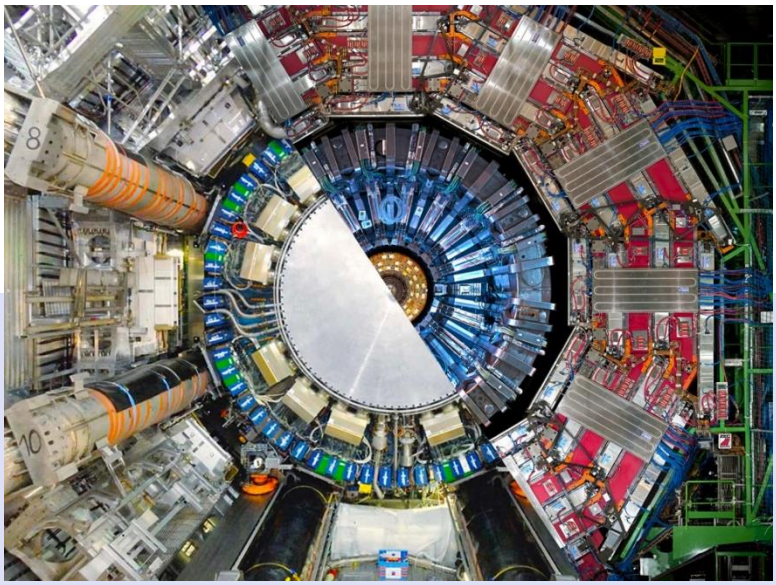


Establishing Communication Between MTD DAQ Software and Integrated xDAQ Services

Prepared by
Maryam Kamashki

Supervisor
Mehmet Ozgur Sahin

DATE

August 18, 2024

Acknowledgement

Firstly, I would like to express my deepest gratitude to my supervisor, Dr. Mehmet Özgür, for his continuous guidance, mentorship, and belief in my capabilities. His expertise and dedication have significantly shaped the course of my project.

I am profoundly thankful to Dr. Mohamed Yusuf for his unwavering encouragement and for the opportunity to explore the Bahrain CERN Exhibition. His support and insights have been invaluable to my journey.

My sincere thanks go to my co-supervisor, Hasan Aljamea, whose support and constructive feedback have been crucial throughout this project.

I am also grateful to my university supervisor, Dr. Abdulla Alasaadi, for his wise counsel and academic guidance.

I would like to extend my appreciation to my advisor, Dr. Faisal Alqaed, for his insightful advice and for helping me navigate the complexities of this training.

Special thanks to my colleagues Noor, Alaa, Sana, Abdulla, and Mohamed, for their encouragement, and the collaborative spirit that made this journey enjoyable.

Most importantly, I want to give my heartfelt thanks to my parents. This journey would not have been possible without their unwavering love, support, and sacrifices. Their belief in me has been my greatest strength.

Table of Contents

Introduction

The MIP Timing Detector (MTD) is poised to become a critical component of the Compact Muon Solenoid (CMS) experiment, significantly enhancing its capabilities in event reconstruction and particle identification through precise timing resolution.

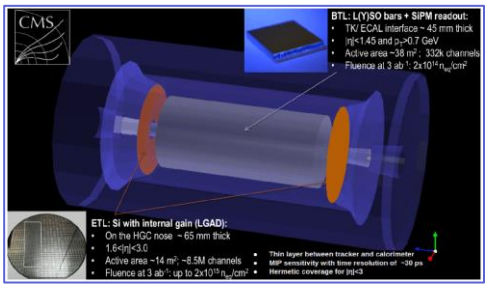


FIGURE 1: MTD ARCHITECTURE

Figure 1 shows the two main components of the MIP Timing Detector (MTD), each providing precise timing information for their respective regions within the CMS detector. The first component is the Barrel Timing Layer (BTL), which encompasses the central barrel region of the detector. The second component is the Endcap Timing Layer (ETL), located at the detector's endcaps, covering the forward and backward regions. These layers work together to achieve a timing resolution target of 30-35 picoseconds. Although it is currently not installed, the MTD is planned for integration into the CMS between the years 2026 and 2028, with full operational capabilities expected by 2029.

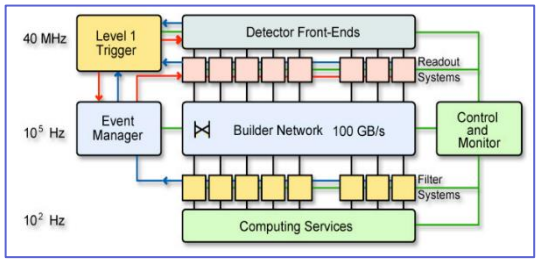


FIGURE 2: THE CMS DATA ACQUISITION

Central to the operation of the MTD is its Data Acquisition (DAQ) system. DAQ, short for Data Acquisition, is the process of collecting, processing, and storing the data generated by particle detectors during experiments. The CMS experiment relies on multiple DAQ systems seen in Figure 2, with the primary CMS DAQ responsible for reading out data from all connected sub-systems. Integration of the MTD DAQ systems involves a cross-stack approach, which makes it essential for ensuring seamless data collection and processing.

