Handling Initialization of NI LabVIEW Controls

Brian Eng
2022-06

Handling Initialization of NI LabVIEW Controls

The Hall B Gas system utilizes National Instruments (NI) CompactRIO (cRIO) controllers to handle reading various sensors and communications with other instruments, e.g. scales, mass flow controllers, etc. As the software was developed the various sub-virtual instruments (Vis [roughly equivalent to functions in other languages]) each handle initializing any values they use on their own.

As an example the mass flow controller (MFC) sub-VI that handles communicating with MFCs would read from a configuration file (using the NI provided configuration file Vis) and set the flow values of the various MFCs on the first run. It also must handle monitoring for changes to update the same configuration file. This same functionality must then be duplicated across to any other Vis that need to have controls be initialized to their last modified value. While generally straight forward to duplicate functionality across multiple subVIs it can be error prone, one example is needing to use the same file path and name on each of the subVIs where typos will result in multiple files being created.

- **Created new subVI to handle initialization/monitoring all controls that require such functionality on a cRIO controller**
- **Tested with real shared variables on a development controller**
- **Will be deployed to production controllers during next extended maintenance window**

Handling Initialization of NI LabVIEW Controls

A new subVI was developed to provide a unified way of initializing controls on first run along with monitoring the controls for changes. In order to simplify the code it was decided to simply provide an array of shared variables, as PSP (NI Publish-Subscribe Protocol) references, and use their name as the key for the file configuration Vis and a fixed section would be used on all of them. Instead of at least 3 values needing to be passed for each variable monitored (shared variable, section, key) only the shared variable itself needs to be passed. The two downsides to this approach are that the key values are the same as the shared variable name as well as not separating the keys into different sections, however not having to pass these parameters in significantly decreased the code needed.

The subVI was tested on a subset of the shared variables on a development cRIO controller, which is a side benefit to using the PSP variables. It performed as expected and will be deployed to the production controllers once a longer maintenance window is available