DESIGN AND IMPLEMENTATION OF THE LCLS-II MACHINE PROTECTION SYSTEM*

J. A. Mock†, Z. Domke, R. T. Herbst, P. Krejcik, L. Ruckman, L. Sapozhnikov
SLAC National Accelerator Laboratory, Menlo Park, United States

Abstract

The linear accelerator complex at the SLAC National Accelerator Laboratory has been upgraded to include LCLS-II, a new linac capable of producing beam power as high as several hundred kW with CW beam rates up to 1 MHz while maintaining existing capabilities from the copper machine. Because of these high-power beams, a new Machine Protection System with a latency of less than 100 µs was designed and installed to prevent damage to the machine when a fault or beam loss is detected. The new LCLS-II MPS must work in parallel with the existing MPS from the respective sources all the way through the user hutches to provide a mechanism to reduce the beam rate or shut down operation in a beam line without impacting the neighboring beam line when a fault condition is detected. Because either beam line can use either accelerator as its source and each accelerator has different operating requirements, great care was taken in the overall system design to ensure the necessary operation can be achieved with a seamless experience for the accelerator operators. The overall system design of the LCLS-II MPS software including the ability to interact with the existing systems and the tools developed for the control room to provide the user operation experience is described.

INTRODUCTION

The Linear Accelerator Facility at the SLAC National Accelerator Laboratory (SLAC) has seen a significant upgrade to the Linac Coherent Light Source (LCLS) facility with the installation of LCLS-II [1]. This new accelerator and undulator complex was designed to increase capacity at SLAC for photon science using the free electron lasers. A high level diagram of the new facility is shown in Fig. 1. Because it uses cryogenic RF cavities, the LCLS-II accelerator is capable of delivering electrons with a minimum bunch spacing of 1 µs and maximum electron energy of 4 GeV, leading to a maximum overall beam power of 120 kW. The high rate and high power potential of these beams requires safety systems capable of reacting to beam events such that operations remain safe. The personnel protection and beam containment systems (PPS and BCS) are safety systems designed to protect people from radiation-based hazards, and the Machine Protection System (MPS) is designed to protect the accelerator from damaging itself. The MPS is not a credited safety system, so it is designed to be more agile to allow a variety of operating conditions while still protecting the accelerator.

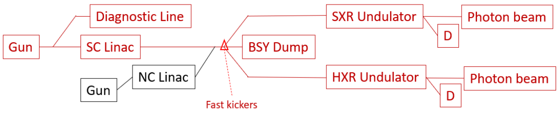


Figure 1: A high level map of the new LCLS-II facility. The MPS has a mitigation device to inhibit beam or reduce beam rate to each of the destinations shown. Additionally, both the new LCLS-II superconducting accelerator and legacy normal conducting accelerator can deliver to the undulator complex, though not yet at the same time.

The scope of the MPS is confined exclusively to shutting off the electron beam when a fault condition occurs that can potentially damage beam line hardware. Other systems that protect high power devices such as power supplies, RF power sources, vacuum systems, or cryogenic systems are handled separately as equipment protection. A key driving parameter of the MPS is the maximum allowable time interval in which the beam must be shut off before damage can occur. The MPS requirement for the original LCLS dictated that the electron beam be shut off within one beam pulse at the full repetition rate of 120 Hz. This is not possible in LCLS-II where the minimum bunch spacing is only 1 µs and propagation delay for a signal in a cable from one end of the accelerator to the other can be as long as 20 µs, not including additional processing delays incurred from electronics. The MPS baseline beam shutoff time, defined as the time between detection of fault and loss of photo-current, is required to not exceed 100 µs to avoid catastrophic damage to the beam line, though in principle the MPS physics requirement is as low as reasonable achievable. Not every fault condition requires the fast shutoff time of 100 µs. For example, a slow change in some state, such as a temperature rising, allows ample time for the control system to warn of the impending change. Therefore, MPS responses to prompt events such as beam loss mitigate within the fast response window, and other, slower events and more complicated logic process within a 360 Hz window, equal to the processing time of the LCLS-I MPS.

