EPICS: CSS-Phoebus

Peter Bonneau
2022-09

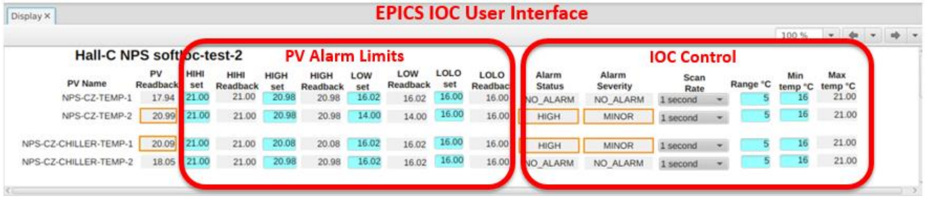
EPICS Alarm System in Phoebus

I am developing an EPICS alarm system based on CS-Studio Phoebus. Phoebus will be used for new EPICS system development and will replace the existing Eclipse-based CS-Studio systems as detailed in my note DSG Note 2021-37 and talk DSG Talk 2021-17.

The Phoebus alarm system monitors process variables (PVs) from EPICS Input / Output Controllers (IOCs) for alarm conditions. I developed an host-based softIOC detailed in DSG Note 2022-06 to perform an integrated alarm system software test of the development work I have accomplished.

I designed this softIOC to test my integrated alarm software components discussed in my software memos 2022-01, 2022-02, 2022-03, and 2022-04. Figure 2 shows the sections of alarm system code being tested by the softIOC I have designed and implemented. When the remaining sections of the Phoebus alarm system code shown in Figure 2 has been designed, debugged, and implemented, these code sections will also be tested by the alarm test system softIOC.

I developed tests to verify the proper operation of specific programs in the alarm system code. Using the EPICS IOC user interface I also developed (Figure 1), I used the process variable *IOC control* and the *PV Alarm Limits* to generate the conditions needed for the tests.



PV Name	PV Readback	HIHI set	HIHI Readback	HIGH set	HIGH Readback	LOW set	LOW Readback	LOLO set	LOLO Readbac
NPS-CZ-TEMP-1	17.94	21.00	21.00	20.98	20.98	16.02	16.02	16.00	16.00
NPS-CZ-TEMP-2	20.99	21.00	21.00	20.98	20.98	14.00	14.00	16.00	16.00
NPS-CZ-CHILLER-TEMP-1	20.09	21.00	21.00	20.08	20.08	16.02	16.02	16.00	16.00
NPS-CZ-CHILLER-TEMP-2	18.05	21.00	21.00	20.98	20.98	16.02	16.02	16.00	16.00

PV Name	Alarm Status	Alarm Severity	Scan Rate	Range °C	Min temp °C	Max temp °C
NPS-CZ-TEMP-1	NO_ALARM	NO_ALARM	1 second	5	16	21.00
NPS-CZ-TEMP-2	HIGH	MINOR	1 second	5	16	21.00
NPS-CZ-CHILLER-TEMP-1	HIGH	MINOR	1 second	5	16	21.00
NPS-CZ-CHILLER-TEMP-2	NO_ALARM	NO_ALARM	1 second	5	16	21.00

FIG.1. EPICS IOC User Interface

- **Developing CS-Studio Phoebus based controls, monitoring, and alarm system - to be implemented on Hall C detectors**
- **Testing the Phoebus integrated alarm code infrastructure**
- **Plan to develop a high PV signal count IOC for the Phoebus test station**

EPICS: CSS-Phoebus

The Phoebus alarm test system user interface (Figure 3), combines both the EPICS IOC control with the displays for alarm summary, alarm hierarchical tree, unacknowledged alarms, and acknowledged alarms. I designed the combined displays to enable the user of the Phoebus alarm test system to see an overall response view of the alarm system while controlling the IOC PVs which generate the alarms.

I developed and implemented the alarm server monitoring program and the Kafka streaming message monitoring program for debugging and to allow the user of the test system to see the bidirectional data. The bidirectional data from the Kafka programs and the alarm server can not be read from any of the Phoebus programs directly. A Linux terminal window is used as the user interface for the alarm server monitoring program and the Kafka streaming message monitoring program. The synopsis for each of the alarm system tests is summarized in Table 1.

Program Name	Program Function Summary	Test Summary
Alarm server	Monitors EPICS process variables (PVs) for alarm conditions via channel access. Stores alarm configuration settings for each PV.	Verified the monitoring of PVs from test IOC, alerts users and latches PV value and time upon an alarm condition. Verified monitoring and UI settings /read-back values for alarm severity, PV alarms on HIHI, HIGH, LOW, LOLO conditions. Verified alarm acknowledgment, PV alarm configuration, PV status indicators.
Alarm server monitoring	Monitors the health of the alarm server	Verified server correctly reports PV alarm configurations stored in server
proServ	Provides remote access to the alarm server command console via Telnet	Verified Telnet connection to alarm server command console.
Kafka Zookeeper	Kafka cluster system management	Verified correct management of the server and generation of logfiles.
Kafka server	Hosts the alarm system message streams	Verified correct generation of the three alarm system Kafka streams.
Kafka message monitoring	Monitors the health of the Kafka system	Verified correct syntax and operation of the state, configuration, and command Kafka streams.
Phoebus alarm test system user interface	User alarm monitoring and system configuration	Verified correct UI control and monitoring of IOC and Phoebus alarm system.

