Application of Loongson-Based EPICS Real-Time Controller in Ultra-High Temperature Molten Salt Pump Test Facility

Authors: Deng Qi, Yang Zhenghan, Han Lifeng, Huang Li, Dai Zhimin, Dai Zhimin

Date: 2024-10-08T00:00:00+00:00

Abstract

The high-temperature molten salt pump test facility is a system designed to investigate the performance of key equipment such as pumps, valves, and heat exchangers under high-temperature molten salt conditions. To enhance the degree of domestic localization of its control system and the autonomous control capability of the core controller, this paper presents the design and implementation of a real-time controller based on the Experimental Physics and Industrial Control System (EPICS) on the domestically developed LoongArch instruction set architecture. First, to resolve the incompatibility between software and the instruction set architecture, EPICS, IgH EtherCAT Master, and other software packages were ported to an embedded development board based on LoongArch, enabling functions including control program development and execution, EtherCAT master-slave communication, and testing of the controller' s minimum bus scan cycle. Then, to address the control requirements of the ultra-high temperature molten salt pump test facility, PID temperature control, pressure monitoring, and other functions were implemented on this controller utilizing an independently developed EPICS extension plugin. Finally, the controller' s real-time performance, CPU utilization, and other metrics were tested and analyzed under actual operating conditions to evaluate its performance. Experimental results demonstrate that the controller' s minimum bus scan cycle is 50 ms, the maximum delay time for control task execution is 12.85 ms, CPU performance is favorable, and it satisfies the application requirements of this project. Currently, the controller has been successfully integrated into the control system of the ultra-high temperature molten salt pump test facility, replacing the original x86 server and operating stably.

Full Text

Preamble

Vol. XX, No. X, XXX 20XX NUCLEAR TECHNIQUES

Application of Loongson-Based EPICS Real-Time Controller in the High-Temperature Molten Salt Pump Testing Facility

DENG Qi[1,2], YANG Zhenghan[1], HAN Lifeng[1], HUANG Li[1], DAI Zhimin[1,2]
[1](Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800, China)
[2](University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract

[**Background**] The high-temperature molten salt pump testing facility is designed to evaluate the performance of pumps, valves, heat exchangers, and other critical equipment under high-temperature molten salt conditions. To enhance the localization of its control system and achieve autonomous controllability of the core controller, this paper designs and implements an EPICS-based real-time controller on the domestic LoongArch instruction set architecture. [**Purpose**] The objective is to develop a controller that is both fully autonomous and capable of meeting the demanding requirements of high-temperature molten salt applications. [**Methods**] First, to address software-architecture incompatibility, EPICS, IgH EtherCAT Master, and related software were ported to a LoongArch-based embedded development board, enabling control program development and execution as well as EtherCAT master-slave communication, followed by testing of the controller' s minimum bus scan cycle. Then, based on the control requirements of the high-temperature molten salt pump testing facility, an independently developed EPICS extension module was used to implement PID temperature control and pressure monitoring functions on this controller. Finally, the controller' s real-time performance and CPU usage were tested and analyzed under actual operating conditions to evaluate its overall performance. [**Results**] Experimental data demonstrate that the controller achieves a minimum bus scan cycle of 50 ms, with a maximum control task execution latency of 12.85 ms, and exhibits good CPU performance, meeting the application' s requirements. **Conclusion** The controller has been successfully integrated into the control system of the high-temperature molten salt pump testing facility, replacing the original x86 server and operating stably.

Keywords: Real-time controller; EPICS; LoongArch; High-temperature molten salt pump

1. Control System Architecture

The high-temperature molten salt pump testing facility constructed by the Shanghai Institute of Applied Physics, Chinese Academy of Sciences, comprises a main loop system, molten salt system, gas circuit system, instrumentation and control system, process system, and physical property monitoring system. Its control system adopts a distributed architecture built upon the Experimental Physics and Industrial Control System (EPICS), an open-source software suite developed jointly by Los Alamos National Laboratory and Argonne National Laboratory for constructing distributed control systems. EPICS is characterized by its stability, reliability, high performance, and scalability, and has been deployed at numerous large-scale scientific facilities including the Shanghai Synchrotron Radiation Facility (SSRF) and the High Energy Photon Source (HEPS).

