Automation efforts for System Testing of ATLAS Inner Tracker Loaded Local Supports

Student: Nguyen Lan Anh
Supervisors: Benedikt Vormwald, Anna Swoboda

August, 2025

Abstract

For the High-Luminosity Large Hadron Collider (HL-LHC) upgrade, the ATLAS Inner Tracker (ITk) is being developed to replace the Inner Detector. A key subsystem of the ITk is the Loaded Local Support (LLS), a feature-complete detector unit, consisting of Pixel Detector modules loaded on the local mechanical supports. In order to verify the full readout chain and ensure the modules' performance after loading, systems tests are being carried out through multiple tests and with sophisticated framework and tools. There are continuous and ongoing automation efforts, aiming at maximizing efficiency for the whole process and individual tests. In this project, two main tasks were achieved: Adaptation of the measurement tools with the Detector System Control (DCS) interface for fully automated thermal performance test, and development of a Plotting Tool of analysis results from Analysis Manager. The LLS and its system tests are configured and controlled using the DCS. Analysis Manager is the user interface and orchestration tool for analysis of quality-control (QC) data from the LLS.

Contents

1 Introduction

1.1 Overview of the system

For the upgrade of the Large Hadron Collider (LHC) to High-Luminosity Large Hadron Collider (HL-LHC), a completely new innermost tracking detector, the Inner Tracker (ITk) (Aad & ATLAS Collaboration, 2017), is being developed to replace ATLAS's Inner Detector to work in the extreme conditions of HL-LHC. The ITk will consist of a Strip Tracker in the outer region and a Pixel Detector in the inner region. The work for this project lies within the Pixel Detector part, which will have five active layers, making up three subsystems, as shown in Figure 1. The the Inner System (IS) consists of the first sensing layers. The Outer System, which consists of the remaining layers, is divided into a central Outer Barrel (OB) (ATLAS ITk Collaboration, 2023) and two identical Endcaps (EC) at either side.

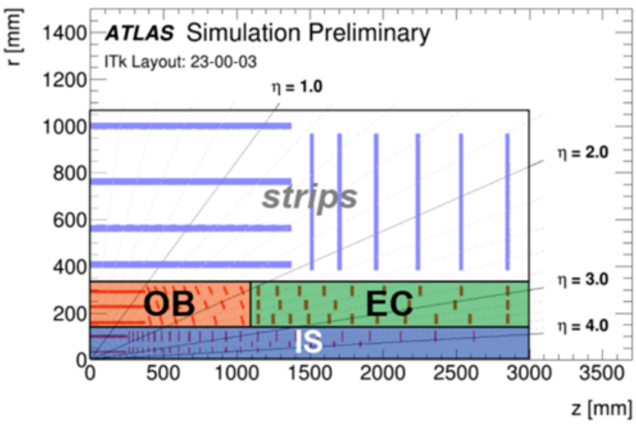

Figure 1: Overview of the Inner Tracker layout, where the Pixel Detector is presented by three colored subsystems: OB, EC and IS. (Vormwald & ATLAS Collaboration, 2024)

More precisely, this project aims to contribute to the automation of the system testing procedures of Outer Barrel (OB) Loaded Local Supports (LLS). An LLS is the smallest feature-complete detector unit, where Silicon Pixel module cells are loaded onto a local mechanical support. An LLS in the context of the Inner Tracker can be seen in the block digram in Figure 2.

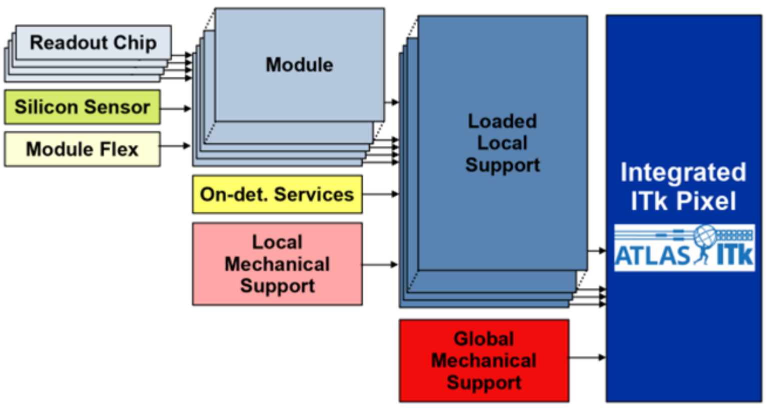

Figure 2: Building blocks of the ITk Pixel detector. (Vormwald & ATLAS Collaboration, 2024)

The module that is loaded onto the LLS structure, as described by Figures 2 and 3, consists of a silicon sensor bump-bonded to four readout chips and a flex PCB. The readout chips will be referred to hereafter as Frontends (FE).

