ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH2BC003

THE LCLS-II EXPERIMENT CONTROL SYSTEM*

A. Wallace†, D. Flath, M. Ghaly, K. Lauer, J. Yin, Z. Lentz, T. Johnson, R. Tang-Kong
SLAC National Accelerator Laboratory, Menlo Park CA, USA

Abstract

The Linac Coherent Light Source (LCLS) has been undergoing upgrades for several years now primarily through two separate major projects: the LCLS-II (Linac Coherent Light Source II) and the LCLS-II Strategic Initiative (or L2SI project). The LCLS-II is a DOE 403.13b project responsible for upgrading the accelerator, undulators and some front-end beam delivery systems. The LCLS-II Strategic Initiative assumed responsibility for upgrading the experiment endstations to fully utilize the new XFEL machine capabilities planned to be delivered by the LCLS-II beam. Both projects included scope to design, install and commission a control system prepared to handle the risks associated with the tenfold increase in beam power we will eventually achieve. This paper provides an overview of the new control system architecture from the LCLS-II and L2SI projects and status of its commissioning.

INTRODUCTION

Experiment Control Systems (ECS) is a division[1] within the LCLS directorate responsible for the X-Ray beam delivery and experiment hutch control systems. The team ensures the control system is well-designed, reliable and capable of supporting a wide range of scientific experiments conducted at LCLS. The team works closely with various stakeholders including scientific staff, other engineering disciplines and technicians to gather and understand operational requirements.

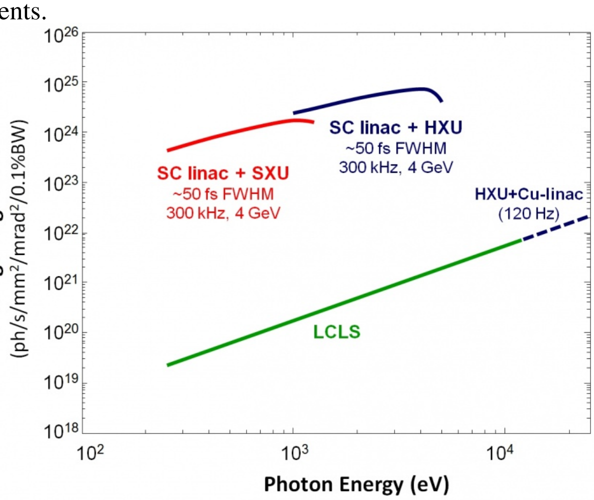

Figure 1: The increase in brightness and average power from LCLS-I to LCLS-II, with its soft x-ray undulators (SXU) and hard x-ray undulators (HXU).

The scope of ECS responsibilities start at the Electron Beam Dump (EBD) and continues through a Front End Enclosure (FEE), into the various experimental hutches. Currently, the team supports both LCLS-I and LCLS-II areas.

LCLS-I experiment controls is primarily based on the Experimental Physics and Industrial Control System (EPICS), integrating a wide variety of components for mechatronics and process control. Data acquisition and analysis systems are handled outside of the purview of ECS.

With the increased power and pulse energy of the new LCLS-II beam (see Figure 1), the necessity for a robust control system with a strong emphasis on machine protection became crucial. This escalation in operating requirements necessitated a different controls system design from LCLS-I. Additional key goals included increased dependency on automation for regular operations and checkouts, and optimization of beam delivery as well as anticipation of thermal effects from high-power beam. Increased machine protection requirements combined with more complex instrument designs were the primary factors influencing the LCLS-II ECS design.

SCOPE

The ECS scope spans from integrating various mechanical and electronics components such as actuators and sensors and extends all the way to the graphical user interface (see Figure 2). This integration involves the development and implementation of advanced logic to achieve complex functionality.

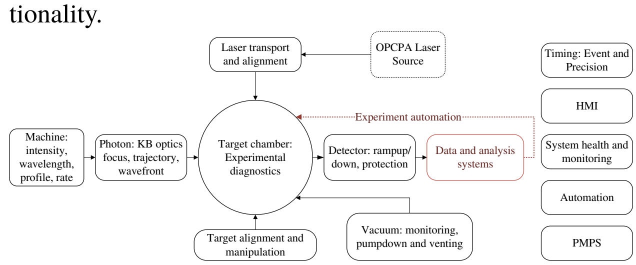

Figure 2: Integrated aspects of the LCLS-II experiment system scope.