Given that each sub-detector within the CMS has its own specialized DAQ development, achieving a harmonious connection with the central CMS DAQ system is critical. The MTD DAQ and control system will play an integral role in this, ensuring that the MTD operates efficiently within the broader CMS infrastructure. This integration is vital not only for maintaining precise timing but also for managing data effectively and enabling real-time event processing.

This report details the development, integration, and operational strategies for the MTD DAQ system within the CMS experiment, highlighting its importance in the pursuit of accurate and efficient data acquisition.

Background

xDAQ Services

The xDAQ framework plays a key role in the development, deployment, and operation of distributed Data Acquisition (DAQ) systems, such as the MIP Timing Detector (MTD). As you can see in Figure 3, the framework integrates several components, each with distinct roles:

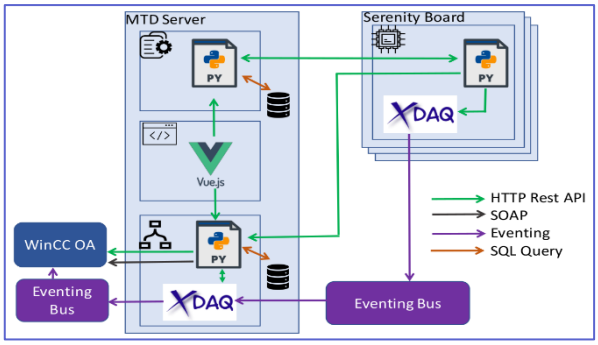


FIGURE 3: MTD DAQ SERVICES ARCHITECTURE

- **MTD Configurator (Top Left):**
  Manages MTD electronics, handles HTTP/RESTful requests, stores configurations, and provides a GUI for interaction.

- **MTD Controller (Right):**
  Deploys on multiple Serenity boards, executes commands from the Configurator, and sends monitoring data through the xDAQ Eventing bus.

- **MTD Collector (Bottom Left):**
  Gathers and monitors sensor data and logs from controllers and provides data access via the GUI.

- **MTD GUI (Middle Left):**
  Provides a web application built using Vue.js, allowing operators to interact with and monitor the DAQ system with ease.

These services collectively ensure that the MTD DAQ system is configured correctly, monitored effectively, and can be interacted with through a user-friendly interface. The communication between these components is facilitated through various protocols, including HTTP RESTful APIs, SOAP, Eventing, and SQL Queries, as shown in the figure.

MTD Controller

To take a closer look on the MTD controller, it primarily uses two interfaces to manage the muon detection modules:

- **Python Interface:**
  Receives configurations and commands from the MTD Configurator and transmits monitoring data to other components through the event bus.

- **xDAQ Interface:**
  Sends monitoring data to the event bus, facilitating communication with external systems.

These interfaces enable the Controller to effectively execute configurations and maintain seamless interaction with both internal and external system components.

MTD GUI

Next, an important component in this project is the graphical user interface (GUI). The MTD GUI facilitates interaction with xDAQ services via a web application using RESTful APIs. It features a command page where users can execute commands and view outputs.

My contribution to this project included developing Python scripts that the MTD Controller runs through the GUI. These scripts are designed to execute commands specifically targeting the ASICs, which will be detailed further in the next section.

Collaboration (ASICs)

Effective collaboration is crucial for the success of the MTD DAQ project. Here's an overview of the main components involved:

SCA (Slow Control Adapter)

As its name suggests, the SCA manages slow control functions by monitoring DAQ component health, sending control commands, logging performance data, and ensuring synchronous communication with the LpGBT and other chips.

LpGBT (Low Power Giga-Bit Transceiver)

- **ADC (Analog-to-Digital Converter):**
  Reads analog data from sensors such as temperature sensors then converts those analog signals into digital values.

- **GPIO (General-Purpose Input/Output):**
  Features 16 Input/ Output pins for synchronous operations.

CC (Chip Classes)

Various chips within the MTD DAQ system are categorized into distinct "chip classes" based on their functionality and architecture. See Figure 4 below.

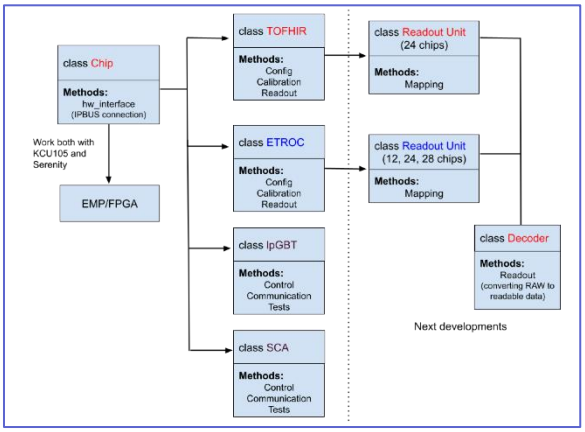


FIGURE 4: MAP OF CHIP CLASSES

Summer Program Summary

During my stay as a summer student at CERN, I had the unique opportunity to participate in a variety of activities and contribute to significant projects. In the first week, I visited three key sites: the ATLAS and Compact Muon Solenoid (CMS) detectors, as well as the Antiproton Decelerator (AD) factory. Additionally, I toured CERN's Data Center, gaining insights into the infrastructure that supports its revolutionary research.