While EPICS enables autonomous control at the software architecture level, the core controllers of the testing facility still rely on imported products. The Input/Output Controller (IOC), a core EPICS component, possesses the capabilities required for industrial control, and numerous precedents exist for its use in process control applications both domestically and internationally. Researchers have also explored various solutions to adapt EPICS IOC to domestic platforms. For instance, Li et al. developed an EPICS and EtherCAT-based control system on the domestic real-time operating system SylixOS for level control in a nuclear energy comprehensive utilization project. The China Spallation Neutron Source (CSNS) deployed EPICS IOC on the domestic HCFA AC800 controller for its associated proton beam experimental platform. Xie et al. ported EPICS IOC to the Loongson 2K1000 processor based on MIPS architecture for remote control of low-level RF systems. However, these solutions still depend on foreign instruction set architectures at the 底层 level. LoongArch, a fully autonomous instruction set architecture independently designed from the top-level architecture down to instruction encoding details, achieves fundamental software-level autonomy. This paper selects a LoongArch-based embedded development board and utilizes open-source software including Linux, the PREEMPT_{RT} patch, and EPICS to implement a completely autonomous and controllable real-time controller. Application results at the high-temperature molten salt pump testing facility demonstrate that the controller executes control tasks accurately and reliably, with bus scan cycle and other key metrics meeting the control system requirements.

The control system architecture is illustrated in [Figure 1: see original paper]. The system comprises three layers: the supervisory layer, control layer, and field device layer. The supervisory layer consists of Operator Interface (OPI) applications that provide human-machine interaction, running operator interfaces, system monitoring, and data visualization to ensure operators can monitor system status in real time. The control layer primarily comprises controllers responsible for real-time control and data processing. The field layer forms the foundation of the control system, containing sensors, valves, and other devices. Specifically, the control layer includes three servers running IOCs that execute control logic and drive EtherCAT devices, serving as controllers, plus one IOC server acting as a data transfer middleware that establishes communication between the OPI layer and air-cooling PLCs as well as heating element current transformers via

RS485/Modbus protocol.

2. System Implementation

2.1 Hardware Platform

The original EPICS IOC in the high-temperature molten salt pump testing facility ran on a Dell Precision 3430 server based on x86 architecture. This work adopts the Loongson Pi Gen2 embedded development board as the controller hardware platform, which features the LS2K1000LA, a 64-bit low-power processor designed for industrial control applications. Table 1 compares the key specifications of the Dell Precision 3430 and Loongson Pi Gen2. While the Loongson board has lower power consumption, fewer cores, and lower clock frequency than the Dell server, it achieves complete autonomy and meets application requirements while significantly reducing overall controller costs.

2.2 Software Architecture

The controller is implemented using open-source software including EPICS, IgH EtherCAT Master, Dls-ethercat, and Linux with the RT-PREEMPT real-time patch. IgH EtherCAT Master is an open-source library that enables EtherCAT master functionality in Linux systems for managing and controlling EtherCAT slave devices. Dls-ethercat is an EPICS module developed by Diamond Light Source based on IgH EtherCAT Master that provides EPICS support for EtherCAT devices. Figure 2 [Figure 2: see original paper] illustrates the software architecture and component relationships. The implementation proceeds as follows: first, Dls-ethercat creates a Scanner bus scanner for reading from and writing to EtherCAT devices; then an I/O IOC is created to communicate with the Scanner via socket, where Asyn is an EPICS module providing asynchronous I/O support; finally, a Main IOC executes specific control logic by accessing PVs in the I/O IOC to perform device read/write operations.

Due to the limited memory and processing resources of the LS2K1000LA, this work employs cross-compilation using LoongArch GCC 8.3 GNU/Linux rc1.2 in a standard Linux environment, with resulting binaries downloaded to the LS2K1000LA. Since EPICS and related software do not officially support LoongArch architecture, a primary implementation challenge involves software-architecture adaptation. Using EPICS Base as an example, Makefiles must be provided for the LoongArch architecture following the naming convention `<CONFIG/CONFIG_{SITE}>.<host-os-arch>.<target-os-arch>`, defining compilation rules for the target platform.

To meet real-time requirements, EPICS was configured with `USE_{{{POSIX}}_{{THREAD}}}_{{{PRIORITY}}_{` enabling EPICS threads to adopt the `SCHED_{FIFO}` scheduling policy. This configuration assigns real-time priorities to EPICS IOC threads and performs memory locking operations, reducing task execution instability caused by thread switching or memory access delays.

2.3 Minimum Bus Scan Cycle

The minimum bus scan cycle represents a critical controller performance metric. This study tests the minimum bus scan cycle by simulating pulse signal input/output and verifying signal waveform integrity and correct pulse counting. The procedure is as follows: (1) The controller IOC simulates a pulse signal with a 1000 ms period, where the IOC Record generating the high-level signal has a scan period of T and outputs to an AO port; (2) The AO port is looped back to an AI port, which the controller IOC reads; (3) The bus scan cycle is set to T; (4) The value of T is progressively reduced until the minimum value that maintains correct pulse counting and waveform integrity is identified as the controller' s minimum bus scan cycle.

