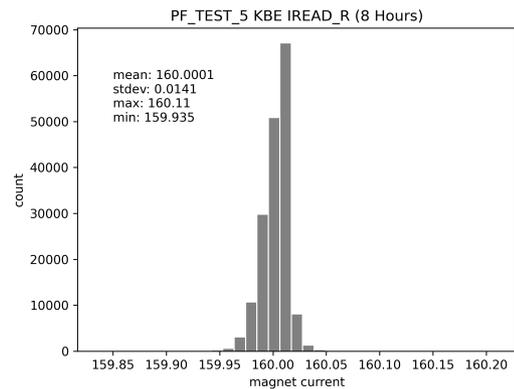


Figure 8: The measured magnet current result of a test magnet for 8 hours operation. The mean value and standard deviation of the current were 160 A and 0.0141 A, respectively.

During 2023 summer, an offline test is performed to examine the long-term stability, after manually establishing

10 thousand CA connections to the EPICS IOC, the new system demonstrated sustained and consistent performance during the operation of one month. Notably, no failures were observed during this period.

EVR Timestamp

During the development phase of the new PMCS, we encountered unexpected system instabilities. Several diagnostic assessments were conducted to isolate the underlying cause of these anomalies. Our investigations revealed that the PCI reading latency was the primary contributor to the observed instabilities.

For a pulsed operation, all data processing inside the PMCS should complete within 20 ms. By default, all EPICS records use the EVR as their primary timestamp source. Within each 20 ms cycle, approximately 300 EPICS records are processed, each requiring a read from the EVR to obtain the corresponding timestamp.

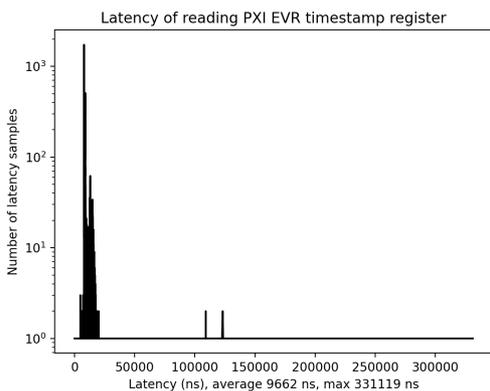


Figure 9: The measured latency of reading PXI EVR timestamp register.

The latency measurement results in Fig. 9 indicates an average latency of approximately 9 μ s, with peak latency reaching as high as 300 μ s in extreme cases. Such long latency can significantly impact the performance of our pulsed control systems, especially when considering the cumulative effect over multiple records.

With this understanding, we have incorporated several measures to mitigate the effects of this latency. By disabling EVR as the default timestamp source for EPICS records, the

latency of data processing is significantly improved.

For specific records where high precision timestamp is needed, we provide an exception mechanism. By adjusting the TSE (Time Stamp Event) field of the designated record, we ensure that it directly accesses the EVR to obtain the timestamp.

CONCLUSION

We have successfully transitioned the PMCS from a Windows 8.1-based LabVIEW framework to a platform-independent operating system and software. This transition involved the development, testing, validation, and refinement of control software on the PXIe platform, all while utilizing the existing hardware. Experimental results indicate that the updated system offers both reliable performance and stability.

As part of our ongoing upgrade to improve the system reliability, we plan to firstly replace one unit with the new PMCS during the 2023 fall LINAC operation. Following a thorough evaluation of its performance and stability, and provided the outcomes meet our requirement, we will proceed to gradually replace all units with the upgraded system in near future.

REFERENCES

- [1] K. Furukawa, M. Satoh, T. Suwada, and T. Nakamura, “Pulse-to-pulse Beam Modulation and Event-based Beam Feedback System at KEKB Linac”, in *Proc. IPAC’10*, Kyoto, Japan, 2010, pp. 1271–1273. <http://jacow.org/IPAC10/papers/tuocmh01.pdf>
- [2] National instruments, <https://ni.com/>
- [3] Micro-research finland, <http://mrf.fi/>
- [4] 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
- [5] Mrfioc2, <https://github.com/epics-modules/mrfioc2>
- [6] MCoreUtils, <https://github.com/epics-modules/MCoreUtils>
- [7] Y. Enomoto *et al.*, “Pulse-to-pulse Beam Modulation for 4 Storage Rings with 64 Pulsed Magnets”, in *Proc. 29th Linear Accelerator Conference (LINAC’18)*, Beijing, China, 2018, pp. 609–614. doi:10.18429/JACOW-LINAC2018-WE1A06