PROJECT REPORT

Automatized analysis of the DCS data of the ALICE Inner Tracking System at the LHC

EP-AID-DT

Viktoriia Starynchuk
Lviv Polytechnic National University

Supervisors: Ivan Ravasenga, Andrea Sofia Triolo

CERN Summer Student Program
Geneva, Switzerland
September 2025

ABSTRACT

This project presents a web-based interactive tool for the automatized analysis of Detector Control System (DCS) data from the ALICE Inner Tracking System at the LHC. The tool replaces manual workflows with a user-friendly interface that enables automatic database queries, efficient data processing, and dynamic visualization of key parameters such as temperature, voltage, and current. Built in Python with Dash, pandas, and Plotly, the system streamlines access to large volumes of DCS data, allowing users to quickly identify anomalies and extract meaningful insights through time series trends and histograms.

Key words: ALICE, ITS2, DCS, GUI, HIC.

ACKNOWLEDGMENT

I would like to sincerely thank my supervisors for their continuous guidance, support, and encouragement throughout this project. Their expertise and advice were invaluable in helping me overcome challenges and develop my skills. I am also deeply grateful for the opportunity to participate in the CERN Summer Student Programme, which allowed me to gain unique research experience, collaborate in an international environment, and contribute to the work of the ALICE experiment. This summer has been an inspiring and unforgettable experience in my academic and personal journey.

CONTENTS

1. INTRODUCTION. THE ALICE INNER TRACKING SYSTEM

1.1 ALICE - A Large Ion Collider Experiment

ALICE (A Large Ion Collider Experiment) is one of the experiments at the Large Hadron Collider (LHC) at CERN. It is designed to study the physics of strongly interacting matter at extreme energy densities, particularly the formation and properties of the Quark-Gluon Plasma (QGP). The QGP is a state of deconfined quarks and gluons that existed in the early universe a few microseconds after the Big Bang [1]. Such conditions can be recreated in high-energy heavy-ion collisions, such as lead-lead (Pb-Pb) collisions, where a fireball of QGP is formed, rapidly expands, thermalizes, and cools. The study of these collisions allows physicists to explore fundamental properties of Quantum Chromodynamics (QCD), the theory of strong interactions.

1.2 ALICE Detector Layout

The ALICE detector, approximately 16 × 26 × 16 m³ in size, has been optimized to handle high particle-multiplicity events expected in Pb-Pb collisions. A sketch of the ALICE apparatus is visible in **Figure 1**. It can detect particles down to very low transverse momentum ($p_T \approx 0.1$ GeV/c) and provides excellent particle identification (PID) capabilities up to 20 GeV/c.

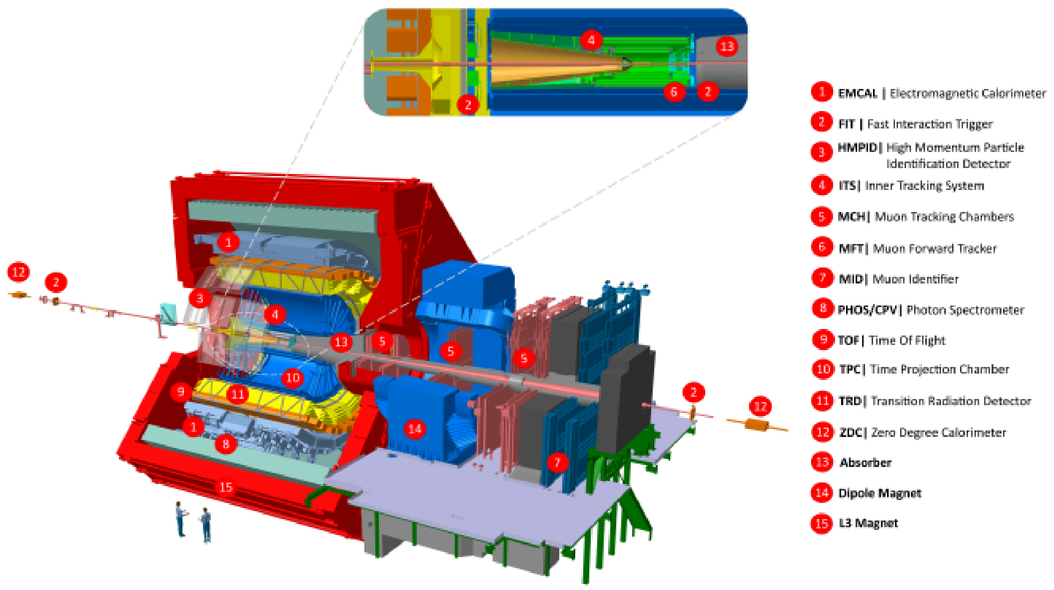


Figure 1: ALICE2 detector systems.