ARCHITECTURE

The LCLS Experiment Control System (ECS) team maintains a network-based distributed controls systems built on EPICS (see Figure 3). The ECS architecture is built using a combination of real-time control platforms (in the form of PLCs), EPICS device support, high-bandwidth networking and computing infrastructure, and Python-based user interface frameworks (see Figure 4).

* Work supported by U.S. D.O.E. Contract DE-AC02-76SF00515
† awallace@slac.stanford.edu
[1] ECS is also the term used for the Experiment Control System itself.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

19th Int. Conf. Accel. Large Exp. Phys. Control Syst. ICALEPCS2023, Cape Town, South Africa JACoW Publishing
ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH2BC003

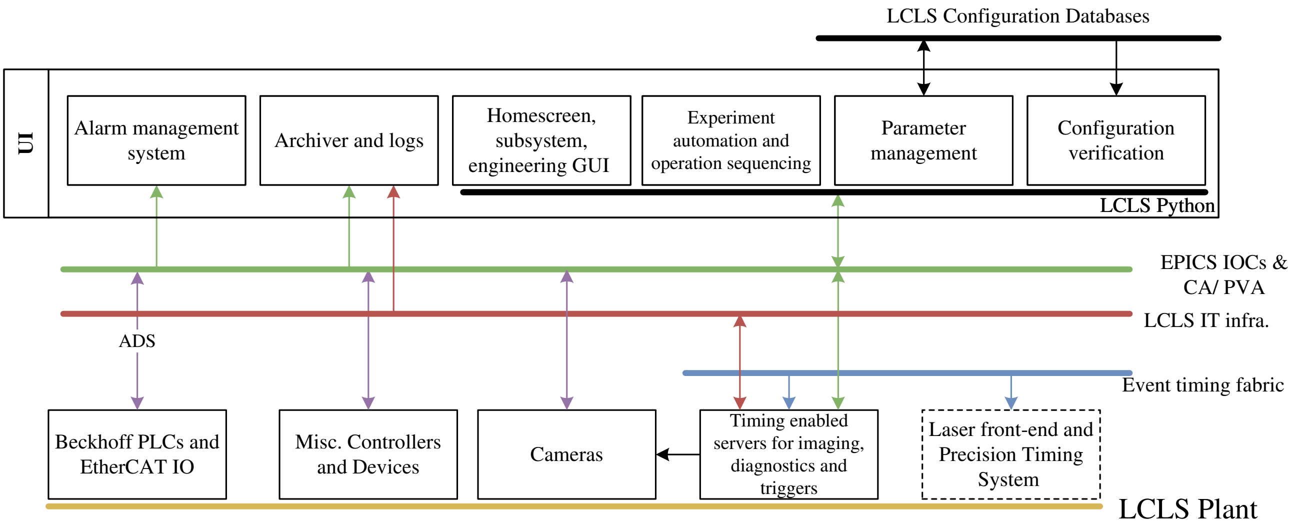

Figure 3: An overview of the LCLS Experiment Control System architecture.

Industrial Controls and Custom Software Libraries

Four Structured Text (ST) libraries form the core codebase for the ECS. LCLS General (`lcls-twincat-general`) consists of a suite of PLC diagnostics, common useful functions for data collection and a single-cycle logger which facilitates process event logging. These log messages are sent through a Logstash and ElasticSearch-based pipeline [1] and made accessible to staff by way of Grafana. LCLS Photon Machine Protection System (PMPS, `lcls-twincat-pmps`) implements the Preemptive Machine Protection System concept as a framework [2]. LCLS Motion (`lcls-twincat-motion`) delivers the fundamental elements for motion control and equipment protection, including gantry motion, and also utilizes the PMPS library to provide a general state machine for controlling multi-state mechatronics in combination with safe beam parameters. Finally, LCLS Vacuum (`lcls-twincat-vacuum`) implements a standardized interlock system for ultra-high vacuum preservation and operation of the most critical vacuum hardware. Additional libraries for specific subsystems such as optics, diagnostics and sample delivery augment this suite to deliver standardized approaches to most experiment system challenges.