Table 1. Program Test Summary

I plan to design and implement a high PV signal count IOC for the test station as the next step in the development of the Phoebus alarm system.

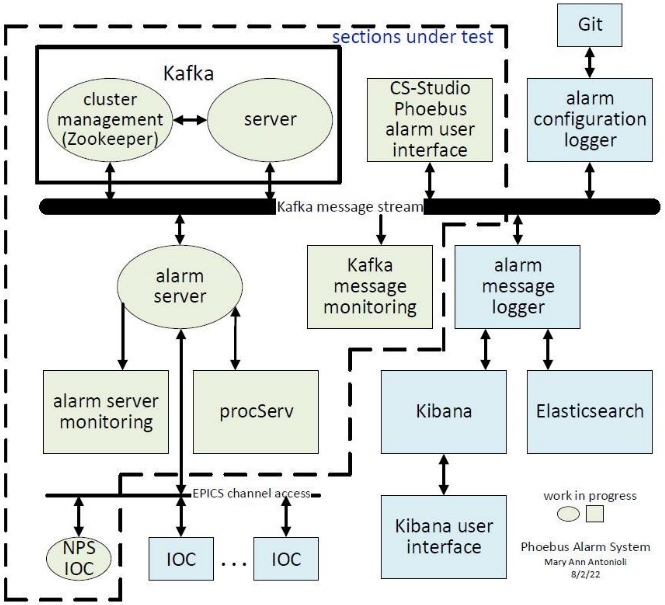
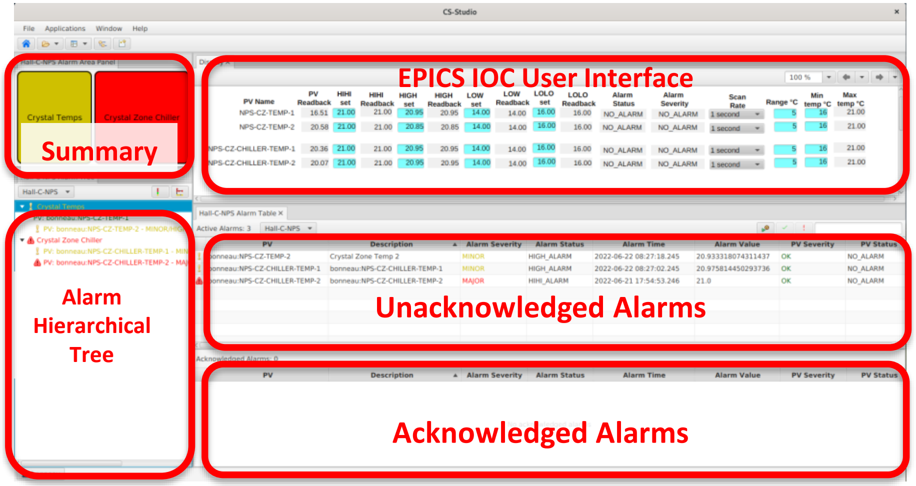


FIG.2. Sections of Alarm System code under test



PV Name	PV Readback	HIHI set	HIHI Readback	HIGH set	HIGH Readback	LOW set	LOW Readback	LOLO set	LOLO Readback	Alarm Status	Alarm Severity	Scan Rate	Range °C	Min temp °C	Max temp °C
NPS-CZ-TEMP-1	16.51	21.00	21.00	20.95	20.95	14.00	14.00	16.00	16.00	NO_ALARM	NO_ALARM	1 second	5	16	21.00
NPS-CZ-TEMP-2	20.58	21.00	21.00	20.85	20.85	14.00	14.00	16.00	16.00	NO_ALARM	NO_ALARM	1 second	5	16	21.00
NPS-CZ-CHILLER-TEMP-1	20.36	21.00	21.00	20.95	20.95	14.00	14.00	16.00	16.00	NO_ALARM	NO_ALARM	1 second	5	16	21.00
NPS-CZ-CHILLER-TEMP-2	20.07	21.00	21.00	20.95	20.95	14.00	14.00	16.00	16.00	NO_ALARM	NO_ALARM	1 second	5	16	21.00

PV	Description	Alarm Severity	Alarm Status	Alarm Time	Alarm Value	PV Severity	PV Status
bonneau:NPS-CZ-TEMP-2	Crystal Zone Temp 2	MINOR	HIGH_ALARM	2022-06-22 08:27:18.245	20.933318074311437	OK	NO_ALARM
bonneau:NPS-CZ-CHILLER-TEMP-1	bonneau:NPS-CZ-CHILLER-TEMP-1	MINOR	HIGH_ALARM	2022-06-22 08:27:02.245	20.975814450293736	OK	NO_ALARM
bonneau:NPS-CZ-CHILLER-TEMP-2	bonneau:NPS-CZ-CHILLER-TEMP-2	MAJOR	HIHI_ALARM	2022-06-21 17:54:53.246	21.0	OK	NO_ALARM

PV	Description	Alarm Severity	Alarm Status	Alarm Time	Alarm Value	PV Severity	PV Status

FIG.3. User Interface for the Phoebus Alarm Test System