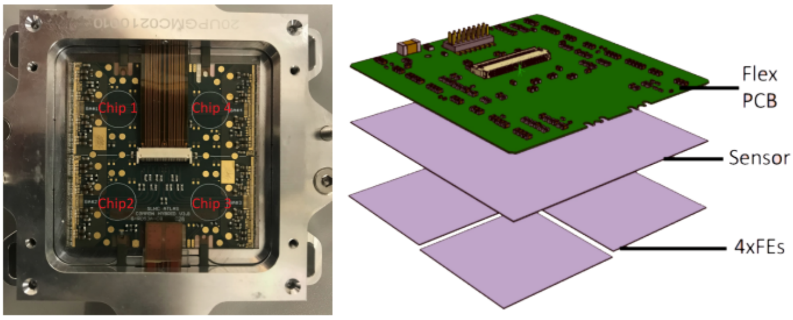

Figure 3: Assembled quad module (left) and exploded view of the module (right). (ATLAS ITk Collaboration, 2021)

In the OB, the mechanical local support can be either a Longeron in the central flat section, or an Inclined Halfring in inclined regions.

1.2 Serial Powering and Data Readout System

The modules on the LLS are serially powered (SP) by a constant current source, making up SP-chains. The power supply is connected to a single module, then the current is propagated through the chain, delivering power to all modules. An LLS has two or four SP-chains and one cooling loop, with organization similar to the definition in Figure 4. For a Longeron, a 6-module SP-chain is placed in the central part, and a 12-module SP-chain is on the outer side.

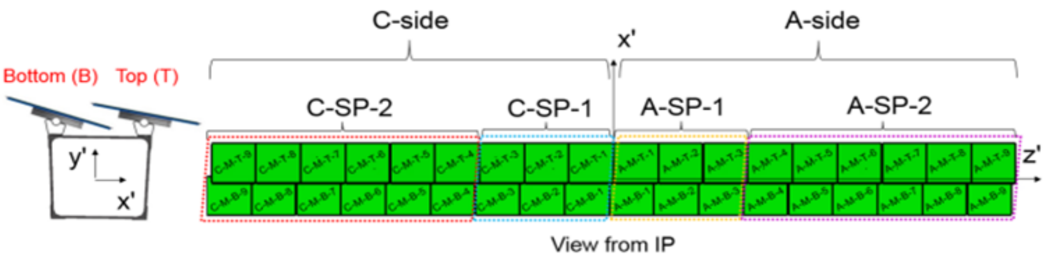

Figure 4: Definition of the serial powering chains in an Outer Barrel Longeron. (ATLAS ITk Collaboration, 2023)

The readout chain

The readout chain for the OB region is demonstrated in Figure 5. A module with one sensor and four frontends is glued to a flexible PCB. A pigtail that connects the module with the Patch Panel 0 (PP0), is responsible for delivering power and transmitting data to the module, as well as connection to the Detector Control System (DCS). Monitoring of Pixel System (MOPS) chip, which a part of the Control and Feedback path of the DCS, are also placed on PP0 to monitor the voltages and temperatures of each module in the SP-chain. Electrical links transmit data at short distances to an Optoboard. The Optoboard converts the electrical into optical signals and sends the data to the DAQ system, which consists of Front-End Link eXchange (FELIX) boards.

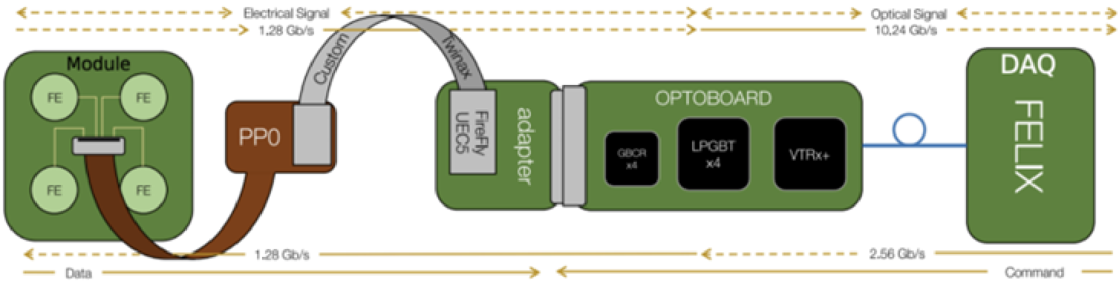

Figure 5: Diagram of the readout chain.

