EPICS: CSS-Phoebus

Peter Bonneau
2022-07

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. To perform an integrated alarm system software test of the development work I have accomplished, an IOC is required. I have designed and implemented an IOC to collectively test my alarm software components discussed in my memos 2022-01, 2022-02, 2022-03, and 2022-04. The IOC is part of the Phoebus alarm test system (FIG.1) and runs on the EPICS base system I configured, debugged, and tested. The EPICS base work is discussed in my 2022-06 memo.

I developed the IOC database for the Phoebus alarm test system using the EPICS Visual Database Configuration Tool program (VisualDCT). I downloaded the source code from GitHub site, compiled the code, and configured the environment variables for VisualDCT. I wrote a database

- **Developing CS-Studio Phoebus based controls, monitoring, and alarm system - to be implemented on Hall C detectors**

- **Developing an IOC to test the Phoebus integrated alarm code infrastructure.**

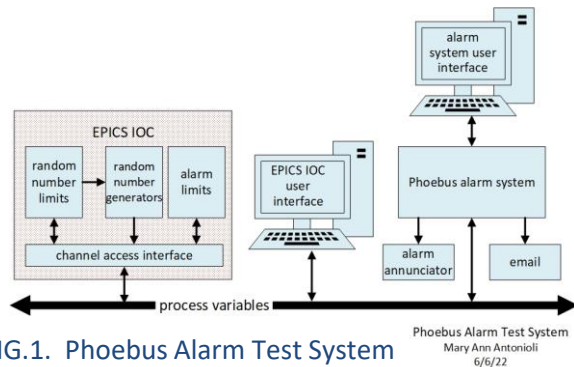- **Plan to use the IOC to test the Phoebus alarm code infrastructure.**

FIG.1. Phoebus Alarm Test System

definition (.dbd) file which must be provided at the start of the VisualDCT design environment. The .dbd file defines the specifications of the EPICS record types I used in the IOC database design.

I designed the IOC to simulate detector signals by generating PVs and transmitting them to EPICS clients via channel access. The transmitted data for each PV hosted by the IOC include the simulated signal value and the alarm fields which will be used to the Phoebus alarm code infrastructure.

I programmed via VisualDCT, an EPICS calculation (calc) record configured as a random number generator with user-defined control limits for each simulated PV. The simulated PV signal range, minimum value, and data scan rate is controlled via my EPICS IOC user interface, FIG. 2. The simulated PV signal controls I programmed via the VisualDCT design environment, FIG. 3, allows the testing of all alarm system parameters.
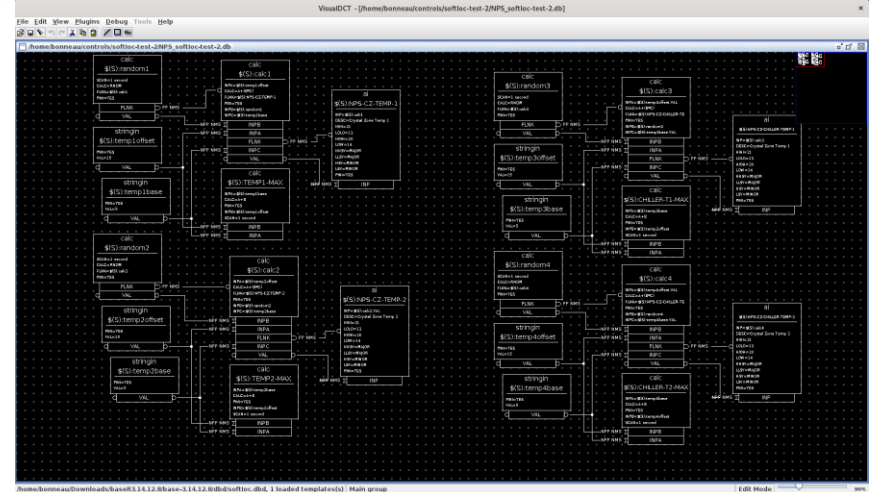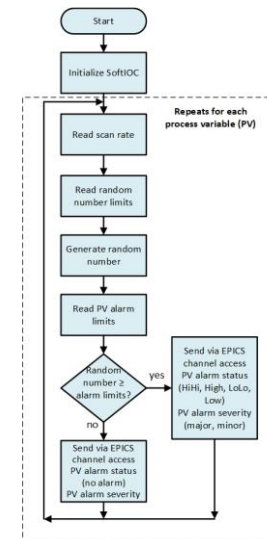
Hall-C NPS softloc-test-2

PV Name	PV Readback	HIHI set	HIHI Readback	HIGH set	HIGH Readback	LOW set	LOW Readback	LOLO set	LOLO Readback	Scan Rate	Range °C	Min temp °C	Max temp °C
NPS-CZ-TEMP-1	20.85	21.00	21.00	20.98	20.98	15.02	15.02	15.00	15.00	1 second	6	15	21.00
NPS-CZ-TEMP-2	20.96	21.00	21.00	20.98	20.98	15.02	15.02	15.00	15.00	1 second	6	15	21.00
NPS-CZ-CHILLER-TEMP-1	17.23	21.00	21.00	20.98	20.98	15.02	15.02	15.00	15.00	1 second	6	15	21.00
NPS-CZ-CHILLER-TEMP-2	15.97	21.00	21.00	20.98	20.98	15.02	15.02	15.00	15.00	1 second	6	15	21.00

FIG.2. EPICS IOC User Interface

The IOC I developed also contains the alarm limits for each simulated detector PV. When the PV record is scanned by the IOC, the EPICS alarm limits is compared with the PV value. If the PV value is outside of the EPICS *HIGH* or *LOW* limits, the IOC sets the PV alarm status field (*STAT*) to *HIGH* or *LOW* and sets the alarm severity field (*SEVR*) to *MINOR*. If the PV value is outside of the *HIHI* or *LOLO* limits, the IOC sets the PV alarm status field to *HIHI* or *LOLO* and sets the alarm severity field to *MAJOR*. If the PV value is within all limits, both the alarm status and alarm severity fields are set to *NO_ALARM* by the IOC.

The PV value, alarm status, and severity fields generated by my IOC will be used to test the Phoebus alarm system. The EPICS IOC flowchart is shown in FIG. 4.

I plan use the IOC to test the alarm code infrastructure I developed as the next step in the development of the Phoebus alarm system.

FIG.3. IOC VisualDCT design environment

FIG.4. EPICS IOC flowchart