Vacuum and motion control is realized through integration with a Beckhoff PLC, typically a CX5010 or CX2020 running TwinCAT 3 on Windows Embedded Compact (WEC) 7 [2, 3]. Plans are underway to evaluate and then migrate to TwinCAT BSD [4], avoiding Windows as a PLC runtime operating system. A wide assortment of other commercial off-the-shelf (COTS) elements, high-voltage power supplies, air quality and environmental sensors (to name just a few), are directly integrated into the control system using Moxa serial port servers, or via Ethernet and EPICS device support. The array of supported devices is managed using an internally published list (via Confluence) of approved hardware

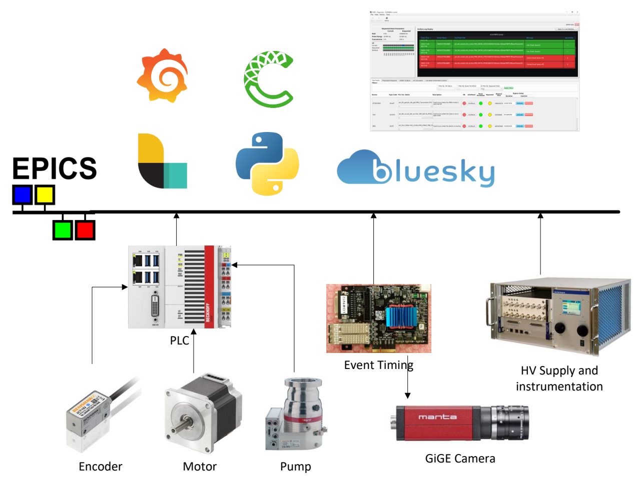

Figure 4: A high-level view of the distributed EPICS-based ECS control system.

and manufacturers which stakeholders and other engineering teams are required to consult during an early design phase.

Vacuum gauges and pump controllers are currently integrated primarily through discrete analog and digital interfaces.[2] The vacuum controls system is responsible for controlling the various vacuum devices as well as for protecting the vacuum system. Interlock logic is implemented to prevent unsafe or undesirable conditions. While most of the vacuum system equipment protection is preemptive, reactive protection is implemented using the MKS 422 cold cathode with fast TTL (transistor-transistor logic) output in combination with a VAT fast shutter on a high-priority PLC task of the vacuum system. The vacuum system also reports beamline valve positions to the PMPS and MPS in order to turn off the beam if one of the valves is closing (see Table 1).

[2] This approach will likely be discontinued in favor of purely serialized interfaces.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH2BC003

Table 1: Selected Technical Specifications for the Vacuum System

Specification	Value
Typical min. measurable pressure	5×10^{-10} Torr
Specialized min. measurable pressure	1.5×10^{-12} Torr
Reaction time of fast shutter system	500 µs-100 ms
Typical pressure measurement delay	120 ms

Motion control is accomplished primarily by generic 2-phase stepper motor integration using the EL7041 and EL7047 Beckhoff EtherCAT terminals (see Table 2.[3] Early designs used ElmoMC Gold DC Bell drives integrated with a Beckhoff PLC, but this approach was abandoned with the release of the EL5042 BiSS-C encoder terminal. Some large x-ray optics rely on EtherCAT and TwinCAT NC coordinated motion for gantry axes. Optical laser system mechatronics are primarily implemented using SmarAct piezo actuators.

The motion control system was designed with absolute encoding as a requirement for any device that might intercept beam. Incremental or relative encoding with homing was deemed unacceptable for such applications as it was determined that absolute encoding would reduce risk in operations as well as improve operational efficiency. Renishaw Resolute absolute encoders (see Table 3) are strongly preferred and used whenever technically feasible. Exceptions are made on occasion when form-factor or other concerns are present. LVDTs (Linear Variable Differential Transformer) are utilized in high-radiation areas such as the electron beam dump.