As shown in Fig. 1, the LCLS-II accelerator can deliver to some combination of an injector diagnostic line, one of two undulator beam lines, or through the linac to a high power dump in the SLAC Beam Switch Yard (BSY). The default destination for the electron beam is this high power dump. Pulsed kicker magnets are used to kick the beam into one of the other beam lines, which means if the pulser does not kick, the beam will need to travel to the BSY beam dump. Therefore, the MPS uses this high power dump as its pri-

[*] SLAC is supported by the U.S. Department of Energy, Office of Science, under contract DE-AC02-76SF00515

[†] jmock@slac.stanford.edu

mary mitigation device. It grants and revokes permits to the various kicker magnets in order to control beam power or completely inhibit beam to the other destinations. Additionally, the MPS can inhibit the beam within the entire facility by mitigating at the electron gun (through an interaction with the drive laser). The mitigation device for each destination can be treated independently and all can operate at the response times listed above. A further complication is the continued availability of the legacy LCLS-I normal conducting linac. Figure 1 shows that each of these two machines are each capable of delivering to the undulator complex, though not simultaneously through the same beamline. The two accelerators can operate side-by-side, but each has its own, independent MPS, so faults in one system do not interact with the other.

The MPS takes inputs from many types of beam line devices such as:

- Obstructions. Objects like vacuum valves or diagnostic screens.
- Beam Loss Monitors [2–4]. Primary mechanism used to verify transport of the beam to the dump.
- Destination dipole and chicane magnets. Secondary mechanism used to verify accelerator conditions are correct for beam transport
- Bunch Charge Monitors. Primary mechanism used to verify power levels are not violated and charge is not lost
- Beam Position Monitors. Secondary mechanism used to verify beam is not mis-steered to preemptively reduce power before beam loss becomes a problem
- General Digital Inputs. Used to incorporate additional interlock logic.

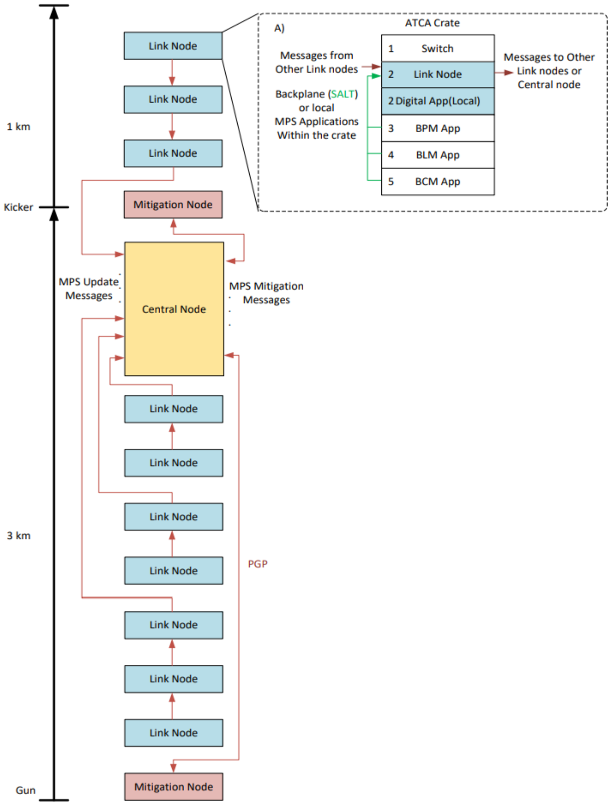


Figure 2: The low-latency MPS communication network.

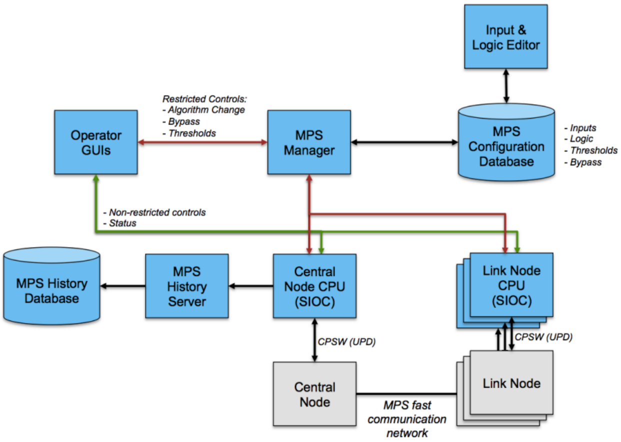


Figure 3: High level software architecture.

