SoftIOC Development for Phoebus Alarm System Testing

Aaron Brown
2022-05

In a previous memo [1], I detailed the creation of the first versions of the alarm system test softIOC and random number generating Python program. At the time, the only PVs generated were for all of the 112 crystal zone temperatures (56 front and 56 back). I created a new Python program (*db-test-2.py*) to create all of the ~180 database records needed to simulate all of the NPS signals.

This program specifies the PV names and the associated fields for each signal (alarm limits, scan rate, and the severity of each alarm type). Also built into this softIOC are four additional PVs that were created specifically for the test system: three input PVs for the scan rate, range, and minimum value of the random number and one PV to calculate and display the maximum value of the random number. This brings the total number of PVs generated by the softIOC to 444.

```
try:
    while True:
        j = 0
        for i in range(len(PVs)):
            rang = caget(rminPVs[j])
            j = j + 1
            rmin = caget(rminPVs[j])
            rmax = rang + rmin + 1
            x = "%.2f" % random.uniform(rmin, rmax)
            val = caput(PVs[i],float(x))
            j = j + 1
except:
    print("Interrupted")
```

FIG.1. Screenshot of portion of randomTestScript.py Python program

- **Completed development of softIOC for NPS Phoebus Alarm System testing**
- **Developed Python program to randomly generate values for all simulated signals**
- **User decides the minimum and range for each random number**

SoftIOC Development for Phoebus Alarm System Testing

I have haveredone the Python program *randomTestScript.py*, Fig. 1, used to provide all of the random numbers. This program creates an array of all of the signal PVs (called PVs) and an array of the random number control PVs (called rminPVs). The program then initializes the range and minimum values for each signal PV to 0.0; these can be changed by the user via the alarm testing Phoebus screen, Fig. 2.

Next, the program sends the Channel Access command caget to retrieve the value for the range of the random number and a second caget command to retrieve the minimum value of the random number. These two values are used to calculate the maximum value for the random number: range + minimum + 1 = maximum (a one is added so that the maximum value is included in the possible random number values). Finally, the program generates a random number for the signal PV using the minimum and the maximum as the input parameters for the *random.uniform* Python command from the random Python package. This was done so that the range for the random number assigned to each signal PV can be controlled by the user. In the first version of this program, the random values for all signal PVs had the same range and minimum value.

The next step is to add in the ability for the user to change the scan rate for the random value. The PV exists, but at the moment changing the scan rate does nothing.

[1] Brown, Aaron DSG Monthly Memo 2023-03

2023-05-30 09:15:

Detector Frame Alarm Testing

PV name	Sensor	read	HIHI set	HIHI read	HIGH set	HIGH read	LOW set	LOW read	LOLO set	LOLO read	Alarm status	Alarm severity	Scan rate	range [°C]	Min T [°C]	Max T [°C]
Temperature																
hcnps_intlk_df_t	frame 1	20.75	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1 second	10.00	20.00	30.00
	frame 2	21.64	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1 second	10.00	20.00	30.00
	frame 3	28.35	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	HIGH	MINOR	1 second	10.00	20.00	30.00
	frame 4	22.31	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1 second	10.00	20.00	30.00
Humidity																
hcnps_intlk_df_rh	frame 1	24.24	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1 second	10.00	20.00	30.00
	frame 2	29.36	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	HIGH	MINOR	1 second	10.00	20.00	30.00
	frame 3	25.50	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1 second	10.00	20.00	30.00
	frame 4	25.99	30.00	30.00	28.00	28.00	20.00	20.00	18.00	18.00	NO_ALARM	NO_ALARM	1 second	10.00	20.00	30.00
Dew point																
hcnps_intlk_df_dp	frame 1	24.91	30.00	30.00	28.00	28.00	20.00	20.00	18	18.00	NO_ALARM	NO_ALARM	1 second	10.00	20.00	30.00
	frame 2	24.76	30.00	30.00	28.00	28.00	20.00	20.00	18	18.00	NO_ALARM	NO_ALARM	1 second	10.00	20.00	30.00
	frame 3	30.04	30.00	30.00	28.00	28.00	20.00	20.00	18	18.00	HIHI	MAJOR	1 second	10.00	20.00	30.00
	frame 4	29.25	30.00	30.00	28.00	28.00	20.00	20.00	18	18.00	HIGH	MINOR	1 second	10.00	20.00	30.00

FIG.2. Screenshot of Detector Frame Alarm Testing Phoebus screen using random values from *randomTestScript.py* Python program