



Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie
AGH University of Science
and Technology

Update and upgrade of the detector control system for ATLAS TRT GGSS

Grzegorz Podsiadło

Faculty of Physics and Applied Computer Science
Reymonta 19
30-059 Cracow

Supervisor: dr hab. inż. Bartosz Mindur

September 22, 2020

- 1 Short introduction to GGSS, DCS and role of GGSS Reader
- 2 Update and upgrade of the detector control system
- 3 GGSS Reader software update
- 4 Summary and plans

About GGSS, DCS and role of GGSS Reader

- GGSS¹ is a part of the Detector Control System for ATLAS TRT².
- GGSS uses the DIM³ interface for data exchange with SCADA⁴ project, based on Siemens WinCC Open Architecture software.
- GGSS Reader is a newly created application that allows you to pretend to be a GGSS server using DIM for mocking example data into SCADA project (old data from previous GGSS runs can be used).

¹Gas Gain Stabilizations System

²Transition Radiation Tracker

³Distributed Information Management System

⁴Supervisory Control And Data Acquisition

WinCC OA Migration

- CERN is currently updating the WINCC OA software version used in the Detector Control system from version 3.15 to 3.16. Migration must be completed before the start of the next run.
- A new `pcat1trttest4` machine was prepared, on which software is being migrated and new changes are tested.
- Version 3.15 is still available on `pcat1trtdev0` machine, but no major changes have been made there except for uploading a new license.

Panels format change in 3.16

- Version 3.16 will use new panel formats - it was necessary to switch to xml from pnl format.
- I have formatted the existing panels using the available tools.
- After migrating, changing panel format and making minor coding corrections, I performed manual tests of the project on a new machine. Everything seems to be working fine in the test environment, but additional tests (including production tests) will be needed.

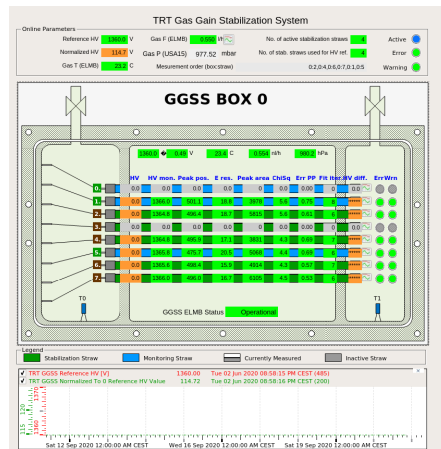


Figure: Main GGS panel on test environment with version 3.16 installed.

Project upgrade

- I am currently working on updating the project, which will remove unused DPEs and code used in panels.
- Additionally, a part of the interface will be refreshed with the use of new solutions.
- A list of unused code has been prepared, but we will still need to do additional testing before removing it.
- There are still many functionalities which need to be improved or fixed, such as running the GGSS processes from the WinCC OA panels level.

Project upgrade with WebView

The new version of the WinCC OA software has added a lot of features, one of the most interesting is the introduction of JavaScript support within the WebView control. It allows rendering of panel views written in HTML with the use of exactly the same API interface that was used so far in the CONTROL language. This allows the use of existing JS libraries, which significantly increases the possibilities during the ongoing upgrade.

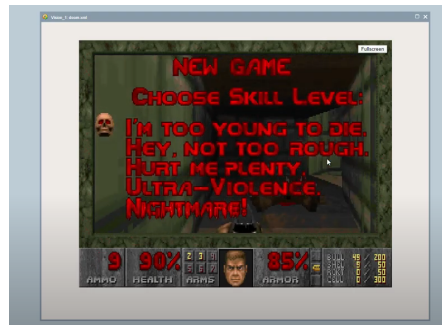


Figure: Funny example of a playable version of Doom implemented as WinCC OA panel using JS.

Project upgrade with WebView

The new controls allow the use of any JS library, for example, Plotly⁵. Communication with the control system takes place using the shared API⁶. The source code with HTML and js files will be placed in a special `data/html` directory. The new solution requires the configuration of one additional control manager to work, which was installed on the test machine.