1.3 LLS System tests and Our LLS setup

The LLS quality-control (QC) system tests, including a variety of electrical and thermal tests, verify the full readout chain and the modules' performance after being loaded onto the local support. Our setup for testing is a half-populated Longeron in Building 154, pictured in Figure 6 below. The setup consists of 18 quad modules (4 FEs each).

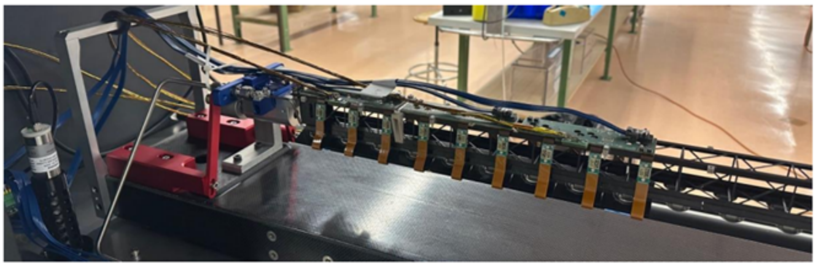

Figure 6: Our test setup: Half-populated Longeron.

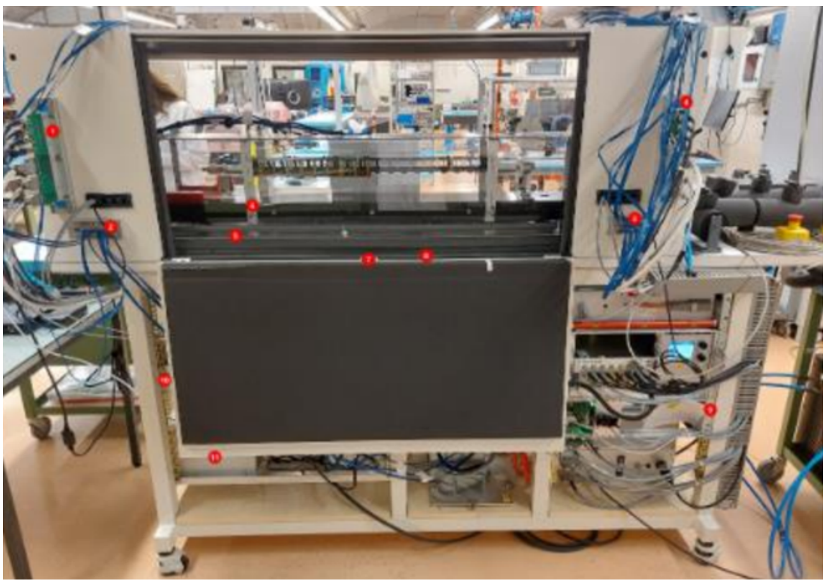

Figure 7: The setup is put in a box which acts as a Faraday cage.

The Longeron is put in a box, which acts as a Faraday cage, with interlock, monitoring of temperature, dew point and connected to a CO2 cooling plant. The setup is currently powered, operational with the Optoboard and FELIX DAQ, and controllable with DCS software interfaces.

1.4 Objectives of the project

Two main tasks were achieved as efforts for the automation of LLS system testing:

- Adaptation of the measurement tools with the DCS interface for fully automated Thermal Performance test.
- Development of the Analysis Manager Plotting Tool.

The following sections of this report will describe these two tasks in detail and in mentioned order.

2 Adapting the measurement tools with the DCS interface for fully automated Thermal Performance test

The goal of this task is adapting existing tools to ensure compatibility with data readout through the Detector Control System (DCS). The existing measurement tools (ATLAS ITk Collaboration, n.d.) were developed by the module group to perform QC on single modules in a lab setup, rather than multiple modules loaded onto a local support in our case. Our LLS setup is controlled and monitored by the DCS, hence, the measurement needs to be adapted to the DCS interface, by implementing DCS API calls, to acquire all necessary data for thermal performance test of an LLS. Specifications of our LLS are adapted for the measuring process; however, the procedure is expected to work on similar setups at other testing sites. Once the workflow is validated, the tools will be integrated into a Docker image, enabling proper setup and configuration of the measurement environment.