Table 2: Motion Mechatronics Quantities

Specification	Value
Linear prec.	0.01 mm-3.5 nm
Angular prec.	0.023 µrad

EL70*7, 32-fold microstepping, 2-phase.

Table 3: Position Feedback Quantities

Specification	Value
Highest precision	1 nm
Highest readout frequency	10 kHz

Assumption: Biss-C Renishaw Resolute Linear.

Power

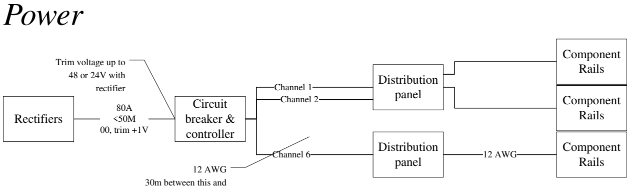

Figure 5: Centralized DC power concept.

The DC power system design aims to provide a standard solution for 24 and 48V DC power throughout the ECS. It was recognized from the LCLS-I design that power for the wide variety of devices integrated into the control system was often an afterthought. There was little to no monitoring of circuits, a prevalence of conventional fuses for circuit protection, and a multiple unique implementations of power supply aspects. Little to no thought was given to the voltage level or tolerance requirements of a selected device, so often devices were powered by specific voltage levels, 9, 12, 15, or worst devices with bipolar power requirements such as +/-15V.[4] For the LCLS-II ECS it was decided to strongly require all devices to be selected or otherwise designed to use 24VDC, or a range including 24VDC. For demanding motion control applications, motor phase voltage was set to 48VDC.

Meanwell RKP units (1U, 3-slot) loaded with Meanwell RCP modules provide either 24 or 48VDC sources. These rectifiers are used as the single power source for a given area such as a hutch, and are installed in a n+1 redundant fashion.[5] The power is distributed first through another 1U COTS component which provides a level of circuit protection using electromagnetic breakers at 15A (the rectifiers sense shorts will also shut off their output), and voltage and current monitoring (see Figure 5).

The 15A branches are distributed again and connectorized to plug into PLC IO panels. The PLC IO panels have one more level of circuit protection for the 24V bus in the form of adjustable electronic fuses. These fuses provide a near instantaneous response to overcurrent conditions and can be reset, avoiding any need to maintain a fuse stock.

Event System and Imaging

The LCLS-II event system is the second SLAC iteration on the MRF EVR protocol and hardware, with key differences including the content of the event datagram and compensation for intrinsic delays in the fiber distribution elements. Where the LCLS-I timing system uses Event Generators (EVGs) and Event Receivers (EVRs) the LCLS-II system uses Timing Pattern Generators and Timing Pattern Receivers (TPG and TPR resp.). The ECS utilizes the same event system as the upgraded machine for events and sequencing, with the primary difference being the ability to switch experiment area event systems between the LCLS-I and LCLS-II timing stream using an element called the X-Ray Timing Pattern Generator (XTPG). The LCLS-II event system is distributed via fanout application boards installed in ATCA chassis. Event data is consumed either in other ATCA AMC boards, rear transition modules, or in other PCIe FPGA boards installed in other generic 1U servers connected by fiber.

Imaging for beam profile monitors, interaction point spatial alignment, wavefront measurement, and optical laser diagnostics is delivered primarily by GiGE Allied Vision cameras integrated into EPICS via soft IOCs (Input/Output

[3] Only the EL70*7 is recommended today.

[4] This would occur in some vacuum gauges which presumeably had purely analog circuitry on-board.

[5] Meaning there is enough power to cover the full demand, with one unit removed in case it fails and needs to be replaced live.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

19th Int. Conf. Accel. Large Exp. Phys. Control Syst. ICALEPCS2023, Cape Town, South Africa JACoW Publishing
ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH2BC003

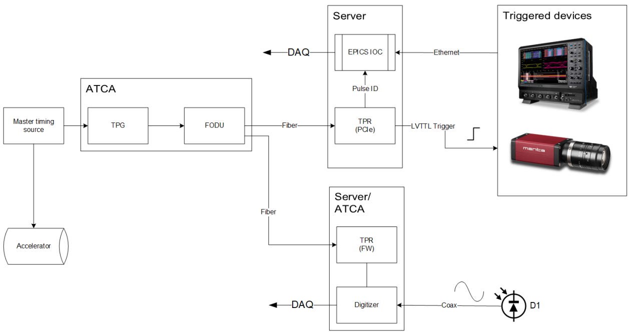

Figure 6: LCLS event system concepts. The event system is primarily used for associating collected signals or data with a particular pulse. Interfaces to the Data Acquisition (DAQ) system intake the "timestamped" data after which it is collated with other data sources.