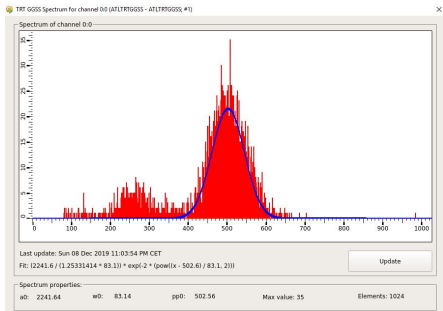
St	Description	No Options	Start time	PID	Mode	Restart	ResetMin	Sec
0	Process Monitor	1	2020.09.13 22:57:46	793	manual	3	1	30
0	Database Manager	0	2020.09.13 22:57:46	797	always	3	1	30
0	Event Manager	0	2020.09.13 22:58:21	890	always	3	1	30
0	Control Manager	1 -f prjs_scripts.lst	2020.09.13 22:58:21	893	always	3	1	30
1	Simulation Driver	1	2020.09.13 22:58:22	895	always	3	1	30
0	Distribution Manager	1	2020.09.13 22:58:22	899	always	3	1	30
0	PCDI Archive Manager	99 -num 99			-1 manual	2	2	30
0	Simulation Driver	7 -num 7	2020.09.13 22:58:22	901	always	2	2	30
0	OPC UA Client	9 -num 9			-1 manual	2	2	30
0	Simulation Driver	9 -num 9	2020.09.13 22:58:22	905	always	2	2	30
0	WCCDataIn	2 -num 2 -dm_osteqi Abetika\Production\ATLAS_DCS_TRT\ATL			-1 manual	2	2	30
0	Control Manager	2 unControlManagerControl.cst	2020.09.13 22:58:22	907	always	2	2	30
0	Control Manager	3 -f prScripts.lst	2020.09.13 22:58:23		-1 once	1	1	30
0	Control Manager	4 fccDMHFunction.cst	2020.09.13 22:58:23	919	always	3	1	30
0	Control Manager	50 never touch me -num 50			-1 manual	3	5	30
0	WCCDataIn	10 never touch me -num 10	2020.09.13 22:58:23	906	manual	3	5	30
0	Control Manager	70 fccEhpPhEhbCheckinval.cst -num 70			-1 manual	2	1	30
0	Control Manager	5 setDevelopmentEnv.cst	2020.09.13 22:58:23		-1 once	2	2	30
0	User Interface	-in god			-1 manual	3	1	30
0	Control Manager	1 -f fccInstallationAgent.lst			-1 manual	2	2	30
0	Control Manager	7 stGGSS\transfer_atrib_data.cst -num 7			-1 manual	2	2	30
0	Control Manager	12 stGGSS\st_start_handling.cst -num 12			-1 manual	2	2	30
0	Control Manager	13 stGGSS\stgs_email_sender.cst -num 13			-1 manual	2	2	30
0	Control Manager	1 -f fccAttestPM.lst			-1 manual	3	5	30
0	Control Manager	3 fccAttestPM.lst	2020.09.20 19:45:07	6312	manual	2	2	30

Figure: Managers configuration for GGSS project with new WebClient manager.

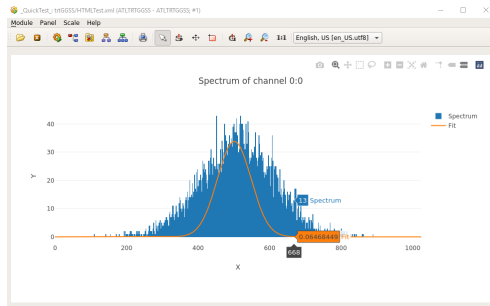
⁵See Plotly JavaScript Open Source Source Graphing Library

⁶You can check oaJsApi documentation here.

Testing new functionalities



(a) v. 3.15: Bar Trend + Control



(b) v. 3.16: WebView EWO + HTML and JavaScript, (*work in progress*)

Figure: WinCC OA panels responsible for drawing channel spectrum.

GGSS Reader Update

During the summer school, I updated the application building scripts using CMake and also improved the dependencies on projects developed by other students. Some minor improvements to file handling and additional unit tests have also been added to the project. Currently, the most optimal version of the program, which passes all tests, is on the master branch.

I am at the end of work on a new modified application console interface, which will include more data replacement options. The interface is written using boost signal library and is multi-threaded. Description of the new interface and how to use it will be included in the readme.md (and doxygen docs) of the project.

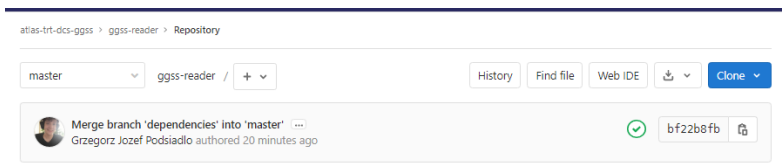


Figure: The latest tested version of the application located on the master branch.

GGSS Reader Interface update and GUI application

Additionally, the GUI application written in QT is under development, which will be a tool accelerating the work with the application. It will use the new interface and will be an optional tool. So it will be possible to use both the console version and the GUI version, which is an overlay for the CLI that improves the work with files.

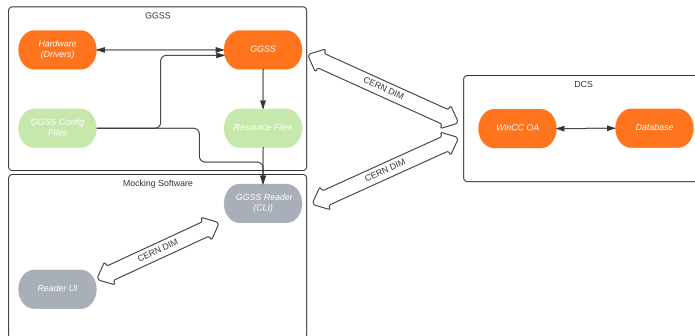


Figure: Simplified architecture of mentioned systems.

Summary and plans

- The GGSS project was successfully migrated to a test machine with the conversion of panels.
- A list of needed changes to the SCADA project and a list of unused DPEs were prepared. I tested new types of controls that allow the use of the JS interface based on the example of a spectrum control. I am currently working on updating the panels.
- The GGSS Reader project has been updated to match the new dependencies, the second version of the interface will be published soon, along with the QT application.
- There are still many functionalities which need to be improved or fixed, such as running the GGSS processes from the WinCC OA panels level.

Thank you for participating. Any Questions?