SYSTEM ARCHITECTURE

The MPS is comprised of a collection of distributed devices known colloquially as Link Nodes that collect and process data and send them off to a central processor referred to as the central node. The central node collects these data and compares them against a pre-programmed logic table to determine the overall state of the accelerator. The output of this calculation is then distributed to other link nodes that grant or revoke permits to mitigation devices to allow beam to certain destinations. The link nodes and central nodes process their data and send out new messages with each 1 µs clock cycle from the LCLS-II timing system. The link nodes connect to the central node via a dedicated, low-latency communication network with a bandwidth of 5 Gbps. To conserve bandwidth and avoid many km-long fiber runs, the link nodes are arranged into a daisy chain whereby one link node feeds its information to the next until no more than half of the available bandwidth is consumed. A diagram of the MPS network is shown in Fig. 2. Because the MPS operates at the full 1 MHz repetition rate of LCLS-II, the core functionality of the link nodes and central node is achieved within an FPGA. The MPS is built upon the SLAC Common Platform, a generalized facility for beam instrumentation based on the ATCA shelf format [5]. The various beam line diagnostics such as BPMs, BCMs, and BLMs also use this platform. The insert in Fig. 2 shows a typical ATCA crate installation at SLAC. Each crate includes a network switch in slot 1, a link node in slot 2, and application payloads in the other slots. The application payloads all contain an MPS firmware block which handles applying thresholds to analog data, converting the analog signals into digital bits to be sent to the central node. The full complement of logic is distributed across the entire system to preserve bandwidth to ensure future expand-ability. The LCLS-II project saw the

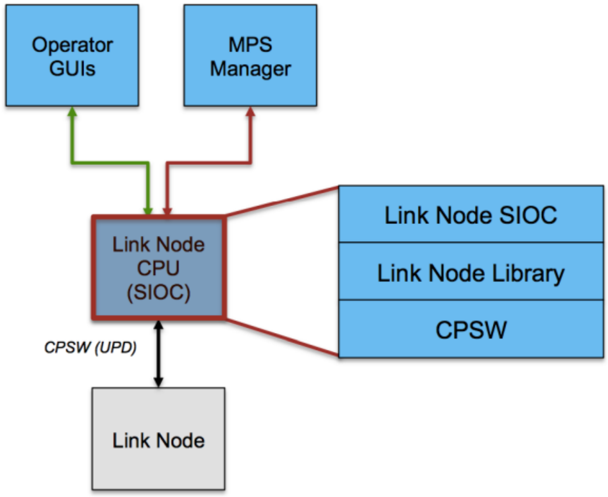


Figure 4: Link node IOC interfaces and internal software layers.

installation of approximately 100 link nodes (ATCA crates) and 500 discrete applications (slots within the crate). The LCLS-II MPS system is described in more detail in [6].

A computer with local 10 Gbps network is connected to each crate to provide a mechanism for data transfer from the FPGAs to the control system [7]. The SLAC Linear Accelerator Facility control system is built with the EPICS toolkit [8]. One EPICS IOC is launched per application (slot) in the ATCA crate. The IOC accesses the FPGA registers with an EPICS asyn driver and lower level Linux driver, called the common platform software (CPSW). The MPS link nodes and central nodes use this driver stack to control the MPS system. The software stacks are described in the subsequent sections. The high level MPS software architecture is shown in Fig. 3.

LINK NODE SOFTWARE

The primary functions of the MPS link node are to collect data from applications that participate in MPS and send that data on to the central node. Additionally, the link node measures analog inputs and applies thresholds. The link node SIOC is therefore responsible for providing a platform to program threshold values, monitoring the status of the health of the link node, and applying an initial configuration to the hardware when the IOC boots up. The link node IOC is composed of a few software layers. On the bottom layer is the CPSW framework allowing the software to access firmware registers for threshold and status information. The link node library interfaces with CPSW and provides an API for the link node IOC via an asyn driver. The link node software stack is shown in Fig. 4.

