

Application of Control System Studio for the NOvA Detector Control System.

Gennadiy Lukhanin¹, Kurt Biery¹, Stephen Foulkes¹, Martin Frank², Athanasios Hatzikoutelis³, Jim Kowalkowski¹, Marc Paterno¹,
Ron Rechenmacher¹

¹Fermi National Accelerator Laboratory, Batavia, IL

²University of Virginia, Charlottesville, VA

³University of Tennessee, Knoxville, TN

{ lukhanin, biery, sfoulkes, mfrank, athansh, jbk, paterno, ron }@fnal.gov

Abstract. In the NOvA experiment, the Detector Controls System (DCS) provides a method for controlling and monitoring important detector hardware and environmental parameters. It is essential for operating the detector and is required to have access to roughly 370,000 independent programmable channels via more than 11,600 physical devices. In this paper, we demonstrate an application of Control System Studio (CSS), developed by Oak Ridge National Laboratory, for the NOvA experiment. The application of CSS for the DCS of the NOvA experiment has been divided into three phases: (1) user requirements and concept prototype on a test-stand, (2) small scale deployment at the prototype Near Detector on the Surface, and (3) a larger scale deployment at the Far Detector. We also give an outline of the CSS integration with the NOvA online software and the alarm handling logic for the Front-End electronics.

1. Introduction

NOvA is the next generation neutrino oscillation experiment presently under construction at Fermilab. It will begin taking data in 2013 and construction will be complete in January 2014 [1]. The NOvA experiment consists of two detectors, the Near Detector and the Far Detector, each of which contains a large number of individual detector elements. The NOvA Detector Control System (DCS) monitors and controls the detector conditions and detector electronics. It also provides an operator with warnings if abnormal conditions occur, issues alarms, and executes control actions to protect the detector electronics from possible damage. The monitored conditions, data, and alarms are recorded and archived into the central NOvA database.

Control System Studio (CSS) [2,3] was chosen for the implementation of the APD (*avalanche photo diode*) monitoring and alarm display. CSS is a feature rich application development platform for implementing graphical operator interfaces, recording and archiving of monitored data, monitoring alarm conditions, and presenting them to an operator.

This paper describes the application of CSS for monitoring APDs for NOvA's Near and Far Detectors illustrated on the NOvA Detector prototype On the Surface (NDOS) case.

2. Background and operational requirements

The Near and Far Detectors consist of di-blocks which are groups of alternating horizontal and vertical planes of extruded PVC cells (rectangular pipes) that are filled with mineral oil [4]. Each cell is read out by a wavelength shifting fiber connected to one pixel of a 32-pixel APD. A Data Concentrator Module (DCM) is connected to 64 Front-End Boards (FEBs) that digitize signals coming from 64 APDs. The APDs are cooled by thermo-electric coolers (TECs).

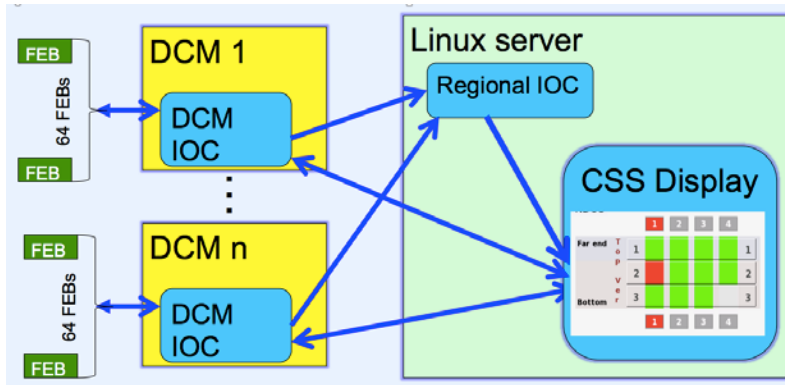


Figure 1: DCS Diagram.