The detector consists of a **central barrel** and a **forward region**, each composed of multiple subsystems:

- **Central barrel detectors**:
  - **Inner Tracking System (ITS):** precise tracking and vertexing, located closest to the beam pipe.
  - **Time Projection Chamber (TPC):** main charged particle tracker with PID via specific energy loss.
  - **Time Of Flight (TOF):** measures particle flight time for PID at intermediate momenta.
  - **Transition Radiation Detector (TRD):** electron identification and tracking.
  - **Photon Spectrometer (PHOS), Electromagnetic Calorimeter (EMCal), High Momentum Particle Identification Detector (HMPID):** specialized measurements for photons and hadrons.

These detectors are enclosed by the L3 solenoid magnet (0.5 T). The ITS, TPC, TRD, and TOF provide full azimuthal coverage with acceptance up to $|\eta| < 0.9$.

- **Forward detectors**:
  - **T0 and V0 detectors:** provide timing and trigger information.
  - **Zero Degree Calorimeter (ZDC):** measures spectator nucleons for collision centrality.
  - **MUON spectrometer:** reconstructs heavy-quark resonances via muons, including a hadron absorber, dipole magnet, and tracking/trigger chambers.

2. INNER TRACKING SYSTEM (ITS2)

The ITS2 is a key detector of ALICE, responsible for tracking particles near the interaction point with high precision, reconstructing particle trajectories, measuring momentum, and identifying short-lived particles [2]. The upgraded **ITS2** consists of seven layers of silicon Monolithic Active Pixel Sensors (MAPS) arranged in two barrels:

- **Inner Barrel (IB):** layers 0–2, 48 staves.
- **Outer Barrel (OB):** layers 3–6, 144 staves.

The detector hosts 24,120 ALPIDE chips (TowerJazz, CMOS 180 nm imaging process) with a pixel pitch of 26.88×29.24 um^2. Each chip contains 512 × 1024 pixels, giving a total of 12.6 billion pixels in the full detector: the largest ever built in a high-energy physics experiment (Fig. 2).

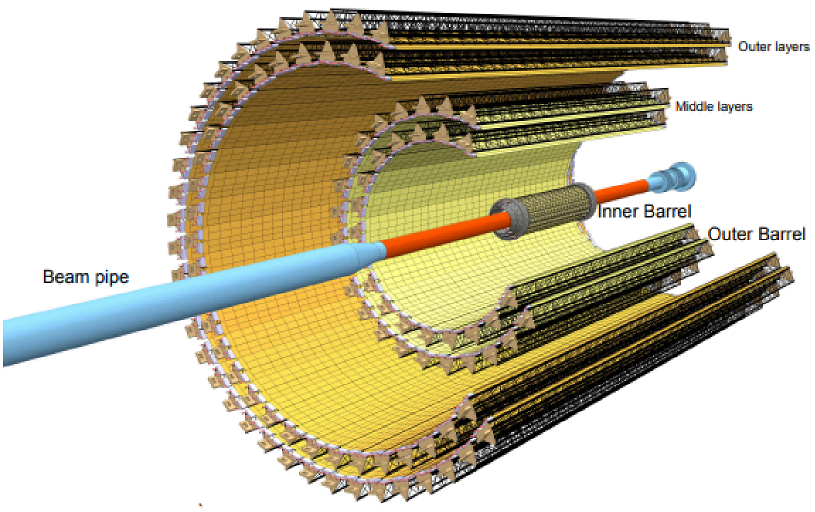


Figure 2: Schematic layout of the ITS2.

The basic detector unit, called stave, consists of the following elements (Fig. 3):

- Hybrid Integrated Circuit (HIC): an assembly of a polyimide Flexible Printed Circuit (FPC) on which a number of pixel chips, namely 9 and 14 for the inner and outer barrel staves, respectively, and some passive components, are bonded.

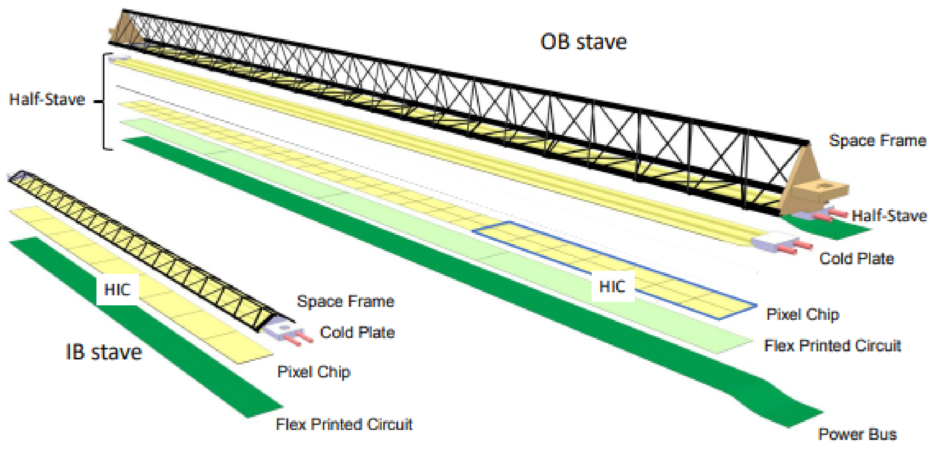


Figure 3: Layout of the staves of the inner and outer barrels.