CENTRAL NODE SOFTWARE

The primary function of the MPS central node is to collect data from the link nodes, evaluate the MPS algorithm, and send permits to mitigation devices. There are two processing chains in the central node: the fast chain that evaluates single bit logic in one clock cycle and the slow processing chain that evaluates more complicated logic with a 360 Hz update rate. The fast chain is evaluated in firmware and the slow chain is evaluated in software by the connected IOC. The central node IOC runs on the server connected to the central node hardware, and there is one central node IOC per central node application. The central node IOC is responsible for:

- Evaluation of the MPS algorithm at 360 Hz using slow link node inputs
- Provide read back of fast algorithm faults (all latching)
- Monitor the central node firmware health and status
- Configure the central node firmware logic configuration, input mapping, and bypass information
- Report changes to the MPS global state to the MPS History Server

Figures 5 and 6 show the central node software stack which is described in the subsections below.

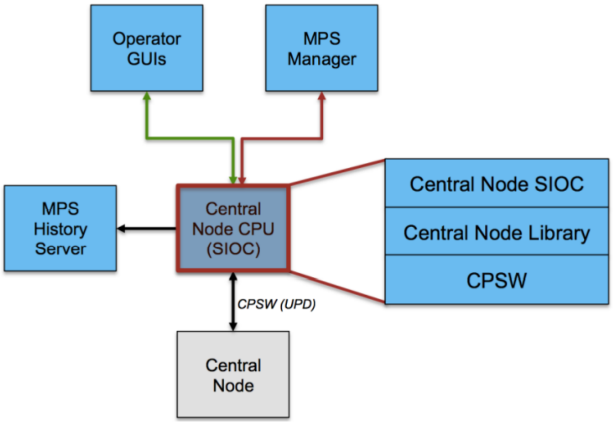


Figure 5: Central node IOC interfaces and internal software layers.

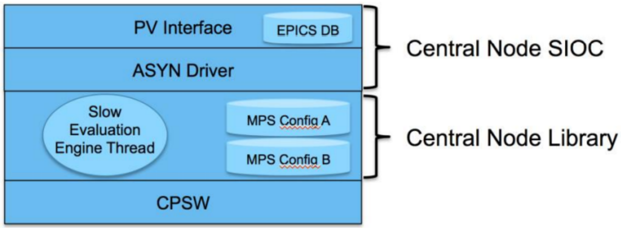


Figure 6: Central node IOC software layer details.

Central Node Configuration

The definition and description of all inputs and fault logic are exported from the configuration database (see next section) into a YAML database snapshot. A PV interface reads in this YAML snapshot and the central node engine translates it into the rules tables which are then loaded into the central node firmware for fast evaluation. Additionally, a PV interface is provided to display the current and latched status of all inputs and all logic tables. The system supports two

YAML snapshots of the database to be loaded into software and firmware memories, and the transition between configurations is minimally invasive to beam operations. The central node health and configuration are monitored and managed by the software with a CPSW layer and asyn PV API.

Slow Fault Evaluation

The central node FW maintains the current state of all MPS inputs based on periodic messages received from the link nodes. The states are updated at 1 MHz, and on every update the fast algorithm evaluates faults and generates fast mitigation responses as needed. The slow fault evaluation engine runs at a slower rate, 360 Hz, in the IOC software. On every IOC fault evaluation cycle, the software engine:

1. Receives and processes input states from the firmware
2. Evaluates inputs according to the active configuration, using bypass values if defined
3. Evaluate logic veto (ignore) conditions
4. Mitigate faults through the firmware (send mitigation back to firmware to be included in final permit)

Upon each iteration through the 360 Hz loop, the software engine determines the safest permit state for each rule according to the input state and the loaded configuration. The software engine compares the safest permit state against that computed for all logic tables to determine the overall safest permit state which is sent back to the firmware to be included in the overall permit state. The firmware compares the software-computed safest permit with the fast-evaluated safest permit to choose the most restrictive permit state and applies that to the next pulse.

Bypass Manager

The central node software is responsible for managing any MPS fault bypasses. Each bypass is defined with an expiration time and bypass value. The slow evaluation engine uses the bypassed value if defined in place of the actual input value to compute the algorithm. For the fast evaluation engine, the bypass manager removes the bypassed fault from the rules table and writes a new rules table to the firmware until the bypass expires. Then the bypass manager puts back the bypassed fault and rewrites the table again. Information about bypasses is presented to the control system with a PV interface.

CONFIGURATION MANAGEMENT

The MPS Database contains a complete representation of the currently deployed MPS hardware, a map of the beam line devices connected to each link node channel, and the logic defining the allowed beam conditions given a set of states at the link node input channels. Conceptually, there are two sections in the schema: the tables that map a link node digital and analog channel to a hardware device, and the tables that define logic for the input signals. A sample of the database schema that deals with logic mapping is shown in Fig. 7.