2.1 Thermal Performance test and existing tools

The Thermal Performance test measures temperature of the NTC and MOS sensors of a specified module, in order to verify that these temperature sensors provide consistent and reliable measurements that can be used for long-term detector operation and safety monitoring. As mentioned before, the measurement tool for this test needs adaptation with our LLS the DCS interface. Specifically, we will retrieve the data of module's voltage, environment temperature, External NTC temperature and the CO2 cooling heads temperatures from the DCS Server.

Detector Control System and workflow of the measurement

The Detector Control System (Joos, 2022) is developed to ensure operational safety and control of multi-module assemblies, like the LLS. The DCS models the LLS by a Finite State Machine (FSM), allows configuration of physical devices (modules, power supplies, environmental sensors, etc.). Their operating states are exposed as endpoints in the WinCC Open Architecture (WinCC OA) project, allowing retrieval of data by API calls. For the Thermal Performance test, we will read out data from MOPS chip.

The API calls are implemented for different measuring functions in a dedicated Python script (winccoaRemote.py). These functions are then referenced in hardware modules (for example: power_supply.py) and in a configuration file. The measurement tool, from module-qc-measuremet-tools, loads the configuration file and uses said hardware module to call functions for measurement. Results are fed into a defined measurement framework and also output a structured .json file.

2.2 Specific contribution

- Added methods in the DCS winccoaRemote.py for measurements of:
  - Module voltage: current measurement tool measures the absolute voltage of the module, while in an SP-chain, each module's operating voltage is defined by the voltage drop across it. The added method retrieves the SP-chain voltages, filters out the module's voltage and computes its voltage drop.
  - Environmental temperature: retrieves the value of the NTC sensor for environmental temperature.
  - External NTC Temperature: retrieves the value of the External NTC sensor.
  - Cooling heads' temperatures: retrieves the temperatures of the cooling heads coupled to the module and to the CO2 cooling circuit.
- Updated the functions of hardware module and the configuration file.
- Updated the measurement script with defined functions and mapped the measurements to the correct output variables.

2.3 Result

The workflow was tested successfully on the setup, demonstrated expected functionality and produced desired output. The measurement command can be executed from the computer's terminal, then the process runs automatically and outputs all needed data.

The results were successfully validated: The temperatures had the same values as the monitor on the DCS software interface. The module voltage drop was around 1.6V, as expected for modules in SP-chain.

2.4 Future work

Since the tool worked, the following steps can be taken:

- Docker containerization: Create a Dockerfile that installs and prepares the updated measurement tool for Thermal Performance test. The resulting Docker container will be pushed to the Registry, available for distribution and deployment at other sites.
- Implement the same approach for other tests, such as SLDO or IV-curves tests.

3 Development of Analysis Manager Plotting Tool

3.1 Motivation for the tool

The pixel performance test evaluates performance of each pixel in a module. The test involves several stages performed at both warm and cold temperatures. At each stage, QC analyses are performed for all modules, done by the module-qc-analysis tool, integrated into the Analysis Manager, which is a microservice in the DeMi (Distributed Detector Microservices) framework. The Plotting tool was developed to:

- Provide visualization of analysis results of FEs/modules for comparison between testing stages.

- Provide flexible selection of test type, modules, and results for generating plots.

3.2 Analysis Manager and ConfigDB

A Microservices framework called Distributed Detector Microservices (DeMi) (Schmeing & ATLAS Collaboration, 2024) is developed by Jonas Schmeing and Wolfgang Wagner's team at University of Wuppertal to orchestrate the ITk system tests. The framework is a collection of microservices; each has a specific function that can operate independently. Each microservice also comes with an HTTP-REST-API for communication with other microservices, and an user interface, along with a backend that handles their specific function.

DeMi and all its components and dependencies, are deployed within immutable Docker containers, facilitating independent deployment, modification and scalability across different sites. The dashboard UI in Figure 8 is the entry point into the system deployed for our setup at CERN, showing all microservices APIs and GUIs and allowing direct access to each service. For the quality control analysis, Analysis Manager is employed. ConfigDB stores both configuration files for the hardware configuration and the analysis results. Therefore, the development of the plotting tool involved the Analysis Manager and ConfigDB.