Controllers). These IOC are hosted on generic 1U Ciara servers equipped with AMD EPYC 7282 16-core processors running at 2800 MHz and 64GB of RAM. These servers provide 2 PCIe slots, one of which is occupied by a TPR. This card also provides configurable, machine event aligned LVTTL trigger outputs for camera frame acquisition and miscellaneous applications which need to be temporally aligned with beam arrival (see Figure 6). The other PCIe slot is used for a 4x NIC for direct connection to the cameras. While not technically designed to be real time, these soft IOC-based camera systems are sufficient to assign a beam pulse ID to an acquired camera frame at rates up to 120Hz. This image data is utilized either by higher-level alignment tools, or recorded by the LCLS-II DAQ via a CA (EPICS Channel Access) or PVA (PVAccess) interface. A control system standalone recorder is also available for situations where the complete DAQ system is unnecessary. This approach is sufficient for the ECS as actual data acquisition for experiments is done by the LCLS-II DAQ.

EPICS, Services, and Toolchains

EPICS IOCs provide a middleware layer into which system components are integrated with the control subsystem (see Figure 7). A tool called `pytmc` [5] autogenerates EPICS process variable databases which use the Beckhoff ADS interface for PLC variables tagged by source code annotations known as pragmas. `pytmc` also enables PV names to be formed via PLC variable structure hierarchies leading to efficient integration of PLC projects with EPICS.

Device support and integration is provided by a template and child IOC structure to ensure consistency. EPICS base versions ranging from 3.14 through 7.0 are deployed around LCLS, with new installations being built on EPICS 7. IOCs are deployed via NFS using procserv encapsulation. A Python tool named `IocManager` facilitates deployment and management of all IOCs by area (which closely corresponds to subnet), enabling easy restarting, and troubleshooting connection via telnet. IOC log files and process variable information are stored on a shared network filesystem. IOC configuration is stored in simple files which are tracked by git and synchronized daily on GitHub, while a separate small tool tracks the status of all deployed IOCs and publishes an up-to-date list to a private Confluence page every 15 minutes. An effort is underway to bring IOC startup script linting and continuous integration to GitHub by way of the package `whatrecord` [6].

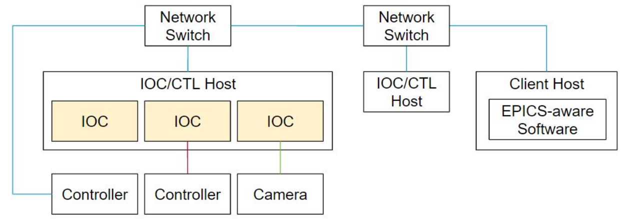

Figure 7: EPICS IOC communication over the network.

The EPICS Archiver Appliance is deployed to record PV history over time. Both readback and setpoint PVs are recorded typically at 1Hz though 10Hz is not rare. Waveforms and multidimensional array PVs are not recorded by the Archiver. A Python IOC provides Archiver Appliance metrics for inclusion in the Archiver itself. While the Appliance ships with a highly functional web interface for displaying PV histories, the ECS uses a deployment of Grafana with the Archiver Appliance integrated with a datasource plugin [7]. Grafana displays are organized into panels and dashboards and the system is configured such that staff has permissions to create their own displays within their folders in the web interface. This has led to rapid growth in customized displays for each experiment area and teams focused on various aspects of LCLS operations (see Figure 8). Additional datasources are included in the Grafana deployment, providing the ability to display and correlate data from across the stack [1].

Figure 8: An example Grafana dashboard showing beam paramter metrics and trends.

