

SoftIOC Development for Phoebus Alarm System Testing

Aaron Brown
2022-03

An EPICS Phoebus alarm system is under development. To test that the Phoebus alarm system can handle the number of PVs that will be used for the NPS control and monitoring program, we are developing a full system test that will simulate all of the NPS PVs. This requires the creation of a softIOC to generate the test PVs and provide random numbers for these PVs.

I started by creating the softIOC on *dsg-c-linux1*, the Linux machine that will host the PVs to be used for testing. This entailed downloading the required configuration, database, and make files to a new folder I named *npsIOC*.

I then created a Python program (*db-test-1.py*) to create the database records for all 112 crystal temperatures (56 front and 56 back), Fig. 1. This program generates all of the crystal zone analog input (ai) PVs and sets their fields (alarm limits, scan rate, and the severity of each alarm type).

```
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

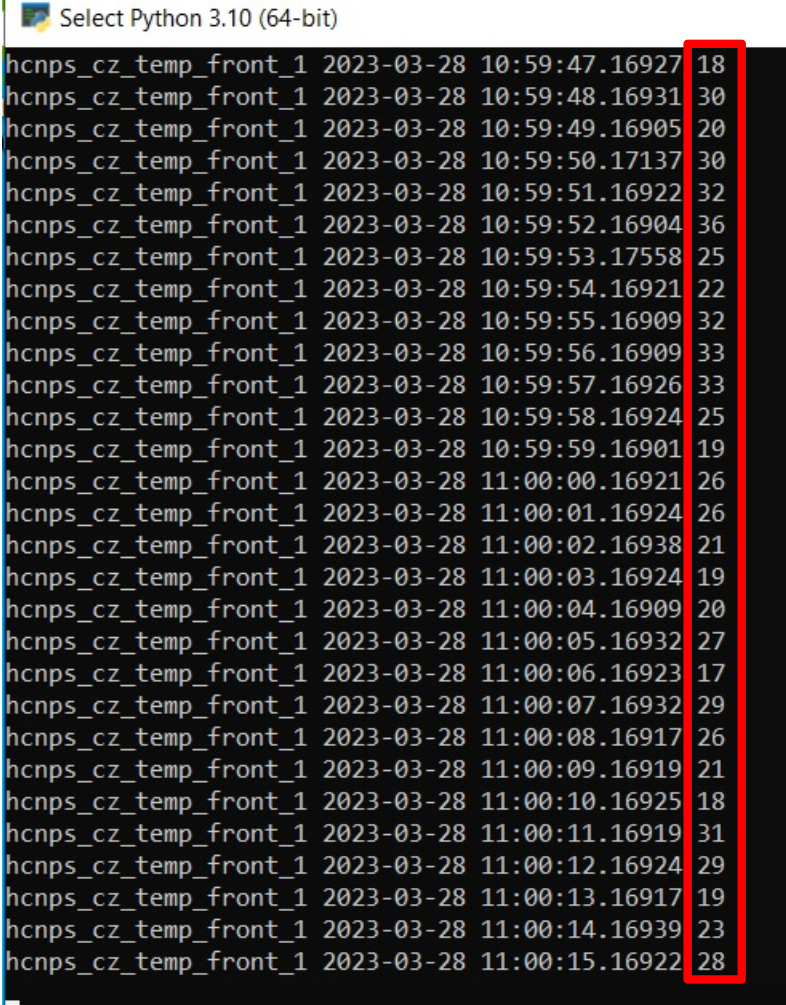
- **Created test softIOC for NPS Phoebus Alarm System testing**
- **Developed Python program to programmatically generate database records for 112 crystal zone temperatures**
- **Developed Python program to continuously generate 112 random temperature values**

SoftIOC Development for Phoebus Alarm System Testing

I then generated an additional Python program (*randomTestScript.py*) to provide random numbers to the crystal temperature PVs. This program uses the random Python package to generate uniform random numbers between 15 and 36 (to represent degrees Celsius) and the Pyepics Python package to use the Channel Access command *caput* to assign these random temperature values to the PVs, Fig. 2.

Currently, the softIOC runs as a background process on the *dsg-c-linux1* Linux machine with each PV being scanned once every second. The Python program, *randomTestScript.py* has to be run separately as it employs an infinite while loop that terminates once the program receives the *KeyboardInterrupt* command (Ctrl + C).

Eventually, I will create database records for all of the NPS PVs (not just the crystal zone temperatures), alter the random number generation program to provide them with random numbers as well, and add them to the softIOC. The next step once all of the PVs have been added to the softIOC is to create a script that will start the softIOC and random number generating programs simultaneously as one background process. This is a work in progress.



```
Select Python 3.10 (64-bit)
hcnps_cz_temp_front_1 2023-03-28 10:59:47.16927 18
hcnps_cz_temp_front_1 2023-03-28 10:59:48.16931 30
hcnps_cz_temp_front_1 2023-03-28 10:59:49.16905 20
hcnps_cz_temp_front_1 2023-03-28 10:59:50.17137 30
hcnps_cz_temp_front_1 2023-03-28 10:59:51.16922 32
hcnps_cz_temp_front_1 2023-03-28 10:59:52.16904 36
hcnps_cz_temp_front_1 2023-03-28 10:59:53.17558 25
hcnps_cz_temp_front_1 2023-03-28 10:59:54.16921 22
hcnps_cz_temp_front_1 2023-03-28 10:59:55.16909 32
hcnps_cz_temp_front_1 2023-03-28 10:59:56.16909 33
hcnps_cz_temp_front_1 2023-03-28 10:59:57.16926 33
hcnps_cz_temp_front_1 2023-03-28 10:59:58.16924 25
hcnps_cz_temp_front_1 2023-03-28 10:59:59.16901 19
hcnps_cz_temp_front_1 2023-03-28 11:00:00.16921 26
hcnps_cz_temp_front_1 2023-03-28 11:00:01.16924 26
hcnps_cz_temp_front_1 2023-03-28 11:00:02.16938 21
hcnps_cz_temp_front_1 2023-03-28 11:00:03.16924 19
hcnps_cz_temp_front_1 2023-03-28 11:00:04.16909 20
hcnps_cz_temp_front_1 2023-03-28 11:00:05.16932 27
hcnps_cz_temp_front_1 2023-03-28 11:00:06.16923 17
hcnps_cz_temp_front_1 2023-03-28 11:00:07.16932 29
hcnps_cz_temp_front_1 2023-03-28 11:00:08.16917 26
hcnps_cz_temp_front_1 2023-03-28 11:00:09.16919 21
hcnps_cz_temp_front_1 2023-03-28 11:00:10.16925 18
hcnps_cz_temp_front_1 2023-03-28 11:00:11.16919 31
hcnps_cz_temp_front_1 2023-03-28 11:00:12.16924 29
hcnps_cz_temp_front_1 2023-03-28 11:00:13.16917 19
hcnps_cz_temp_front_1 2023-03-28 11:00:14.16939 23
hcnps_cz_temp_front_1 2023-03-28 11:00:15.16922 28
```

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