Monitoring and controlling of APDs is carried out by the DCM IOCs. IOC stands for Input/ Output Controller that is implemented in the Experimental Physics and Industrial Control System (EPICS) framework [5]. A DCM IOC provides access to individual EPICS process variables, such as FEB and APD temperatures, TEC drive currents and FEB status information. The regional IOC aggregates all FEB and DCM status information, so it can be efficiently presented in CSS. All alarm logic is implemented in the DCM IOCs and the Regional IOC.

The APD monitoring and alarm display is required to present aggregated status and alarm information from 11,600 independent channels (corresponding to 370,000 Data Acquisition channels) in a uniform way regardless of controls interface. The NOvA project calls for a scalable and modular implementation of the DCS system that can be partitioned into production and calibration/commissioning variants. The DCS system has five distinct functions: 1) configuring detector parameters and providing their readbacks; 2) handling alarm conditions; 3) archiving conditions data; 4) providing operators with a visual representation of the detector status; and 5) executing control actions to protect the detector electronics from possible damage.

3. Implementation

The APD monitoring and alarm display are implemented in CSS as Operator Interface (OPI) pages presenting the detector status information as four different levels of the detector view: full detector, di-block, DCM, and FEB.

The first page is the main Near (Far) detector startup page (Figure 2), which is loaded first by a CSS startup script. It is used for:

- 1) starting and checking the state of APD monitoring;
- 2) running the APD cooling script and verifying that the entire detector has reached a stable cold state; and
- 3) changing the page to the detector overview page.

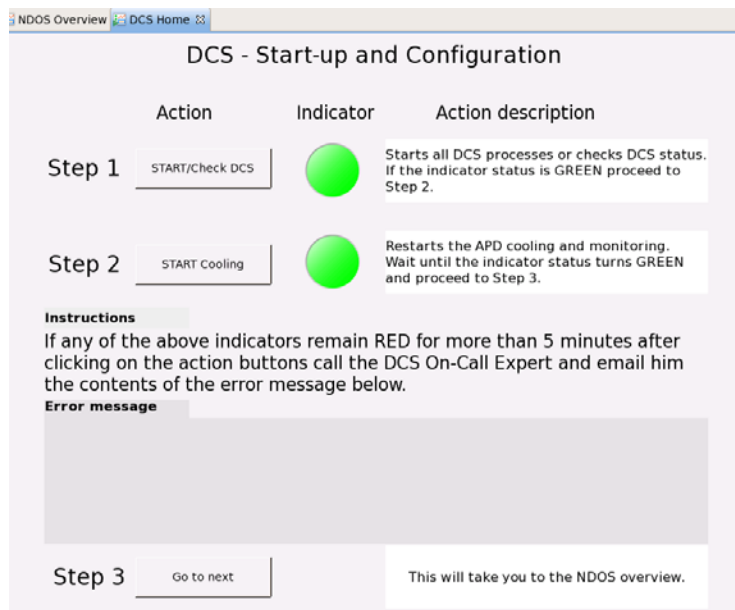


Figure 2: The DCS Start-up and Configuration page.

In the NOvA experiment, active cooling of the readout electronics is essential for producing quality data and thus it is an important function that the DCS performs. It is a precondition for taking data. Cooling is configured at the startup of detector operations and then monitored by the DCS for any inconsistencies from the initially set target parameters. The DCS provides a visual notification on the operator interface if such events occur and record these events into the conditions database. The procedure was developed at the NDOS, which is a prototype Near detector of NOvA but on the surface. It is built for gaining expertise with the NOvA detector technologies, and the development of the custom data acquisition and detector controls systems.

Operators enable cooling by clicking on the “Start Cooling” button that executes a sequence of EPICS channel-access commands, and configures warm and cold set-points that set the FEB cooling mask for each DCM in parallel. The “error message” text field will show errors returned by the cooling script, if any.

The detector overview page (see an example implementation for NDOS in Figure 3) provides an operator with a high-level overview of the entire detector status including individual statuses of DCMs and di-blocks. DCMs are grouped into columns by di-block.