- Coldplate: a carbon fibre sheet with high thermal conductivity with embedded polyimide cooling pipes, which is either integrated within the space frame (for the inner barrel staves) or attached to the space frame (for the outer barrel staves).

- Space Frame: a carbon fiber truss-like support structure providing the mechanical support and the necessary stiffness to the assembly of HICs on cold plates. .

The HICs are glued onto the cold plate: 1 HIC for the inner barrel and 8 and 14 HICs, for the middle and outer layer staves, respectively. The cold plate is in thermal contact with the pixel chips to remove the generated heat. For the inner barrel, each stave consists of a single HIC+cold plate assembly. In the outer barrel, staves are further segmented in azimuth in two identical half-staves. Each half-stave extends over the full length of the stave and consists of a cold plate on which four or seven modules (HICs) are glued depending on the length of the stave.

Moreover, the ITS2 achieves a low material budget of 0.36% X_0/layer in the innermost layers, which significantly enhances the pointing resolution and standalone tracking efficiency down to very low transverse momenta compared to the previous tracker [3].

3. DETECTOR CONTROL SYSTEM (DCS)

The operation and monitoring of the ITS2 are managed by the Detector Control System (DCS). The DCS continuously records critical parameters such as temperature, voltage, current, and calibration constants, which are essential for ensuring the proper functioning, stability, and safety of the detector. This information is stored in dedicated databases and accessed through DCS control panels, enabling engineers and physicists to monitor detector performance in real time. The DCS provides remote control and monitoring of all systems using a combination of industrial and custom devices, such as low- and high-voltage power supplies, PLC computers, data acquisition boards, and front-end electronics. These devices are hierarchically interconnected within a distributed control system, with most communication carried out via Ethernet using industrial standards such as OPC or the DIM protocol, which was developed at CERN. Additional devices rely on standard buses including CANbus, Profibus, or RS232. [4].

Supervision of individual devices is handled by the WinCC OA SCADA/HMI software from Siemens, integrated with the JCOP framework. Within ALICE, each detector has its own independent distributed control system, typically consisting of two to sixteen WinCC OA systems. These local systems are integrated into a central DCS, forming one complex, interconnected architecture. The control hierarchy is organized as a multi-layer tree structure of finite state machines (FSMs), which define the states and transitions governing both the detector as a whole and each of its subsystems. The main WinCC OA user interface of the ITS2 DCS is shown in **Figure 4**.

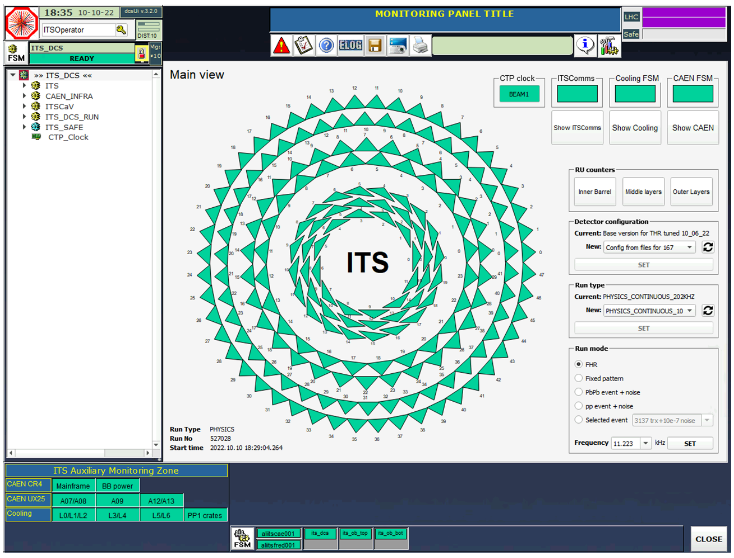


Figure 4: Detector Control System

To facilitate the storage, retrieval, and analysis of historical DCS data, the DCS Archive Manager (DARMA) is employed. DARMA is an information system designed to collect, organize, and provide access to the large datasets produced by DCS experiments. It enables users to download historical data efficiently and makes it available for further processing and analysis (Fig. 5).

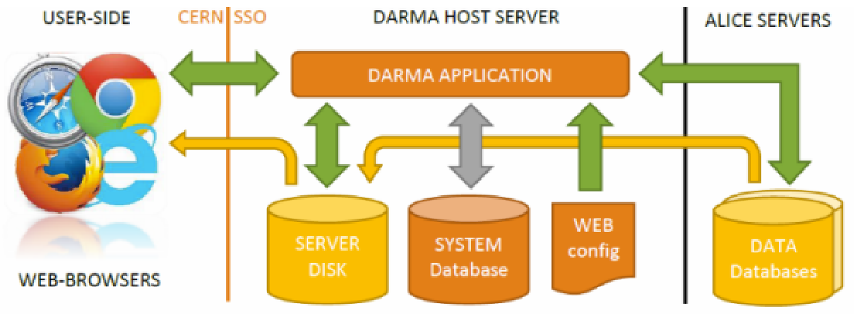


Figure 5: Schema of DARMA architecture.