The MPS Configuration Database is stored as an SQLite format and handled through Python scripts that use the SQLAlchemy object relational mapper. The database tables are defined as classes in the Python language, allowing the database to be manipulated through scripts. It also provides mechanisms for checking database integrity. The control GUIs are also python processes, allowing a native integration of the configuration database into the GUI, allowing for fewer PVs to be published.

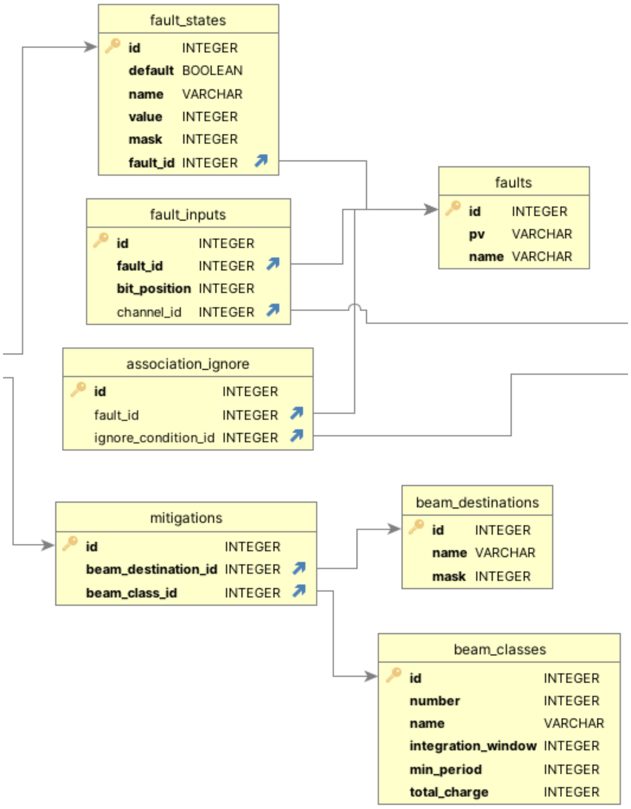


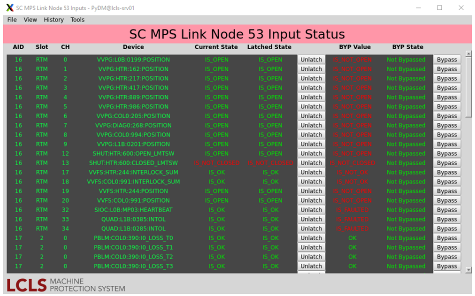
Figure 7: A portion of the MPS Database that stores information about faults.

The contents of the MPS Configuration Database are exported to a single YAML file per central node containing the tables and entries. The YAML export provides a tagged released snapshot of the database used by the central node IOC to load the MPS configuration inputs and logic.

The MPS Configuration Database is the source of information for all MPS inputs, and tools are provided to generate EPICS databases to be loaded by the central node and link node IOCs. All EPICS databases containing MPS configuration must be exported directly from the database using the software database tools. Additional scripts are used to generate alarms, basic control screens, and link node configurations.

CONTROL ROOM TOOLS

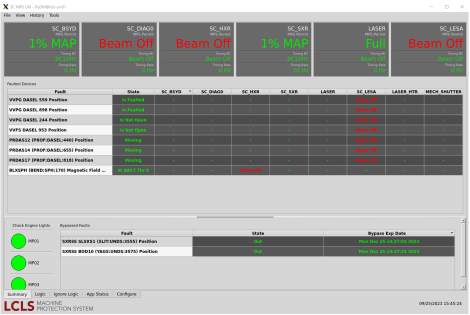
MPS provides various operator interfaces for use in the control room. The operator interfaces are written in python


SC MPS Link Node 53 Inputs - PyDM@lcls-srv01
File View History Tools
LCLS MACHINE PROTECTION SYSTEM