Figure 3 [Figure 3: see original paper] shows the input and output signals for the last five cycles when $T = 50$ ms. The measured minimum bus scan cycle is 50 ms, with an average response delay of 113.5 ms.

3. Application and Performance Evaluation

The temperature control, over-temperature alarm, and gas flow acquisition functions of the high-temperature molten salt pump testing facility have been migrated to the EPICS real-time controller developed in this work, replacing the original Dell Precision 3430 server. Controller performance under actual operating conditions with numerous advanced control tasks was evaluated by monitoring CPU load and usage metrics and by designing programs to 统计 control task latency.

3.1 Control Function Development

Control function development comprises two aspects: defining I/O Records and defining calculation Records. The former, stored in the Scanner IOC, specify device types, addresses, and other attributes through DTYP, INP, and other fields to implement I/O control. The latter, stored in the Heater IOC, implement control logic for temperature control and over-temperature alarm using Record types such as calc and compress.

The temperature control system employs a PID algorithm implemented using the "epid" Record type provided by Argonne National Laboratory. The control target is the slope of the historical temperature curve (i.e., temperature increment per unit time), with the PID target slope dynamically adjusted based on the difference between measured and target temperatures. When the measured temperature exceeds the alarm threshold, the system triggers an alarm that persists until manually reset by operators. Figure 4 [Figure 4: see original paper] presents a schematic of the IOC Database configuration (simplified for clarity). The Record `$(pid):TFB 负责收集过去 180 秒的实际测量温度，$(pid):slopeRaw` performs slope fitting on the historical temperature curve using least squares method, feeding the result into the PID control Record '$(pid) 计算输出功率，从

而实现温度的调节。为优化代码结构、提升代码一致性和健壮性，本研究还设计开发了 EPICS 扩展包 eFunBlocks，提供最小二乘法斜率拟合、RS 触发器等常用算法。

Provided by eFunBlocks SNAM=range INAM=range_{init} compress $(pid):slopeRaw'. The over-temperature alarm logic is implemented through calculation Records that compare measured temperature against alarm thresholds.

3.2 Operational Status

The controller has been operating stably in the high-temperature molten salt pump testing facility, implementing 78 PID temperature control loops and over-temperature alarms while monitoring gas flow and pressure in the gas circuit, totaling approximately 2200 Process Variables (PVs). Control tasks are executed via two strategies: I/O triggering and periodic scanning, with PID control tasks using a 1 s scan period. Figure 5 [Figure 5: see original paper] shows the temperature curve of a pipeline during a heating test from 100°C to 200°C. The results demonstrate that the temperature ramp rate and steady-state error meet preset targets, confirming that the controller successfully maintains temperature near the setpoint.

3.3 Performance Evaluation

A comprehensive performance evaluation was conducted by designing experiments to quantify controller real-time performance and CPU metrics. First, a new algorithm module was added to eFunBlocks to calculate jitter for periodically scanned tasks—the deviation between actual and expected execution times—enabling quantitative analysis of real-time performance. Then, 78 PID control loops and other periodic tasks were configured with different scan periods (100 ms, 200 ms, and 500 ms), with each configuration running for 2 hours. Finally, jitter, memory usage, and CPU utilization were recorded and analyzed for each scan period setting.

The data reveal that across all three scan periods, 50% of tasks exhibited jitter greater than 0 ms (i.e., experienced delays). Table 2 summarizes the latency distribution for the three scan period settings. The latency characteristics are similar across configurations, with 95% of delays not exceeding 1 ms and a maximum delay of 12.85 ms.

Memory usage and CPU performance during the test period are shown in Table 3 . The average memory usage remains approximately 29% across all scan periods. Since IOC memory locking size is primarily determined by the IOC Database size, scan period variations do not significantly affect memory usage when Record count and control logic remain unchanged. CPU performance, however, is strongly affected by scan period, particularly the average CPU load. This occurs because the number of tasks that can execute simultaneously is limited to the number of CPU cores (2). When task execution requests exceed core count, queuing occurs. The CPU load average represents the mean number of tasks waiting for execution over the past minute, sampled every 5 seconds using a weighted average that prioritizes recent values. Consequently, CPU load average is primarily influenced by the number of waiting tasks. With constant per-task execution time and core count, a 100 ms scan period generates twice as many task requests per minute as a 200 ms period, resulting in nearly double the number of waiting tasks. Similarly, shorter scan periods increase IOC thread execution frequency and CPU time consumption, raising CPU usage.