DARMA provides access to data through both a web interface and a set of RESTful APIs, making it flexible for integration with software tools. The system supports structured queries to extract specific parameters or time intervals while preserving metadata such as timestamps, measurement units, and detector identifiers. [5].

By interfacing with DARMA, analysis tools can automatically extract the relevant data, clean and structure it, and generate visualizations, significantly reducing manual effort and enabling faster insights into the detector's status.

4. SOFTWARE DEVELOPMENT

4.1. PROJECT MOTIVATION AND OBJECTIVES

The main challenge is the massive amount of DCS data that needs to be analyzed.

The motivation of this project is to develop an automatized, interactive analysis tool that simplifies the retrieval, processing, and visualization of data. The system is designed to replace manual workflows with a user-friendly interface, making data analysis faster, more reliable, and easier to manage.

The objectives of the project include:

1. **Interactive Data Selection:** Allow users to select detector layers, staves, data points, and aliases, as well as define time ranges and measurement units such as temperature, voltage, current, or DAC values.

2. **Automated Data Retrieval:** Enable automatic construction of database queries to the DARMA system, reducing the need for manual scripting.

3. **Visualization and Analysis:** Generate dynamic plots, histograms, and time series trends, allowing users to quickly spot anomalies, errors, or unusual values.

4. **Predefined Query Templates:** Provide templates for common queries, including:
    - Readout Unit (RU)–related parameters
    - Hybrid Integrated Circuit (HIC)–related parameters
    - Power Unit (PU)–related parameters
    - Custom aliases

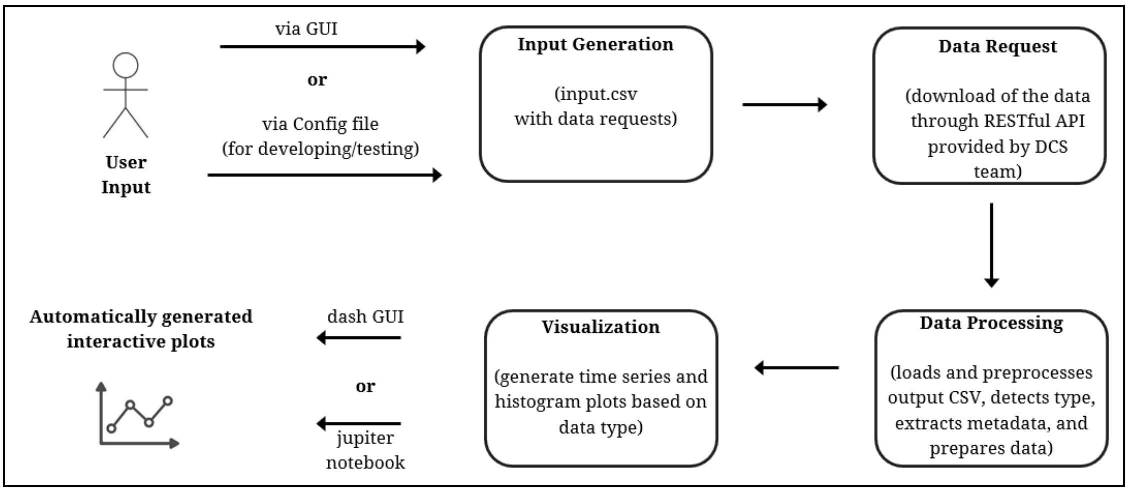
The workflow of the system is designed to be simple and efficient is shown on Figure 6:

- The user interacts with the web interface, selecting detector layers, staves, data points, aliases, time ranges, and measurement units.

- The selections are transformed into an input file and sent to the DARMA database through RESTful APIs.

- DARMA retrieves the requested data and returns it as an output file.

- The retrieved data is then processed, cleaned, and visualized in interactive plots.



- Figure 6: Dataflow of the system.

Overall, this workflow allows users to generate plots and analyze data in a matter of seconds, significantly speeding up the process compared to manual methods.

By achieving these objectives, the tool supports physicists and engineers in efficiently monitoring the ITS2, improving data analysis workflows, and facilitating the rapid identification of potential issues.

4.2. METHODOLOGY / SOFTWARE DEVELOPMENT

The project was implemented entirely in Python, leveraging modern libraries and frameworks to develop an interactive, web-based tool for analyzing ITS2 DCS data. The software architecture is modular, allowing for clear separation of responsibilities, maintainability, and scalability.

