Debugging RICH-2 Hardware Interlock Trip Delay

Tyler Lemon
2022-06

Debugging RICH-2 Hardware Interlock Trip Delay

I implemented and debugged a hardware interlock trip delay for the RICH-2 hardware interlock system's temperature sensors.

During the first week of the RICH-2 hardware interlock system running in Hall B, there were two occasions where I^2C communication errors between the sbRIO and SHT35 sensors saw temperature values jump to 130° C for up to three seconds. Despite logic being included in the hardware interlock system's LabVIEW program to prevent these false, communication-error induced temperature spikes from triggering an interlock, disabling RICH-2's CAEN power supply mainframe, these false readings still caused an interlock on two occasions.

After the second occurrence, it was determined that the trip delay feature for temperature sensors should be enabled. This trip delay feature was included in initial development of the LabVIEW program since it is a standard feature of all DSG-developed hardware interlock systems, but it was not enabled since it is not used for RICH-1.

The trip delay feature works by waiting to take the final action of interlocking the CAEN mainframe until the sensor reading is over-limit for a duration of time set by the user. Additionally, if the sensor reading drops down below the user-set limit at any point in the delay period, the delay timer is reset.

When first enabled, the delay time was set for 30 seconds, but unfortunately did not work as expected, causing one additional unwanted interlock trip due to communication errors.

- **I^2C communication errors cause false 130° C temperature readings in some SHT35 sensors, in turn trigging a false interlock**
- **30-second trip delay implemented for temperature interlocks to prevent false interlock trips**
- **Investigation into I^2C communication planned to see if changing communication signals' timing results in a better, more stable behavior**

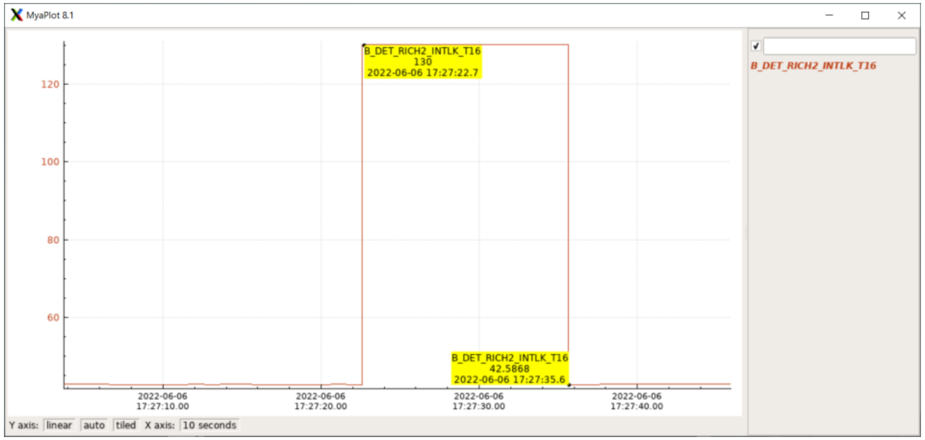
Debugging RICH-2 Hardware Interlock Trip Delay

After investigation, a flaw in the Boolean logic used to implement the trip delay was found. In the initial version, the logic was causing a “trip postponement” rather than the desired delay. With the flaw, if a temperature reading was over the set limit, the program would wait 30 seconds, and then trip the interlock, regardless of whether the sensor reading had returned to below the interlock limit. When this was behavior was found, the Boolean logic was able to be corrected to implement the desired trip delay.

Since implementing the fix, the RICH-II hardware interlock system has ran smoothly without any false interlock trips induced by I^{2}C communication error.

In the upcoming months, I intend to further debug the cause of the I^{2}C communication errors that result in the incorrect 130° C temperatures. This will be done by setting up a development test station using DSG’s spare sbRIO, RMC, backplane PCB, and SHT35 sensors.

During this investigation, the first task will be to see if the errors can be induced in any way, or if they occur periodically on their own. Next, I will modify the I^{2}C communication timing to test whether different clock frequency or different phases between clock and data signals results in a more stable communication without any errors. Once the optimal settings for I2C communication are found and the system is stable (in terms of communication errors), the changes will be pushed to the system that is operational in Hall B during the next maintenance period.



Y axis: linear auto tiled X axis: 10 seconds

Fig. 1: Archived EPICS data for RICH-2 temperature sensor 16 showing a 3-second period where it incorrectly gave a 130 C reading. This time period was before the trip delay was implemented.