Overall, under the load of 78 PID control tasks (approximately 2200 PVs), the controller exhibits low CPU load average and usage at a 500 ms scan period, demonstrating good CPU performance that meets project real-time requirements. As scan period decreases, the CPU load average exceeds 100% due to the limitation of only 2 CPU cores. Therefore, if control tasks cannot be reduced or scan periods extended, a higher-performance CPU should be selected.

Conclusion

This work implements an EPICS real-time controller on a hardware platform based on the domestic LoongArch instruction set architecture and tests its minimum bus scan cycle. To meet the control requirements of the high-temperature molten salt pump testing facility, an EPICS extension module was developed to optimize and port existing control programs to the new controller. Comprehensive evaluation of real-time performance and CPU metrics under actual operating conditions and task loads demonstrates that the controller meets project requirements for bus scan cycle, real-time performance, and other key indicators. The controller has successfully replaced the original x86 server and is operating stably in the high-temperature molten salt pump testing facility. This achievement realizes localization at the underlying instruction set architecture level, enhances the overall domestic content of the control system, and effectively reduces construction costs. Additionally, the performance test results provide important references for future controller selection.

Author Contributions

DENG Qi was responsible for control program development and porting, performance testing, data analysis, and manuscript writing. YANG Zhenghan handled EtherCAT-related software adaptation and testing. HAN Lifeng proposed the research concept and reviewed and revised the manuscript. HUANG Li provided manuscript revision suggestions. DAI Zhimin supervised research planning, progress, and project management.

References

1. EPICS - Experimental Physics and Industrial Control System[EB/OL]. https://epics-controls.org, 2024-08-23.

2. HONG Chunxia, ZHOU Ping, HUA Wenqiang, et al. Automatic calibration system of small angle X-ray scattering experiment station at SSRF[J]. Nuclear Techniques, 2021, 44(1): 18-24. DOI: 10.11889/j.0253-3219.2021.hjs.44.010102.
3. ZHANG Huijie, MA Yinglin, WANG Yufei, et al. Design of Personal Protection System Platform Based on EPICS for HEPS[J]. Nuclear Electronics & Detection Technology, 2021, 41(5): 763-769. DOI: 10.3969/j.issn.0258-0934.2021.05.007.
4. Projects - EPICS Controls[EB/OL]. https://epics-controls.org/epics-users/projects/, 2024-09-26.
5. WU W B, XUAN K, XU W, et al. Development of a control system for the fourth-harmonic cavity of the HLS storage ring[J]. Nuclear Science and Techniques, 2018, 29(11): 153. DOI:10.3788/HPLPB20132504.1001.
6. JIANG Geyang, FANG Wending, SHEN Liren. Embedded EPICS based control system of insertion devices of Shanghai Synchrotron Radiation Facility[J]. High Power Laser and Particle Beams, 2013, 25(4): 1001-1004. DOI:10.3788/HPLPB20132504.1001.
7. LI Guohui, HAN Lifeng, LI Danqing, et al. Development of EtherCAT control system based on SylixOS[J]. Journal of Computer Applications, 2020, 40(S1): 238-242. DOI: 10.11772/j.issn.1001-9081.2019112033.
8. HE Yongcheng, WU Xuan, ZHANG Yuliang, et al. Development of CSNS Associated Proton Experimental Platform Control System[J]. Nuclear Electronics & Detection Technology, 2023, 43(1): 49-55. DOI: 10.3969/j.issn.0258-0934.2023.01.009.
9. XIE Chunjie, ZHU Wenchao, FENG Guangyao, et al. Development of Digital Low-level RF System of S-band Linac[J]. Atomic Energy Science and Technology, 2023, 57(6): 1271-1280. DOI: 10.7538/yzk.2022.youxian.0620.
10. LoongArch[EB/OL]. https://www.loongson.cn/system/loongarch, 2024-08-23.
11. EtherCAT-Master[EB/OL]. https://etherlab.org/en_{GB}/ethercat, 2024-09-04.
12. DiamondLightSource/ethercat[EB/OL]. https://github.com/DiamondLightSource/ethercat, 2024-08-28.
13. EPID Record[EB/OL]. https://millenia.cars.aps.anl.gov/software/epics/epidRecord.html, 2024-09-23

Note: Figure translations are in progress. See original paper for figures.