Development of the EPICS SoftIOC for Phoebus Alarm System Testing

Aaron Brown, Mary Ann Antonioli, Peter Bonneau, Pablo Campero, Brian Eng, George Jacobs, Mindy Leffel,
Tyler Lemon, Marc McMullen, and Amrit Yegneswaran

Physics Division, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606

April 19, 2023

This note details the development of an EPICS softIOC for testing the Phoebus alarm system being developed for the Hall C Neutral Particle Spectrometer (NPS).

In the EPICS Phoebus alarm system under development, when a process variable (PV) enters an alarm state, a graphical user interface (GUI) alerts the user of the alarm. The user can acknowledge the alarm and view details, such as the alarming PV value and alarm limits, as well as view guidance on how to respond to the alarm [1].

To test whether the Phoebus alarm system can handle the number of PVs that are to be used for the NPS control and monitoring program, a system test has been developed for the crystal temperatures. The system test required the creation of an EPICS softIOC on the Linux machine that hosts the PVs, to generate test PVs and provide random values for the PVs. The EPICS softIOC simulates detector signals for testing and transmits the signals as PVs to clients via EPICS Channel Access [2].

The Python program *db-test-1.py* was developed to create database records for all 112 crystal temperatures with sensors, (56 each on the front and back faces), to generate all crystal zone analog input PVs and to set for each alarm type their alarm limits, scan rate, and the severity, Fig. 1.

```
dbLines = [
'record(ai, "pvname")',
'{',
'        field(DESC, "Temp at crystal")',
'        field(SCAN, "1 second")',
'        field(HIHI, "30")',
'        field(HIGH, "28")',
'        field(LOW, "20")',
'        field(LOLO, "18")',
'        field(HHSV, "MAJOR")',
'        field(HSV, "MINOR")',
'        field(LSV, "MINOR")',
'        field(LLSV, "MAJOR")',
'}']
```

FIG. 1. Screenshot of portion of *db-test-1.py* program used to generate database records for crystal zone temperatures.
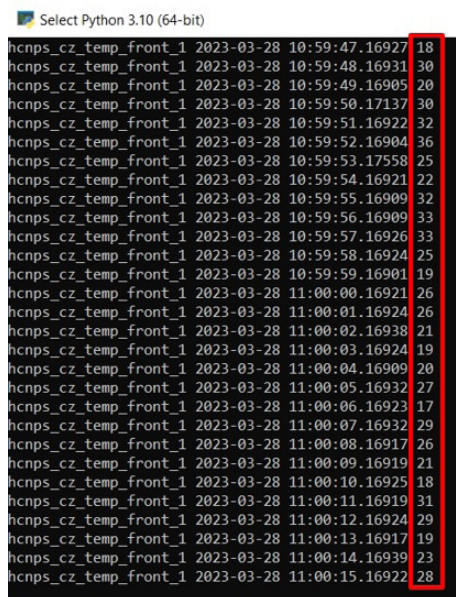
If the

$$\text{PV value} \in (\text{LOLO, LOW}] \lor [\text{HIGH, HIHI}),$$

the softIOC sets the PV alarm status field (STAT) to HIGH or LOW and sets the severity field (SEVR) to MINOR. If the

$$\text{PV value} \leq \text{LOLO} \lor \text{PV value} \geq \text{HIHI},$$

the softIOC sets STAT to HIHI or LOLO and sets SEVR to MAJOR. If the PV value is within limits, both STAT and SEVR are set to NO_ALARM by the softIOC.

Python program *randomTestScript.py* was developed to provide random values to the crystal temperature PVs, Fig. 2, using the Python package *Random* to generate uniform random numbers between 15 and 35 (which represent the temperature in °C) and using the Python package *Pyepics* EPICS Channel Access command *caput* to assign the random temperature values to the PVs.

FIG. 2. Screenshot of monitored test PVs with randomly generated temperature values (boxed in red).

In the current version, *RandomTestScript.py* is run independently as it employs an infinite while loop that only terminates when the program receives the *KeyboardInterrupt* command (Ctrl + C).

In summation, an EPICS softIOC has been developed and used to test the crystal temperature control and monitoring section of the NPS Phoebus alarm system.

[1] P. Bonneau et al., *Proposal to Implement Alarm System in Control System Studio Phoebus for the Hall C Neutral Particle Spectrometer*, DSG Note 2021-37, 2021.

[2] P. Bonneau et al., *Development of the EPICS Software Input/Output Controller for Testing the Phoebus Alarm System of the Hall C Neutral Particle Spectrometer*, DSG Note 2022-06, 2022.