Python Environment and User Interfaces

LCLS operations had been supporting the development of a uniform Python framework for experiment automation and this effort was continued under the L2SI project scope.

PyDM, the Python Display Manager, an EDM replacement, was officially adopted by SLAC and development of its core capabilities was carried out by a collaboration between Technology Innovation Directorate (TID), LCLS and

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH2BC003

Accelerator Directorate (AD). PyDM's capabilities were put to the test with the delivery of a dynamic and information dense machine protection system diagnostic tool for operators. Thousands of PV connections with conditional display logic sorting, and even an embedded web display for Grafana logs demonstrate the power of PyDM [2]. Artisanal - i.e., hand-made - screens are required in some cases but ECS GUIs are automatically generated to the extent possible.

LCLS is subdivided into areas or hutches, with different teams responsible for each area. Home screens for these areas are generated dynamically from device listings stored in a database managed by the Python package `happi`. An LCLS-made software component called `LUCID` uses a simple YAML file to specify the arrangement of a device status synoptic display for the area. Additional tools and shortcuts are included in the home screens for operator convenience and are specified in the YAML file. From LUCID (or directly from the command-line), operators can access beamline controls in the form of auto-generated screens or the hand-tailored screens when available. These screens are created from device class definitions by a component called `typhos`. The device classes are contained in a package called `pcdsdevices`, which is built on the more ubiquitous EPICS Python module, Ophyd. Typhos is based on the Model-View-Controller (MVC) pattern and is aimed to deliver a consistent interface to the control system for CLI[6]- and GUI-centric stakeholders. The approach delivers a highly maintainable system. The full potential of Typhos has not yet been explored, but creating a framework for the autogeneration of highly accurate and easily re-configurable UIs justifies the development cost. Automated Confluence documentation leverages the happi database to generate per-device information with user-editable notes sections. These pages can be shown directly in user-facing typhos screens with a single mouse-click. Similarly, device-specific issues can be reported to Jira with a click on its respective typhos screen.

Lightpath is a GUI application that aims to provide operators a rapid assessment of the state of the beam path, and the devices that may potentially block beam. Lightpath gathers all currently active[7] devices from the happi database, and places them within the facility. The facility is represented by a directed graph, with the devices as nodes. At startup, the devices are each placed on a branch based on their input and output branches (as listed in the happi database). These branches are then merged to form the full facility graph. Once the facility graph has been constructed, Lightpath finds all possible paths from source to the requested destination. In most cases this results in a single beam path, but in some cases there are multiple ways to get to the same end station.

As previously mentioned, in addition to the GUIs, a CLI can be used to control beamline devices from a terminal. The CLI enables complex operations such as scans and other automated routines that are less efficient through a GUI. A variety of Python modules, collectively referred to as "Hutch Python" at LCLS, have been developed to interact with EPICS through CA, based on Ophyd and BlueSky[8]. The BlueSky Python package provides a scanning and data acquisition control framework for executing data collection and automating processes. It simplifies the scanning and automation process by providing a high-level interface for sequencing control actions such as step-scanning. It also contains built-in automation for pausing and resuming scans under predefined conditions. A typical example is a rastered sample pausing while the machine protection system signals beam is not permitted and resuming once beam is back.

COMMISSIONING STATUS

Four areas of LCLS have been upgraded to the LCLS-II ECS in concert with the new equipment installation: the Front End Enclosure (FEE), the LCLS optical laser labs (generation and distribution), the Resonant Inelastic X-ray Scattering instrument (RIXS) and the Time resolved atomic, Molecular and Optical instrument (TMO). The Tender X-Ray Instrument (TXI) is projected to receive first light in 2024 and will consist of two beamlines converging on a single interaction point.

Commissioning of the new soft X-Ray and hard X-Ray photon beam lines and diagnostics in the FEE using the original LCLS accelerator (Cu, warm linac) and new adjustable gap undulators began in July of 2020, with instrument commissioning (in TMO first) proceeding shortly thereafter in September.

Figure 9: Remote commissioning and first light activities taking place via the LCLS web portal (access to Grafana), SSH+X11 forward for control screens and Zoom for operator assistance, all from home.