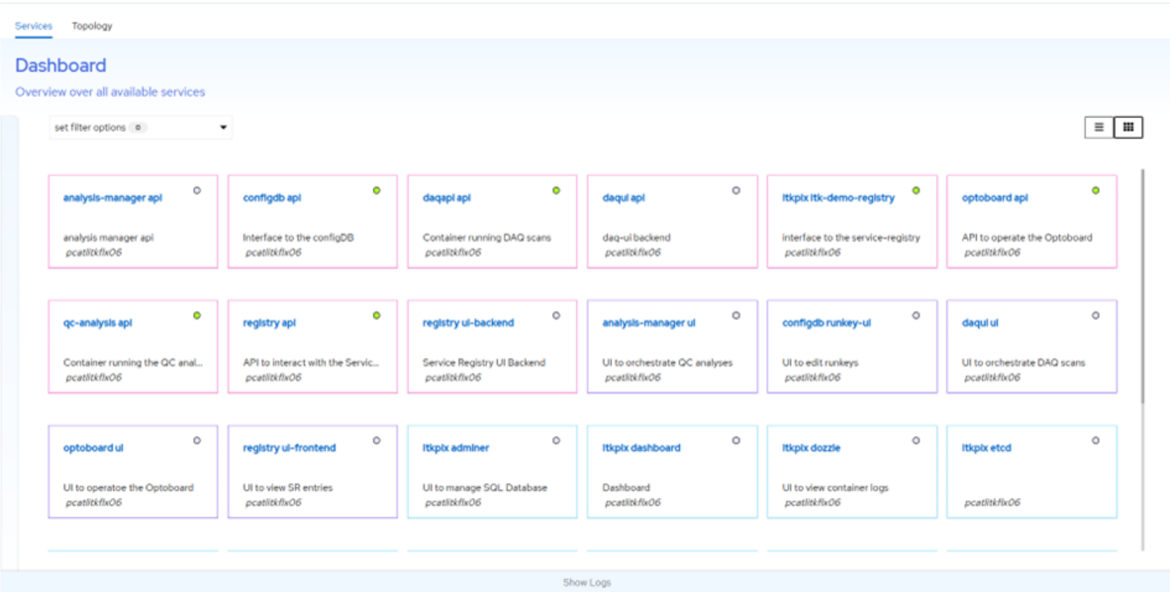

Figure 8: Dashboard UI for all microservices.

Analysis Manager

Analysis Manager is a user interface and orchestration tool that allows:

- Loading and selecting LLS data
- Selecting the QC analysis to be performed (Minimum Health Test, Pixel Failure Analysis, electrical tests).
- Submitting analysis jobs (which runs the analysis with the module-qc-analysis-tools)
- Displaying results of the analysis (scans, histograms, etc.)

ConfigDB

ConfigDB is the configuration database microservice that provides the configuration files needed to configure the readout hardware. Scans and analysis results are also stored in the database and can be retrieved using API calls.

3.3 Development of the tool

3.3.1 PyQt5 application

First, the Plotting Tool was developed as a PyQt5-based application.

Backend functions were implemented to: retrieve analysis metadata, retrieve hardware metadata, retrieve available analysis tags, modules and analysis results from ConfigDB. The data was then cleaned up, filtered and organized. A plotting function was developed using the `matplotlib` package.

Finally, a graphical user interface (GUI) was built with PyQt5, enabling user interaction for selecting inputs, displaying the generated plots, and saving them to a local directory.

Result

Features of the application:

- Search for available analysis tags of an LLS by entering its serial number.
- Support two test types: Minimum Health Test and Pixel Failure Analysis.
- Display available analysis tags (for the selected LLS and test type) and corresponding modules, with the ability to select desired tags and modules.
- Display available results, allow multiple result selections.
- Provide a robust interface to change or update selections without restarting application.
- Generate plots with the following configuration:
  - x-axis: FE Chip ID (bottom) and module serial number (top)
  - y-axis: Result values
  - Legend: Analysis tags
  - Information box: Lists FEs without valid data for a given analysis tag
- Save opening plots as .png files to a user-specified directory.

The Plotting Tool GUI is presented in Figures below:

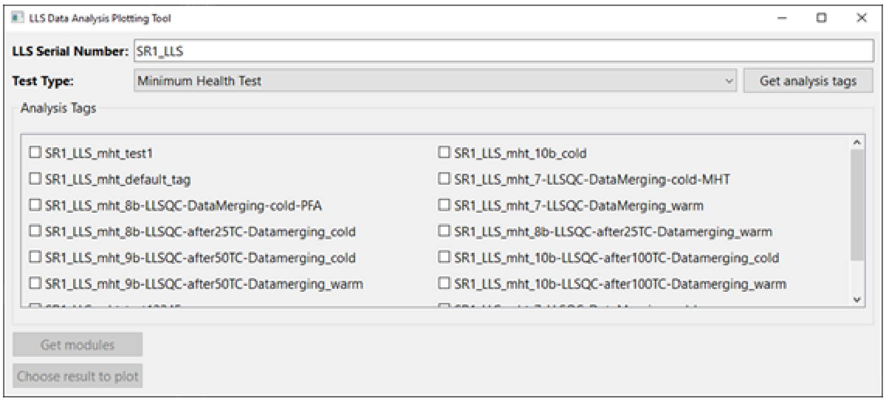

Figure 9: After entering the LLS serial number (`SR1_LLS`) and selecting the test type in the application, clicking "Get analysis tags" retrieves and displays all available analyses performed for that specific LLS and test type, by their tag names.

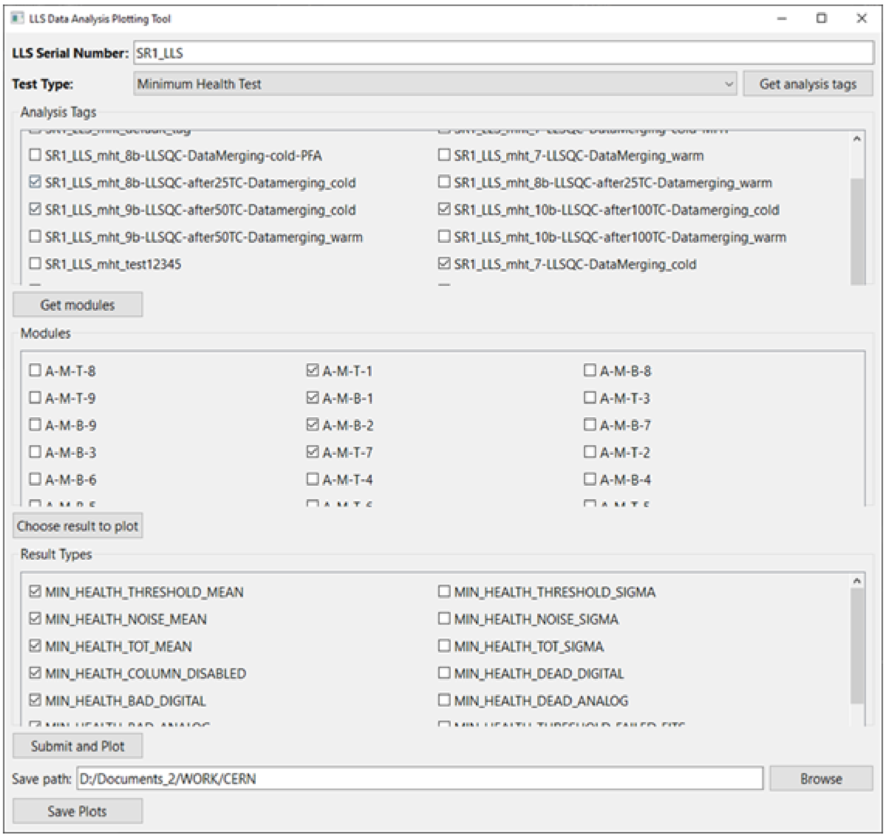

Figure 10: Similarly, after selecting the analysis tags to plot, the available modules are retrieved and displayed according to their positions on the LLS. Once the desired modules are chosen, their results are fetched and shown. Clicking "Submit and Plot" generates all plots simultaneously, and these plots can be saved by specifying or browsing to a local directory.

Figures 11 and 12 are example plots generated by the Plotting Tool. The plots can provide a quick overview of the result, as well as quick identification of failing chip/module. Comparing results across temperature stages can also provides valuable information on the module's physical response and stability.