The graphical user interface (GUI) was developed using Dash and Dash Bootstrap Components (Fig.7), providing a responsive web-based interface that runs locally (http://127.0.0.1:8050/). The interface enables users to select detector layers, staves, datapoints, and aliases, define time ranges, and choose measurement units. Based on these selections, plots are dynamically updated, providing immediate visual feedback and facilitating interactive data analysis.

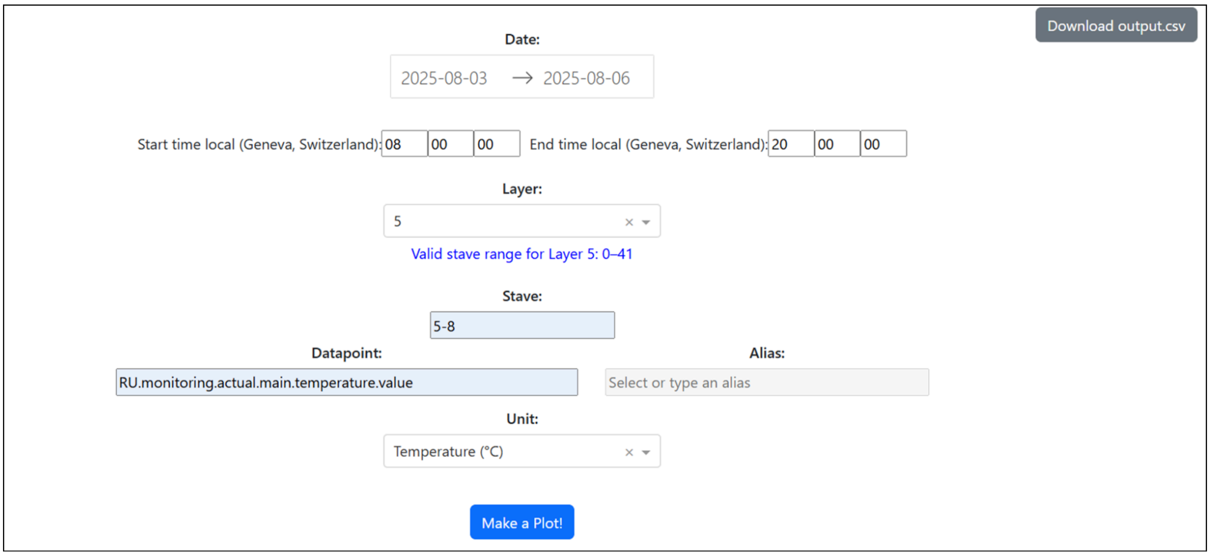


Figure 7: Dash web-based graphical user interface (GUI).

The Input Generator Module converts the user-friendly selections into a format compatible with the DARMA database. This involves mapping aliases and datapoint names to their full database element strings using regular expressions and predefined mapping logic, as well as validating the stave and layer selections. The module then generates an input CSV file (Fig.8), which serves as the query request for the backend system.

```
input.csv
timefrom;03.08.2025;10:00:00.000000;
timeto;05.08.2025;20:00:00.000000;
schema;ITSARCH;
element;its_ob_ti:ITS/L4_05/RU.monitoring.actual.SCA.V_IN
element;its_ob_ti:ITS/L4_06/RU.monitoring.actual.SCA.V_IN
```

Figure 8: Example of INPUT.CSV.

Communication with the DARMA database is handled by the Data Client Module, which uses the RESTful APIs provided by the DCS team. The module encodes the input CSV and sends it to a Flask backend via HTTP POST requests. Upon submission, client IDs are received to track the query, and the resulting data files are fetched and saved as output CSV files (Fig.9). This process ensures seamless and automated retrieval of historical DCS data, removing the need for manual database queries.

```
utput.csv
date;time;element_name/alias;value
2025-08-04;07:57:31;its_ob_ti:ITS/L4_05/RU.monitoring.actual.SCA.V_IN;6.856410026550293E+
2025-08-04;08:02:33;its_ob_ti:ITS/L4_05/RU.monitoring.actual.SCA.V_IN;6.8512821197509766E
2025-08-04;08:07:36;its_ob_ti:ITS/L4_05/RU.monitoring.actual.SCA.V_IN;6.856410026550293E+
2025-08-04;08:12:38;its_ob_ti:ITS/L4_05/RU.monitoring.actual.SCA.V_IN;6.846153736114502E+
2025-08-04;08:17:41;its_ob_ti:ITS/L4_05/RU.monitoring.actual.SCA.V_IN;6.846153736114502E+
```

Figure 9: Example of OUTPUT.CSV.

Once the data is retrieved, the Data Processing and Plotting Module preprocesses it, handling metadata, time zones, and measurement units. Pandas is employed for data wrangling, cleaning, and structuring, while Plotly is used for generating interactive visualizations. Seaborn is employed to manage color palettes, ensuring that multiple signals or aliases can be clearly distinguished within the plots. This module automates the full workflow from query submission to plot generation, producing time series trends, histograms, and other visual representations of key parameters.