SC MPS Link Node 53 Input Status

AID	Slot	CH	Device	Current State	Latched State		BYP Value	BYP State	
16	RTM	0	VVPG:L0B:0199:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	1	VVPG:HTR:162:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	2	VVPG:HTR:217:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	3	VVPG:HTR:417:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	4	VVPG:HTR:889:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	5	VVPG:HTR:986:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	6	VVPG:COL0:205:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	7	VVPG:DIAG0:268:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	8	VVPG:COL0:994:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	9	VVPG:L1B:0201:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	12	SHUT:HTR:600:OPEN_LMTSW	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	13	SHUT:HTR:600:CLOSED_LMTSW	IS_NOT_CLOSED	IS_NOT_CLOSED	Unlatch	IS_NOT_CLOSED	Not Bypassed	Bypass
16	RTM	17	VVFS:HTR:244:INTERLOCK_SUM	IS_OK	IS_OK	Unlatch	IS_NOT_OK	Not Bypassed	Bypass
16	RTM	18	VVFS:COL0:991:INTERLOCK_SUM	IS_OK	IS_OK	Unlatch	IS_NOT_OK	Not Bypassed	Bypass
16	RTM	19	VVFS:HTR:244:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	20	VVFS:COL0:991:POSITION	IS_OPEN	IS_OPEN	Unlatch	IS_NOT_OPEN	Not Bypassed	Bypass
16	RTM	32	SIOC:L0B:MP03:HEARTBEAT	IS_OK	IS_OK	Unlatch	IS_FAULTED	Not Bypassed	Bypass
16	RTM	33	QUAD:L1B:0385:INTOL	IS_OK	IS_OK	Unlatch	IS_FAULTED	Not Bypassed	Bypass
16	RTM	34	QUAD:L1B:0285:INTOL	IS_OK	IS_OK	Unlatch	IS_FAULTED	Not Bypassed	Bypass
17	2	0	PBLM:COL0:390:I0_LOSS_T0	IS_OK	IS_OK	Unlatch	OK	Not Bypassed	Bypass
17	2	0	PBLM:COL0:390:I0_LOSS_T1	IS_OK	IS_OK	Unlatch	OK	Not Bypassed	Bypass
17	2	0	PBLM:COL0:390:I0_LOSS_T2	IS_OK	IS_OK	Unlatch	OK	Not Bypassed	Bypass
17	2	0	PBLM:COL0:390:I0_LOSS_T3	IS_OK	IS_OK	Unlatch	OK	Not Bypassed	Bypass

Figure 8: A portion of a link node input screen that shows the current and latched states of the inputs, as well as bypass states.



	SC_BSYD	SC_DIAG0	SC_HXR	SC_SXR	LASER	SC_LESA
MPS Permit	1% MAP	Beam Off	Beam Off	1% MAP	Full	Beam Off
Timing BC	BC10Hz	Beam Off	Beam Off	BC10Hz	Beam Off	Beam Off
Timing Rate	0 Hz	0 Hz	0 Hz	10 Hz	0 Hz	0 Hz

Faulted Devices

Fault	State	SC_BSYD	SC_DIAG0	SC_HXR	SC_SXR	LASER	SC_LESA	LASER_HTR	MECH_SHUTTER
VVPG DASEL 559 Position	Is Faulted	-	-	-	-	-	Beam Off	-	-
VVPG DASEL 898 Position	Is Faulted	-	-	-	-	-	Beam Off	-	-
VVPG DASEL 244 Position	Is Not Open	-	-	-	-	-	Beam Off	-	-
VVFS DASEL 953 Position	Is Not Open	-	-	-	-	-	Beam Off	-	-
PRDAS12 (PROF:DASEL:440) Position	Moving	-	-	-	-	-	Beam Off	-	-
PRDAS14 (PROF:DASEL:655) Position	Moving	-	-	-	-	-	Beam Off	-	-
PRDAS17 (PROF:DASEL:818) Position	Moving	-	-	-	-	-	Beam Off	-	-
BLXSPH (BEND:SPH:170) Magnetic Field ...	I0_BACT Thr 0	-	-	Beam Off	-	-	-	-	-

Bypassed Faults

Fault	State	Bypass Exp Date
SXRSS SLSXS1 (SLIT:UNDS:3555) Position	Out	Mon Dec 25 14:37:02 2023
SXRSS BOD10 (YAGS:UNDS:3575) Position	Out	Mon Dec 25 14:37:33 2023

09/25/2023 15:45:24

Figure 9: The interactive MPS GUI developed for LCLS-II.

to take advantage of the direct integration of the python libraries that define the configuration database. The link nodes and central nodes provide a complete diagnostic interface that can be used to understand problems and verify proper system operation. Additionally, each link node has an interface that shows the current and latched states of each input, as well as their bypass states. An example screen is shown in Fig. 8.