Operator interfaces are implemented in the “Best OPI, Yet (BOY)” framework [6], which currently offers around 40 widgets that include a variety of shapes, indicators and buttons; grouping and layout containers; 1 and 2 dimensional graphs and heat maps; and linking containers. Most widgets are connected to EPICS process variables (PVs) and dynamically change their properties such as colors and displayed values, based on values of corresponding PVs. Macros and linking containers were used for combining several widgets into reusable OPI page fragments, which were included into higher-level OPI pages. Overlaying several text widgets over the intensity graph allowed a compact presentation of 64 FEBs on the DCM and FEB overview pages.

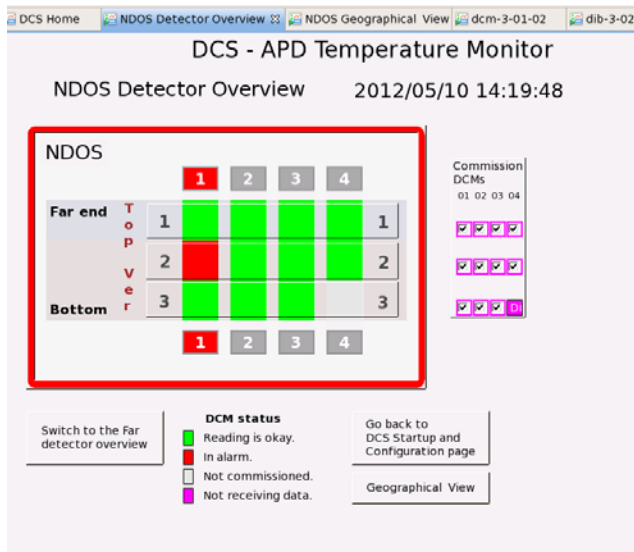


Figure 3: “NDOS Detector Overview” page.

If any of the DCMs are in alarm, the outer border, which represents the entire detector, turns red. In addition, the di-block number boxes at the top and bottom of a column turn red when a DCM in that column has an alarm. As an example, Figure 3 shows the second DCM within the first di-block in an alarm state.

The di-block overview page (Figure 4 implemented for the NDOS as well) goes one level deeper by displaying status information for each DCM belonging to that di-block and each FEB belonging to these DCMs. This allows operators to see how many FEBs are in alarm and possibly derive further conclusions. Typically, operators would look for the red and pink rectangles representing alarms and connectivity issues, respectively, interpreting the detailed legend in Figure 4. Operators can navigate to the DCM level view by clicking on one of the three rectangular areas representing individual DCMs.

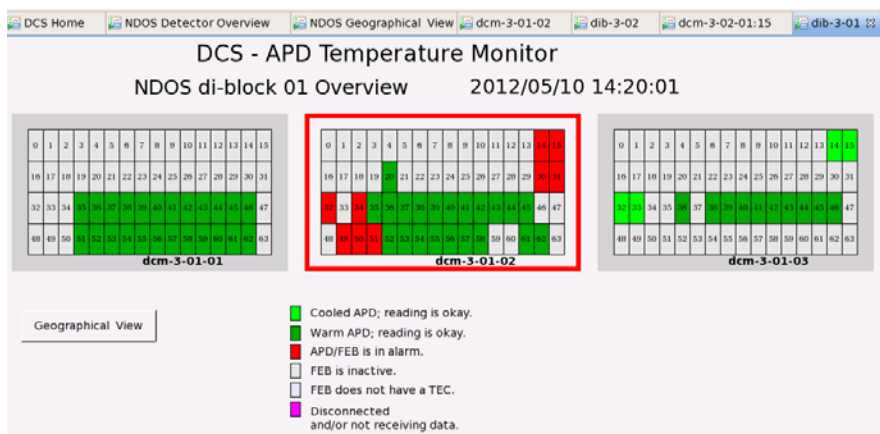


Figure 4: “NDOS Di-block Overview” page. The DCMs of this di-block as well as all the FEBs of each of these DCMs are shown numerically labeled.