During development, YAML configuration files were used to define default parameters for queries and visualization settings, while Jupyter notebooks facilitated rapid prototyping of plots and analysis workflows prior to integration into the GUI.

Overall, the combination of Python, Dash, Plotly, Seaborn, Pandas, and RESTful APIs results in a robust, user-friendly, and interactive system capable of efficiently analyzing and visualizing large volumes of ITS DCS data. The modular design ensures flexibility for future enhancements and provides a reliable platform for supporting physicists and engineers in monitoring the ITS2.

5. RESULTS

The developed tool successfully demonstrates the ability to retrieve, process, and visualize ITS2 DCS data in an automated and interactive manner. The system produces several types of plots that allow users to explore detector performance and identify potential anomalies.

Figure 10 shows an example of **time series trends per stave**, where parameters such as temperature, voltage, or current can be tracked over a selected time interval. This view is particularly useful for observing the stability of a stave and detecting sudden fluctuations or drifts in the measurements.

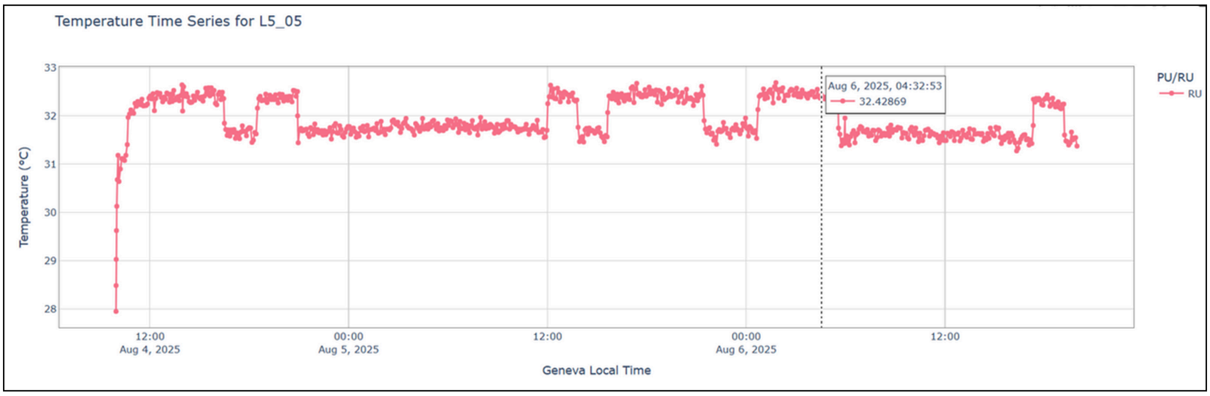


Figure 10: Example of time series trends per stave.

Figure 11 presents a **histogram per stave**, which provides a statistical overview of the distribution of parameter values for an individual stave. Such plots make it possible to quickly evaluate whether the behavior of a single stave deviates from expectations.

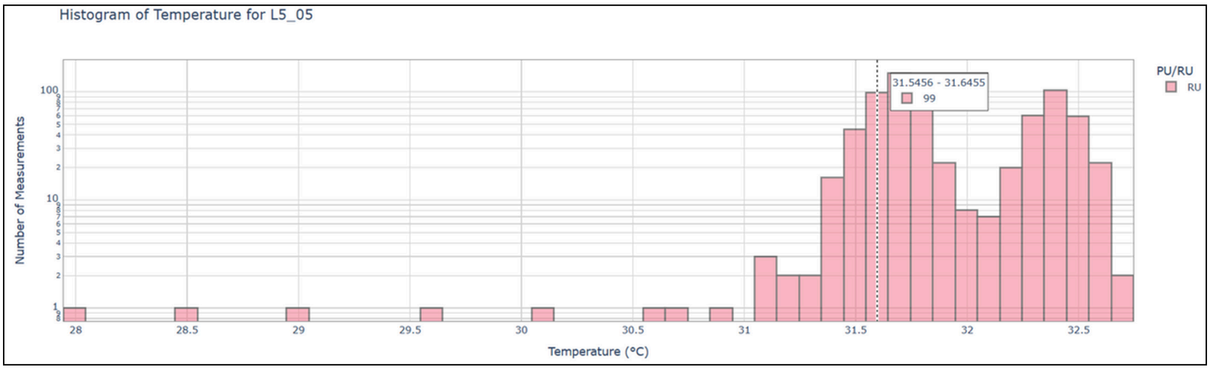


Figure 11: Example of histogram per stave.

Finally, Figure 12 illustrates a **combined histogram per layer**, where data from all staves within a given layer are aggregated. This representation highlights overall trends in a detector layer and helps identify whether anomalies are isolated to a single stave or common across multiple staves.