An interactive graphical interface provides the operators a live look at the current permits. The GUI also provides searchable and sort-able logic tables, information about logic ignore devices, and bypass information and expiration dates. Additionally, the GUI has a mode to increase font size and contrast so it can be displayed continuously on monitors in the control room to give heads up display to the current state of the system. Figure 9 shows the front page of the GUI. The top section shows the current permits issued for each destination. The middle section shows what is currently faulted, sorted by destination, and the bottom section shows the bypasses currently implemented and their expiration dates. This GUI takes full advantage of the python libraries available to describe the configuration database. The central node publishes a single EPICS PV per fault which the GUI can read and use to reconstruct the rest of the information by querying the configuration database directly.

Finally, the MPS provides a history server to save fault data for future debugging and posterity. The central node publishes all fault data to an external listener which writes each fault as a line in a time series database, currently MongoDB. To save bandwidth, the history listener reconstructs as much information as possible from the MPS configuration database, requiring the central nodes to send only one value associated with a fault which the history server can use to query the configuration. The control room tools then provide a facility to monitor the history in real time or go back and search for faults or other events that may have happened.

CONCLUSION

The LCLS-II project installed a new, high power accelerator into the SLAC linear accelerator facility that required the development and installation of a new Machine Protection System. The LCLS-II MPS is built with the EPICS toolkit and required the development of new software drivers and control room tools. To save bandwidth and processing time, the IOCs publish fewer PVs and the control room tools use a configuration database to reconstruct fault events. Overall, the system and tools have demonstrated their ability to protect the machine and provide the necessary utility to be operable.

REFERENCES

[1] LCLS-II Project Team, "LCLS-II Final Design Report", SLAC, Menlo Park, CA, USA, Rep. LCLSII-1.1-DR-0251-R0, 2015.

[2] A. S. Fisher, R. C. Field, and L. Y. Nicolas, "Evaluating Beam-Loss Detectors for LCLS-2", in *Proc. IBIC'16*, Barcelona, Spain, Sep. 2016, pp. 678–681. doi:10.18429/JACoW-IBIC2016-WEPG23

[3] A. S. Fisher *et al.*, "Commissioning Beam-Loss Monitors for the Superconducting Upgrade to LCLS", in *Proc. IBIC'22*, Kraków, Poland, Sep. 2022, pp. 207–210. doi:10.18429/JACoW-IBIC2022-TU2C3

[4] A.S. Fisher, N. Balakrishnan, G.W. Brown, E.P. Chin, W.G. Cobau, J.E. Dusatko, *et al.*, "Commissioning the Beam-Loss Monitoring System of the LCLS Superconducting Linac", in *Proc. 12th Int. Beam Instrum. Conf. (IBIC'23)*, Saskatoon, Canada, Sep. 2023, pp. 188–191. doi:10.18429/JACoW-IBIC2023-TUP005

[5] J.C. Frisch *et al.*, "A FPGA Based Common Platform for LCLS2 Beam Diagnostics and Controls", in *Proc. IBIC'16*, Barcelona, Spain, Sep. 2016, pp. 651–654. doi:10.18429/JACoW-IBIC2016-WEPG15

[6] J.A. Mock, A.S. Fisher, R.T. Herbst, P. Krejcik, and L. Sapozhnikov, "Commissioning of the LCLS-II Machine Protection System for MHz CW Beams", in *Proc. 12th Int. Beam Instrum. Conf. (IBIC'23)*, Saskatoon, Canada, Sep. 2023, pp. 155–160. doi:10.18429/JACoW-IBIC2023-TU3I01

[7] J.A. Vásquez, J.M. D'Ewart, K.H. Kim, T. Straumann, and E. Williams, "YCPSWASYN: EPICS Driver for FPGA Register Access and Asynchronous Messaging", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 1707–1709. doi:10.18429/JACoW-ICALEPCS2017-THPHA138

[8] L. R. Dalesio, M. R. Kraimer, and A. J. Kozubal, "EPICS Architecture", in *Proc. ICALEPCS'91*, Tsukuba, Japan, 1991.