In order to investigate problems even further, operators can navigate to the DCM level page (Figure 5) by clicking on the rectangle areas representing individual DCMs. The page at this level shows readings from each of the FEB boards connected to this DCM. Each FEB has one APD connected to it. The monitored values are the APD temperature, TEC drive current and FEB temperature for each of the FEBs overlaid on top of the FEB status.

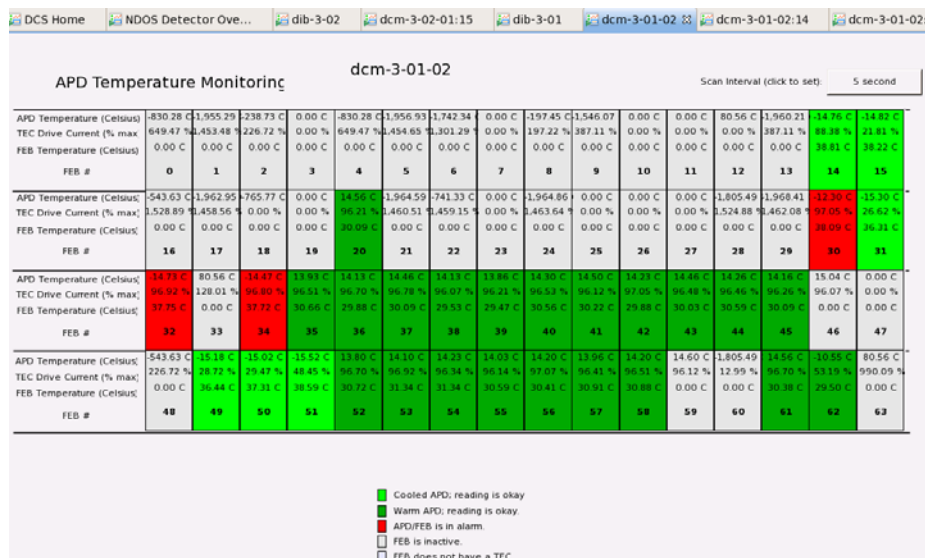


Figure 5: DCM Overview page for one of the DCMs on the NDOS.

Operators can view FEB level pages (Figure 6) by clicking on the rectangles representing FEBs (in Figure 5), the FEB level page shows the time series of all monitored FEB parameters, which is valuable when doing low level troubleshooting of FEBs.

Logging and archiving of the conditions data are essential functions of the DCS. The conditions data is written into the online database every 15 and 60 seconds. The 15-second resolution data is kept in the database for two weeks and operators can view it using the data browser plugin in CSS. The 60-second resolution data is permanent and is used for the offline analysis.



Figure 6: FEB level page for one of the DCM in one of the di-blocks of the NDOS.

The left three plots in Figure 6 are produced by reading a waveform record containing the last 24 data-samples from the device cache. The three plots on the right are showing trends from the last hour and are retrieved from a local MySQL database. Operators can interact with these plots by adjusting their time window and scale. This functionality is activated by enabling a toolbar on that plot.

Figure 7 illustrates an extended view of the Detector Overview page. It shows the alarm tree on the left and the alarm table at the bottom. Alarm sources need to be configured with the Alarms server before they can be displayed in the GUI. The alarm configuration is stored in the database. Operators can acknowledge, un-acknowledge and configure alarms. They can also access guidance information

from both the alarm tree and the alarm table. The alarm subsystem is not yet fully integrated into the DCS system at the NDOS and development work is ongoing.