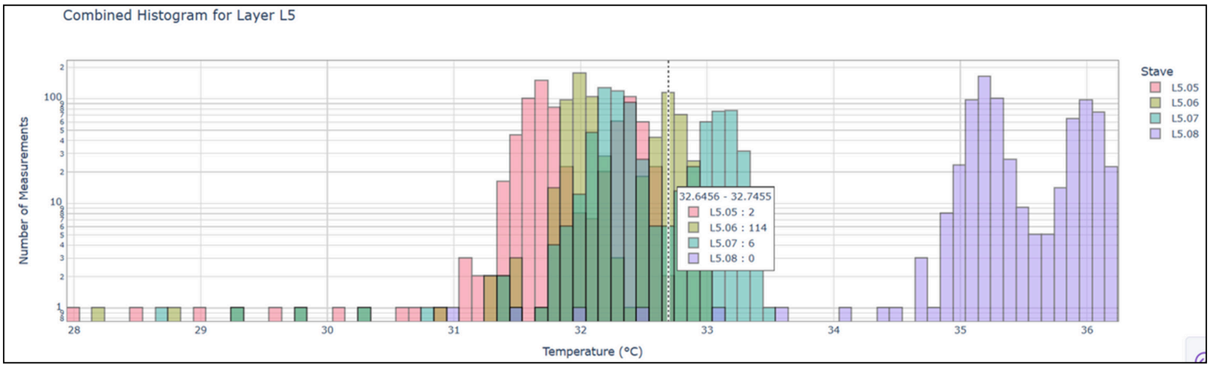


Figure 12: Example of combined staves histogram per layer.

The system has been tested with various types of DCS parameters, including voltages, currents, temperatures, and calibration constants. In each case, the tool automatically generated the corresponding plots, enabling efficient comparison between staves and layers. By visualizing these quantities interactively, users can easily detect abnormal behavior, such as sudden drops, spikes, or unusual value ranges.

To validate the functionality, the tool was applied to a broad set of datapoints. Examples of parameters tested include:

- Current: monitoring.actual.current.AVDD
- Voltage : monitoring.actual.voltageDiff.HIC.AVDD, monitoring.actual.voltage.output.DVDD, monitoring.actual.voltageDiff.PU.AVDD, RU.monitoring.actual.SCA.V_IN
- DAC Unit: monitoring.actual.trimmerOffset.DVDD
- Temperature: monitoring.actual.temperature.value (tested for PU0 and PU1)
- Other: monitoring.actual.bias.voltage, monitoring.stat.current.AVDD

In addition to raw datapoints, the system also supports the use of aliases for simplified referencing. For instance, ITS_L2_02_RU_IMON was among the aliases tested to ensure proper integration and readability within the plotting interface.

Overall, the results confirm that the developed tool provides a robust and user-friendly solution for the analysis of ITS2 DCS data, significantly reducing the time required to produce plots and improving the reliability of anomaly detection.

APPENDIX

Some more examples of automatically generated plots:

