Configuration File Control for Hardware Interlock LabVIEW Program

Aaron Brown
2022-06

Configuration File Control for the NPS Hardware Interlock LabVIEW Program

I am working on creating a configuration file control VI for the NPS hardware interlock LabVIEW program. The configuration file contains all of the set point values for the hardware interlock system.

When the hardware interlock LabVIEW program is run for the first time, this configuration file needs to be read, parsed, and used to set various parameters as well as specifying which sensors and values to use.

There are six things that the configuration file control VI needs to be able to accomplish: read the current configuration file, create a new configuration file, create a backup configuration file, copy a configuration file, delete a configuration file, and check the status of the cRIO's SD card where the configuration file is stored.

I wrote a Python program to generate the initial configuration file, configTest.py. This program generates a .csv file that contains the set point values for the NPS hardware interlock system in a 27 x 222 grid. I first started with reading in the configuration file.

The LabVIEW program checks to make sure that the configuration file is valid (contains data), records the date and time the file was created, and records the size (in bytes) of the file.

- **Developed Python program to generate configuration file for hardware interlock system**
- **Creating LabVIEW VI to read configuration file and pass values to parameter variables**
- **VI will be used to read, create, copy, backup, and check the status of configuration files**

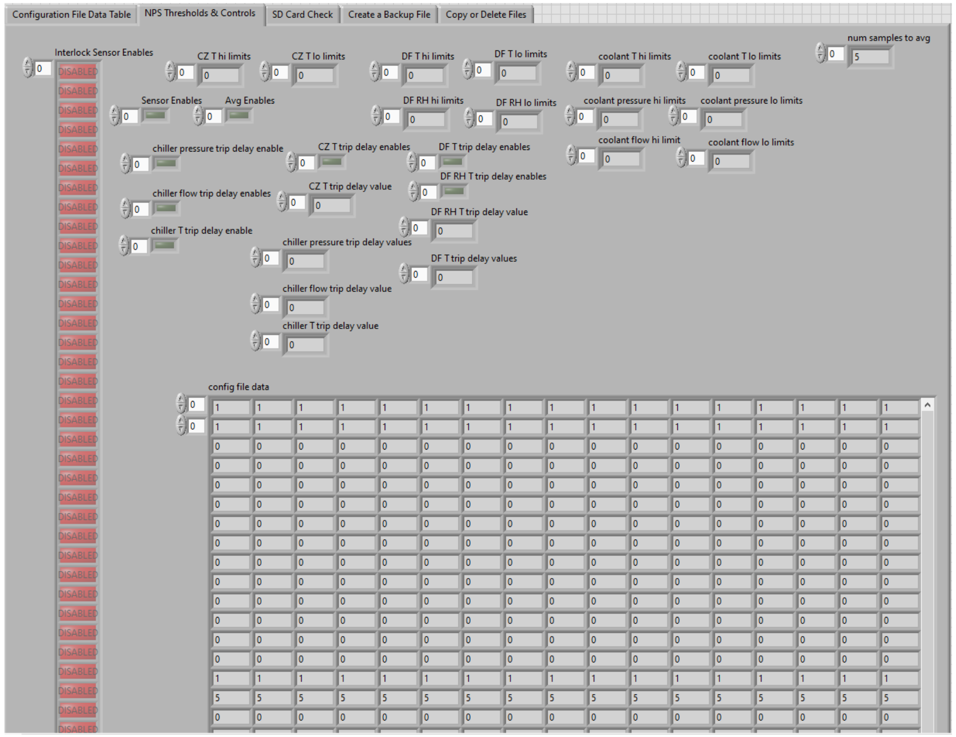
Configuration File Control for Hardware Interlock LabVIEW Program

Next the program parses the configuration file row by row and displays the data on the front panel via indicators for each individual parameter.

One problem that I encountered was when I was initially reading in the configuration file, the program was only retaining one column of data. This was because I had the delimiter set to tabs instead of the comma that was being used in the configuration file generated with the Python program configTest.py. One this was corrected, I was able to read in and display the entire configuration file.

Another problem that is currently being debugged is that the data from the configuration file is not being passed to some of the parameter variables properly. I am still in the process of debugging this issue.

As of now, I am able to read in the data from the configuration file and check to verify that the file is not corrupted and actually contains data. The next steps will be to resolve the issue with passing the data values to all of the corresponding parameter variables, and then work on the remaining portions of the VI that can create a new configuration file, create a backup configuration file, copy a configuration file, delete a configuration file, and check the status of the cRIO's SD card where the configuration file is stored.



Interlock Sensor Enables: 0
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED
DISABLED

CZ T hi limits: 0 0
CZ T lo limits: 0 0
DF T hi limits: 0 0
DF T lo limits: 0 0
coolant T hi limits: 0 0
coolant T lo limits: 0 0
num samples to avg: 0 5
Sensor Enables: 0
Avg Enables: 0
DF RH hi limits: 0 0
DF RH lo limits: 0 0
coolant pressure hi limits: 0 0
coolant pressure lo limits: 0 0
chiller pressure trip delay enable: 0
CZ T trip delay enables: 0
DF T trip delay enables: 0
coolant flow hi limit: 0 0
coolant flow lo limits: 0 0
DF RH T trip delay enables: 0
chiller flow trip delay enables: 0
CZ T trip delay value: 0 0
DF RH T trip delay value: 0 0
chiller T trip delay enable: 0
chiller pressure trip delay values: 0 0
DF T trip delay values: 0 0
chiller flow trip delay value: 0 0
chiller T trip delay value: 0 0

config file data: 0, 0

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIG. 1. Screenshot of Configuration File Control VI