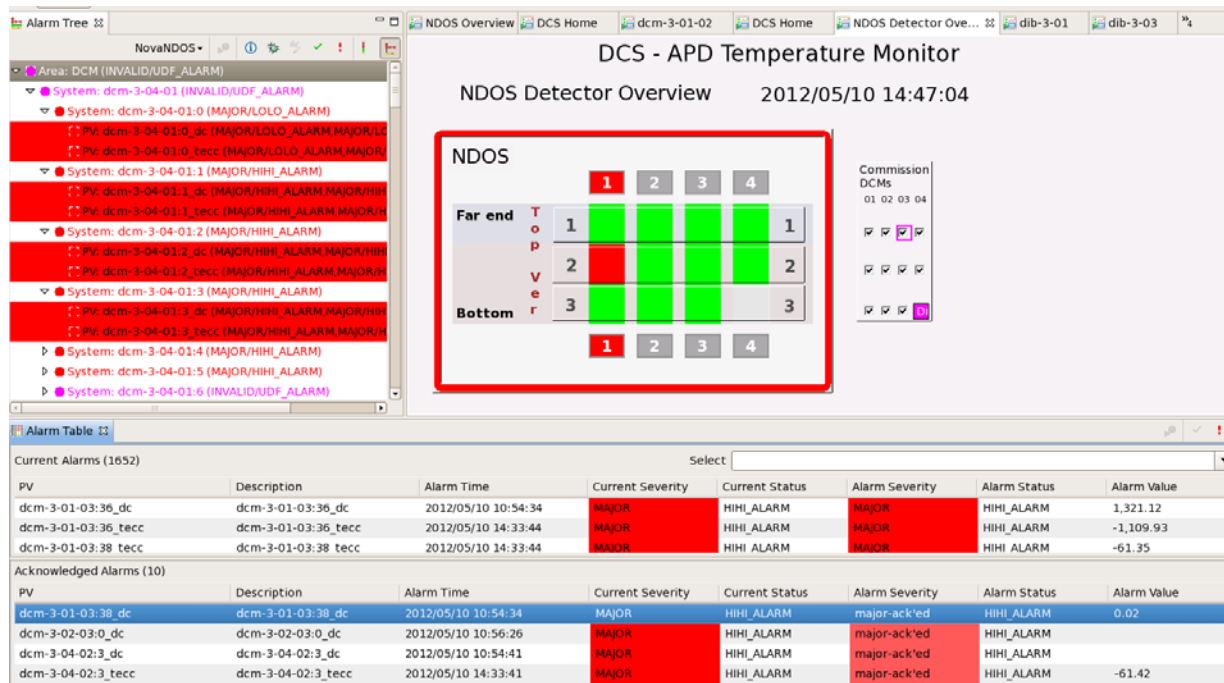


Figure 7: An extended view of the Detector Overview page.

4. Conclusions

We have successfully deployed CSS, which provides a feature-rich environment for implementing detector control systems. The initial prototype and improved versions of the DCS system are currently running on the NDOS. The gained experience with CSS has allowed us to improve the performance and achieve a higher scalability that is required for the Far detector.

The Fermi National Accelerator Laboratory is operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy. This contribution is FERMILAB-CONF-12-292-CD.

References (need to update references)

- [1] "NOvA." *Fermilab*. N.p., n.d. Web. 17 June 2012. <<http://www-nova.fnal.gov/index.html>>.
- [2] J. David Purcell et al. "CSS - We didn't Invent It, We Made It Better." Proc. of ICALEPCS2009, Kobe, Japan.
- [3] K.-U. Kasemir et al. "The Best Ever Alarm System Toolkit." Proc. of ICALEPCS2009, Kobe, Japan
- [4] "Technical Design Report." *NOvA*. N.p., n.d. Web. 17 June 2012. <http://www-nova.fnal.gov/nova_cd2_review/tdr_oct_23/tdr.htm>.
- [5] "Experimental Physics and Industrial Control System." *EPICS* -. N.p., n.d. Web. 17 June 2012. <<http://www.aps.anl.gov/epics/>>.
- [6] Chen, X.H., and K.-U. Kasemir. "BOY, A Modern Graphical Operator Interface Editor and Runtime." Proc. of 2011 Particle Accelerator Conference, New York, NY.