Site access was restricted due to the COVID-19 pandemic starting in March of 2020. All of remaining work leading up to the FEE area closure for beam had to be done with limited access to the facility, and through remote support. Several technologies were brought to bear including: Zoom, Nomachine session sharing, SSH+X11 (for conventional remote access), Double3 robotics (telepresence), and Realware headsets (for hands-free troubleshooting assistance).

[6] Command Line Interface
[7] Defined as installed in the beam path.
[8] Python packages developed at the National Synchrotron Light Source II (NSLS-II)

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

(Figure 9) A SLAC-hosted bandwidth and connection tester was deployed to provide a baseline metric for connection performance. [9] Nearly all of the initial commissioning was conducted using Zoom screen sharing from home with some on-site presence in instrument control rooms augmented by entirely remote teams of engineering and users [8].

Operations in TMO, and chemRIXS proceeded with the warm linac, 120 Hz beam from 2020 through 2022. Due to inclement weather leading to a helium vent in the LCLS-II cryoplant as well as other SLAC-wide issues, superconducting (SC) linac commissioning activities were postponed until July of 2023. SC beam key performance parameters were measured and marked as achieved shortly thereafter, and subsequently the LCLS-II project completed critical decision 4 in late September marking the end of the 10 year project. Commissioning of the L2SI areas, TMO and RIXS is proceeding with SC beam delivery now.

COMPARISON TO LCLS-I

It is illuminating to compare the LCLS-I and LCLS-II ECS architectures to highlight key differences and identify notable evolutionary changes.

In both, EPICS is the integrating layer for the control system. Most elements of the original EPICS framework remained unchanged with the exceptions of EDM to PyDM and deployment of EPICS 7. Channel Access remains as the dominant protocol for the EPICS installation, as opposed to PV access. All new LCLS-II ECS IOCs were built with caPutLog enabled. Logging and archiving is prioritized in the LCLS-II systems [1].

Imaging systems in LCLS-I included a substantial number of cameras using the CamLink interface and framegrabber technology to ensure synchronized acquisition at full 120 Hz rate. Since then CamLink camera availability was limited and then completely stopped. As a result only a very small number of beamline diagnostics in LCLS-II and L2SI have been deployed with CamLink cameras, and even these are considered legacy and will likely be changed for GiGE cameras in the future in part for uniformity and better long-term support. It was also recognized as we began to operate that 120 Hz or greater beam synchronous acquisition was not essential for commissioning or operations purposes. In fact, it was determined that diagnostic cameras could sufficiently run using the GiGE interface as described above in the Imaging section. Given the extra framegrabber hardware and lack of available CamLink camera options the GiGE solutions became de facto for control system imaging. This change simplified the imaging system design by further reducing the number of types of cameras requiring support and the hardware required.

In LCLS-I, PLCs were only used for vacuum systems. Mechatronics were provided primarily by "smart" motors with integrated controllers which were interfaced via serial connections (again through serial port-servers) and the EPICS Motor record. While the integrated smart motor concept is excellent for its reduced wiring and corresponding flexibility, attempting to deliver complex motion control schemes such as kinematic transforms and gantried axes in a reliable fashion would have been exceedingly difficult. This would have required either developing a communication fabric for direct connections between smart axes, or using EPICS. EPICS is not typically deployed to be deterministic. Also, it is uncommon in the EPICS community apply the framework to provide hard real time equipment protection interlocks.[10] In addition to this consideration, the PMPS design intentionally merged into the same low-level control platform (the PLCs and TwinCAT) [2]. As a result, in LCLS-II PLCs play a more prominent role in low-level control of all mechatronics and other process elements.

CONCLUSIONS

The ECS as designed and deployed for LCLS-II and L2SI has met the requirements for equipment and machine protection, control, observability, reliability and flexibility. Another notable influence on the design includes a drive for internal standardization in a variety of ways. Using code libraries for vacuum device interlocks, and axis control is one example of standardization. Consolidating a list of supported hardware and creating architectures which dictate exactly how a certain device is integrated into the control system is another. The DC power system is yet another. This drive towards standardization was fueled in part by the notion that eventually the ECS will require minimal extension or non-recurring cost to cover any new project in the lifetime of LCLS.