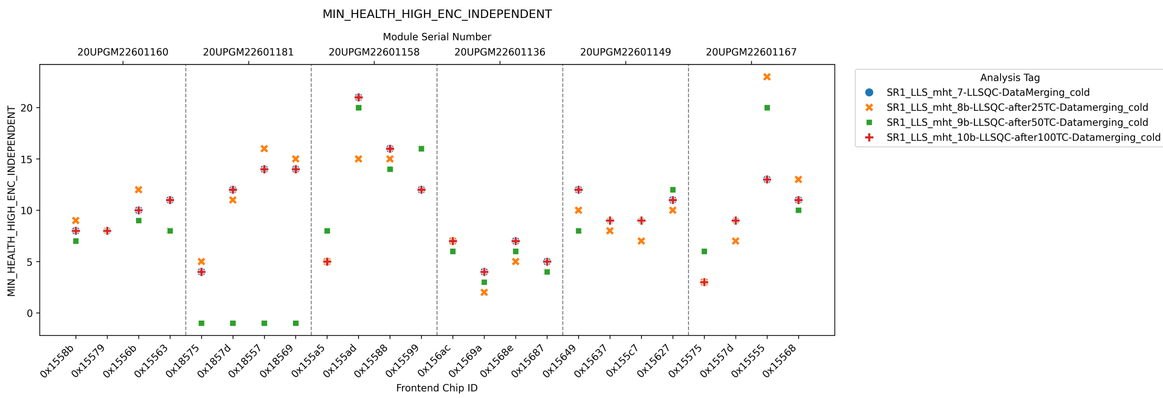

Figure 11: MIN HEALTH HIGH ENC INDEPENDENT plot for cold-temperature stages.

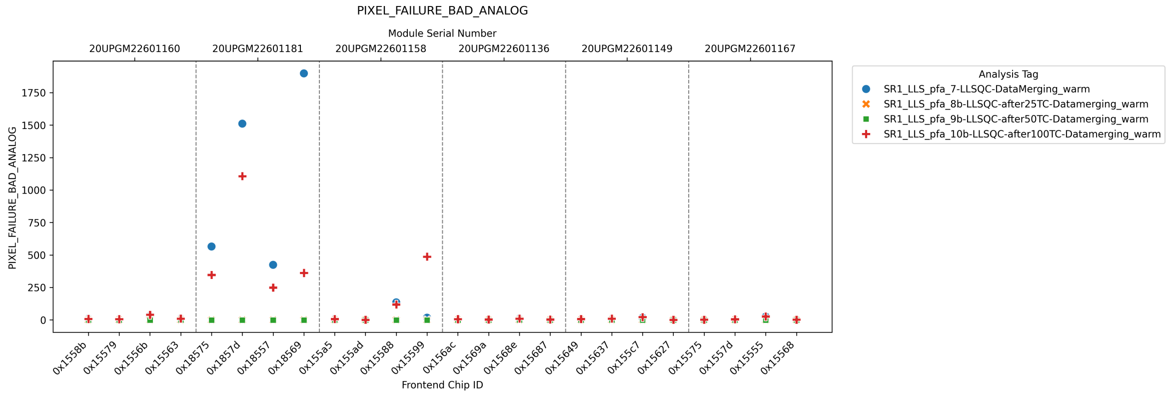

Figure 12: PIXEL FAILURE BAD ANALOG plot for warm-temperature stages.

3.3.2 Panel on Analysis Manager

As an update of the Analysis Manager, it was agreed that embedding the Plotting Tool as panel on the Analysis Manager would be beneficial for usage at other sites, avoiding the necessity to install and run the application separately.

For development of the panel, most of the backend functions from the PyQt5 application was reused. The interface was rebuilt with Reflex, making it compatible with the Analysis Manager. For the plots, instead of `matplotlib`, `plotly` is a newly investigated package and integrated to generate interactive plots.

Result

The latest progress on the Panel is shown in Figure 13.