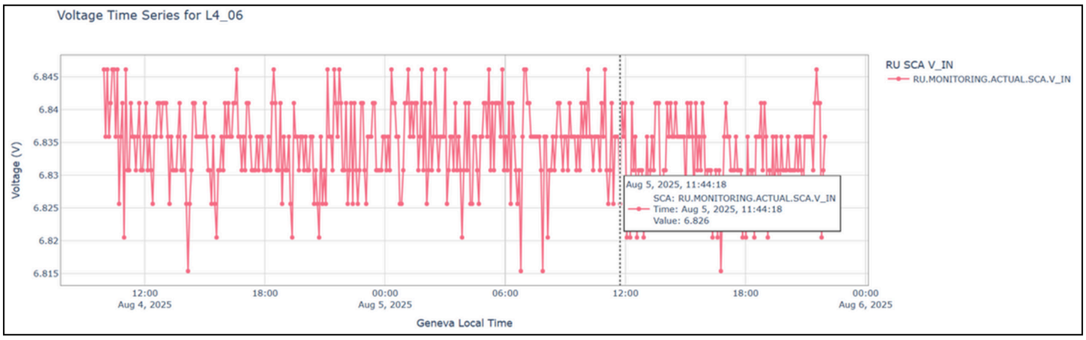


Figure 1: Voltage time series trend for layer 4, stave 6

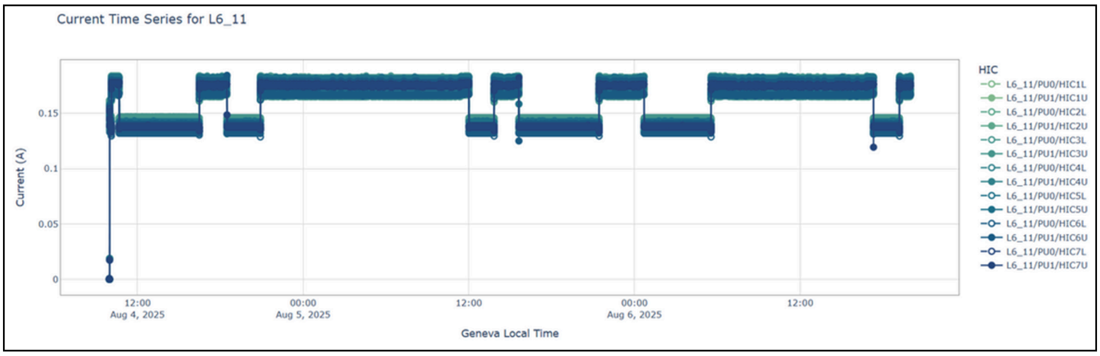


Figure 2: Current time series trend for layer 6, stave 11

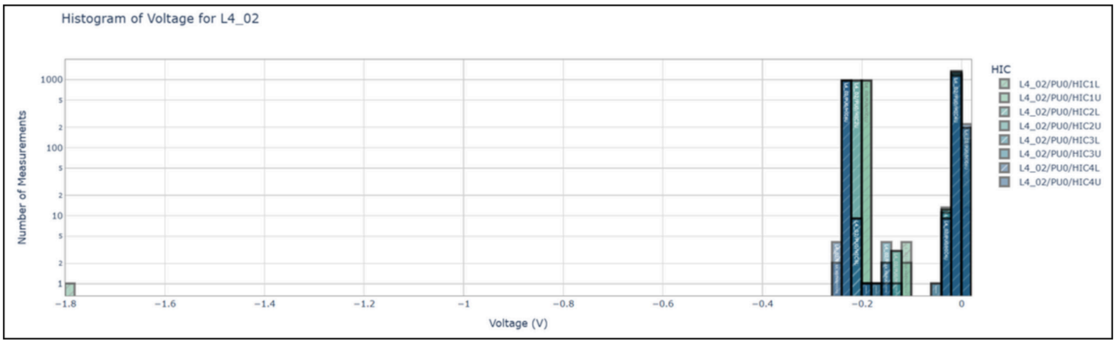


Figure 3: Histogram per stave for layer 4, stave 2

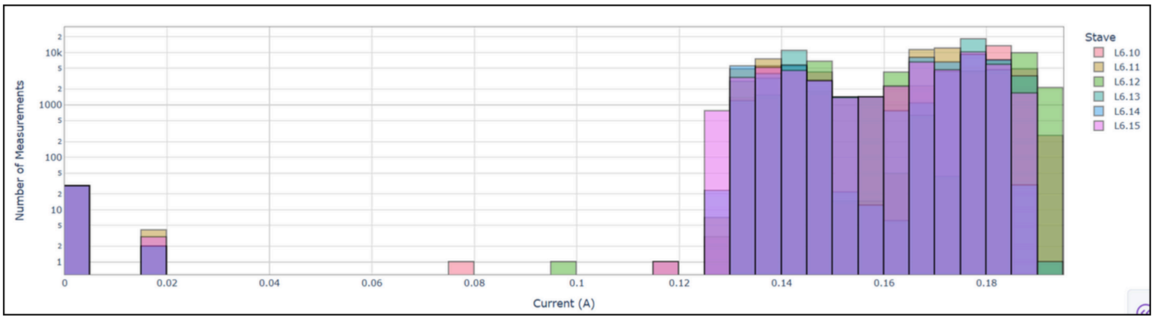


Figure 4: Combined histogram of staves per layer, shown for layer 6 (staves 10–15)

REFERENCES

[1] The ALICE collaboration, The ALICE experiment at the CERN LHC, vol. 3, 2008, S08002, doi: 10.1088/1748-0221/3/08/S08002.

[2] ALICE Collaboration, “ALICE upgrades during the LHC Long Shutdown 2,” Journal of Instrumentation, vol. 19, no. 5, P05062 (May 2024), doiI:10.1088/1748-0221/19/05/P05062.

[3] J. Schambach et al., The ALPIDE pixel sensor chip for the ALICE ITS upgrade, J. Instrum. 14 (2019) P05062. doi:10.1088/1748-0221/14/05/P05062

[4] J. Jadlovský, A. Jadlovská, S. Jadlovská, M. Oravec, D. Voščček, M. Kopčík, J. Čabala, M. Tkáčik, P. Chochula, and O. Pinazza, “Communication Architecture of the Detector Control System for the Inner Tracking System,” in Proc. **16th Intl. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS ’17)**, Barcelona, Spain, Oct. 2017, JACoW Publishing, Geneva, Switzerland. doi:10.18429/JACoW-ICALEPCS2017-THPHA208.

[5] J. Jadlovský, P. Bond, J. Cabala, J. Cerkala, P. Chochula, E. Hanc, A. Jadlovská, S. Jadlovská, M. Kopčík, M. Oravec, M. Tkáčik, and D. Voščček, “Information system for ALICE experiment data access,” in *Proc. 16th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS ’17)*, Barcelona, Spain, Oct. 2017, paper THPHA041, JACoW Publishing, Geneva, Switzerland, 2018. doi:10.18429/JACoW-ICALEPCS2017-THPHA041