One challenge faced early in the process was the lack of engineering experience using industrial controls platforms such as PLCs. PLC programming is substantially different from programming in Python or even producing EPICS components. A related challenge was the lack of software engineering best practices in the PLC programming world. These best practices including linting, CI, unit testing and even version control to name a few are still percolating into PLC programming. The Beckhoff platform was specifically selected for its early investment in these practices. Unit testing for PLC code played a key role in the successful delivery of essential aspects of the PLC systems [9]. A complete continuous integration (CI) pipeline for ECS PLC systems is still under active development.

Finally a core goal for the ECS design was to make it exportable. This was in part the result of working with 3rd party contractors and other integrators. Some L2SI components were bid out to be built by firms with more experience and the manufacturing capabilities. LCLS decided to require the acceptance testing of these systems be carried out with the same control electronics and software as would be used in operations at LCLS. This is was received as a somewhat

[9] It probably should be acknowledged more just how vital high-bandwidth connections to homes and offices are.

[10] It should be noted that EPICS does not have to be ruled out of these kinds of applications. As a framework it could be extended and applied to hard real time requirements.

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH2BCO03

Content from this work may be used under the terms of the CC BY 4.0 licence (© 2023). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

unusual approach, however in the end has proved to be an important caveat for systems with the highest requirements.

ACKNOWLEDGEMENTS

The ECS design and commissioning was the culmination of effort from a large team over multiple years. Along with the listed authors of this paper, credit and acknowledgement is deserved by the following individuals (and is not limited to): Michael Browne, Dan Damiani, Jeremy Lorelli, Marcio Paduan, Bruce Hill, Jyoti Joshi, Christina Pino, Allen Pai, Nolan Brown, Sameen Yunnus, Rajan Plumley, Van Xiong, Amedeo Perazzo, Riccardo Veraldi, Omar Quijano, Murali Shankar, Teddy Rendahl, Lupe Salgado, Adam Egger, Jackson Sheppard, Danh Du, Gibson Locke, Abdullah Rashed, Ernesto Paiser, Ray Rodriguez, Karl Gumerlock, Hugo Slepicka, Ernest Williams, Silke Nelson, Mark Freytag and Nick Waters.

Collaboration with other facilities and teams around the world and here at SLAC also played a large part in the success of the ECS. Many thanks to the following facilities and teams: SLAC, TID and EED for their participation in our design reviews and working with us to ensure our systems mesh well together. ESS, for their invaluable pioneering spirit which led to the ADS based device support which the ECS used exetensively. Also for their ongoing collaborative efforts which have provided so much value in both directions. EuXFEL, for teaming up to get the EL5042 produced which became a crucial component in the ECS design.

REFERENCES

[1] K. Lauer, "Centralized logging and alerts for EPICS-based control systems with logstash and grafana", presented at ICALEPCS'23, Cape Town, South Africa, 2023, paper THPDP089, this conference.

[2] A. Wallace, "The LCLS-II experiment controls preemptive machine protection system", presented at ICALEPCS'23, Cape Town, South Africa, 2023, paper TUPDP129, this conference.

[3] M. Ghaly and A. Wallace, "The LCLS-II experiment system vacuum controls architecture", presented at ICALEPCS'23, Cape Town, South Africa, 2023, paper WE1BCO04, this conference.

[4] K. Lauer, "TwinCAT BSD virtual machines and Ansible provisioning", presented at ICALEPCS'23, Cape Town, South Africa, 2023, paper THMBCMO13, this conference.

[5] pytmc TwinCAT PLC code to EPICS database tool, `https://github.com/pcdshub/pytmc`

[6] K. Lauer, "`whatrecord`: A python-based EPICS file format fool", presented at ICALEPCS'23, Cape Town, South Africa, 2023, paper THMBCMO08, this conference.

[7] Epics archiver appliance datasource, `https://github.com/sasaki77/archiverappliance-datasource#epics-archiver-appliance-datasource`

[8] Remote participation in experiments, `https://confluence.slac.stanford.edu/display/PCDS/Remote+Participation+in+Experiments`

[9] Tcunit, `https://github.com/tcunit`