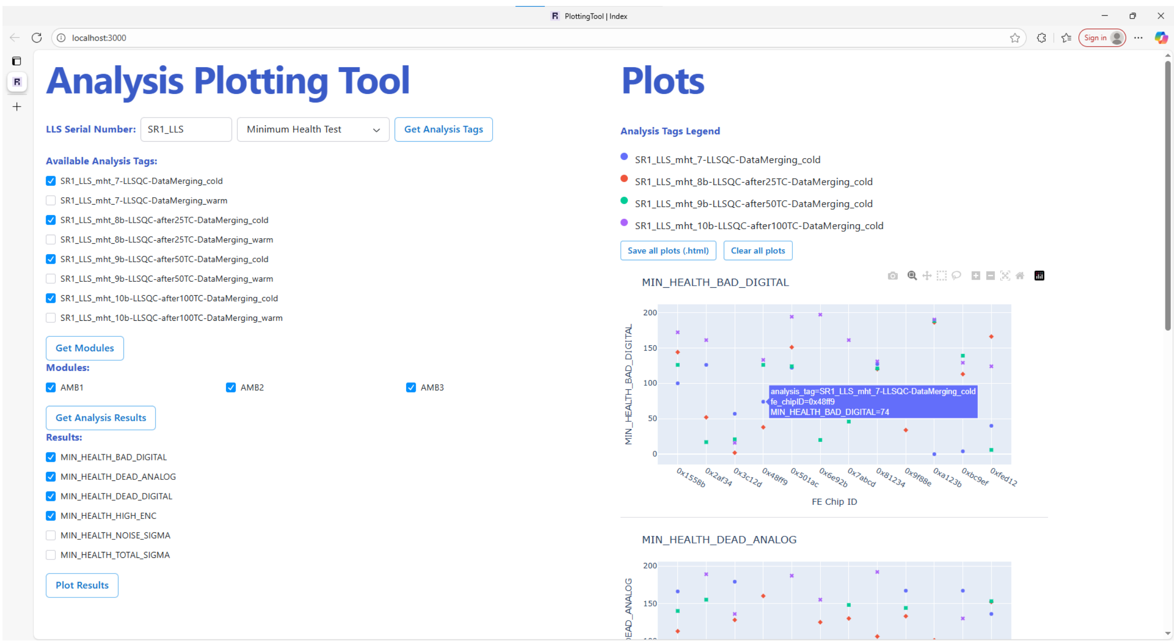

Figure 13: The Plotting Tool as a Panel - current progress.

Features of this panel are mostly similar to that to the PyQt5 version, with a few following differences:

- Generate all plots on the right half of the page.

- Generate plots with the following configuration:
  - x-axis: FE Chip ID
  - y-axis: Result values
  - Legend: Analysis tags (on the top of the plots)
- Plots are interactive, showing information by hovering over data points.
- Download of each plot as .png or of all plots as a .html link.
- Restart by clearing of all plots.

The panel still needs some further implementation to be complete, that includes:

- Top x-axis of module's serial number
- Dynamic update of the plots with changes in selection
- Information box: List FEs without valid data for a given analysis tag

4 Conclusion

LLS system tests' purpose is to verify the readout chain and performance of modules after being loaded. This project contributed to the automation of the process by working on: (1) Measurement tool for the Thermal Performance test, and (2) the Plotting Tool for analysis result data from Analysis Manager. For deployment at other LLS testing sites, the measurement tool would need to be containerized; and the panel plotting tool would need further development to completion.

5 Acknowledgement

I would like to express my heartfelt gratitude to my supervisors at CERN: Anna Swoboda and Benedikt Vormwald, who have guided me through every step of this project with their expertise and kindness. Special thanks to Estel Perez Codina, Jonas Schmeing, Hans Joos and Yuma Uematsu for offering me their generous help with my tasks.

I would also like to thank CERN, the ITk Pixel Team and Coordinators of the Summer Student Program for providing me with this invaluable opportunity to learn and work.

My sincere thanks to my Department of Space and Earth Sciences at the University of Science and Technology of Hanoi for their efforts in enabling my participation in CERN's program, as well as their continued support.

Finally, I am deeply grateful to my classmates and the friends I have made at CERN, who have made this journey of mine incredibly happy and meaningful.

References

Aad, G., & ATLAS Collaboration. (2017). *Technical design report for the atlas inner tracker pixel detector* (tech. rep.). CERN. https://doi.org/10.17181/CERN.FOZZ.ZP3Q

ATLAS ITk Collaboration. (2021). *Module development for the atlas itk pixel detector* (tech. rep. No. ATL-UPGRADE-PROC-2021-003). CERN. https://cds.cern.ch/record/2782433/files/ATL-UPGRADE-PROC-2021-003.pdf

ATLAS ITk Collaboration. (2023). *Design overview of the loaded local supports for the itk pixel outer barrel - design vr110_01032023* (tech. rep.). CERN.

ATLAS ITk Collaboration. (n.d.). Module qc measurement tools [Created July 12, 2022]. https://gitlab.cern.ch/atlas-itk/pixel/module/module-qc-tools

Joos, H. (2022). *Developement of the detector control system of the itk outer barrel demonstrator for the atlas experiment* [Master's thesis, University of Gottingen]. https://www.uni-goettingen.de/de/document/download/93bbaeddd4d78b69655a8363aefefad7.pdf/MSc_Joos_Hans.pdf

Schmeing, J., & ATLAS Collaboration. (2024). A modular approach to software in atlas: Microservices framework and configuration database for the atlas itk. *Proceedings of the 27th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2024)*.

Vormwald, B., & ATLAS Collaboration. (2024). Performance of the atlas itk pixel detector prototype. *JPS Conference Proceedings*, *42*, 011015. https://doi.org/10.7566/JPSCP.42.011015