Throughout the program, I attended morning lectures that provided a strong theoretical foundation in particle physics and advanced computing. I also participated in several specialized workshops, including ROOT, a data analysis framework widely used in high-energy physics, and web application security penetration testing, which offered hands-on experience in safeguarding digital systems.

My project involved contributing to the Data Acquisition (DAQ) system, which is responsible for collecting, processing, and storing data generated during particle physics experiments. It was specifically focused on the MTD DAQ system, where I worked on establishing communication between the MTD DAQ software and integrated xDAQ services. The Minimum Ionizing Particle Timing Detector (MTD), set to be a crucial component of the Compact Muon Solenoid (CMS), will enhance timing resolution for precise event reconstruction and particle identification.

To ensure seamless integration of the MTD DAQ with the central CMS DAQ system, I worked with the xDAQ software framework, concentrating on the MTD Controller, and MTD graphical user interface (GUI). My role included writing Python scripts to be executed by the MTD Controller via the GUI, which enabled precise control and monitoring of various chips, particularly the ASICs.

The project involved collaboration with essential components like the Low Power Giga-Bit Transceiver (LpGBT) and the Slow Control Adapter (SCA) ASICs, which are critical for data conversion, monitoring, and communication within the MTD DAQ system.

As the project neared completion, testing the scripts on real hardware in the lab became the final and crucial step, ensuring all systems function correctly before full deployment.

Conclusion

As we approach the final stages of the MTD DAQ project, we are on the brink of a significant milestone. The development and integration of the MTD DAQ system, including the xDAQ framework and the MTD GUI, have progressed smoothly, setting the stage for the crucial next step: testing.

The Python scripts I developed are designed to manage and control various ASICs within the MTD system, ensuring precise functionality and integration with the xDAQ services. With these scripts now prepared, our focus shifts to rigorous testing on the actual hardware in the lab.

This final testing phase is crucial. It will validate that the scripts perform as expected, verify the correct operation of all components, and allow us to make any necessary adjustments before the full deployment of the MTD system. Successfully completing this step will confirm that the MTD DAQ system is ready for

its integration into the CMS experiment, enhancing the precision and efficiency of data acquisition for future particle physics research.

In summary, the project is nearing its completion, and the upcoming testing phase will be pivotal in ensuring that all systems are fully operational and ready for deployment.

Appendix

Appendix A: MTD xDAQ Services Architecture

The diagram below illustrates the architecture of the MTD DAQ services, which is divided into four main components: the MTD Configurator (top left section) the MTD GUI (middle left section), the MTD Collector (bottom left section) and the MTD controller (right side, in the section labeled as "Serenity Board"). The red-bordered areas indicate the specific sections where my contributions were focused. On the left side, within the MTD GUI, I was responsible for adding and updating the JSON file from which the Vue.js front-end interface retrieves its data. On the right side, for the MTD controller, my work involved scripting in Python that are responsible for issuing commands that configure the ASICs such as LpGBT and SCA. The components communicate through HTTP REST APIs, SOAP protocols, and Eventing bus, as depicted by the colored arrows in the diagram.

Appendix B: Python Script

Below is a snippet illustrating how the controller executes commands to configure the LpGBT ASIC. As you can see, we use a decorator to wrap the mtddaqfunc function. This decorator is responsible for capturing and returning the console output generated during the execution of mtddaqfunc. The function dynamically imports the lpgbt_main module from a specified file path (where the script resides) and invokes its lpgbt_main function with the necessary parameters (arguments of configuring the chip).

```python
@command_route.post('/lpgbt_main')
async def lpgbt_main(payload: LPBGTMainPayLoad) -> FunctionOutput:
    @capture_console_output_and_return
    def mtddaqfunc():
        mtd_module = import_module_from_file(repo_rel_path=Path('lpgbt_main.py'))
        return mtd_module.lpgbt_main(
            str(get_repository_root()),
            link=payload.link,
            connection=payload.connection,
            read_length=1,
            address="0x1d9")

    return mtddaqfunc()
```

FIGURE 5

Please note that I cannot provide the actual script used to configure the LpGBT ASIC as it is confidential. However, the provided snippet demonstrates how the system interfaces with the configuration script and captures its output.

Appendix C: JSON Configuration

This snippet shows the structure of the JSON file, including fields like CommandName, URL, and Arguments, with details about their expected values and validation rules. Please note that I cannot provide the complete JSON file due to confidentiality. However, this excerpt should give you an understanding of how the configuration data is organized.

```
{
  "CommandName":"initialize SCA",
  "URL":"sca_main",
  "Arguments": [
    {
      "Link": {
        "Value": 4,
        "Nullable": true,
        "validation": "number"
      }
    },
    {
      "fpga": {
        "Value": "x0",
        "Nullable": true,
        "validation": "text"
      }
    },
    {
      "Connection": {
```

FIGURE 6

Appendix D: In the Lab

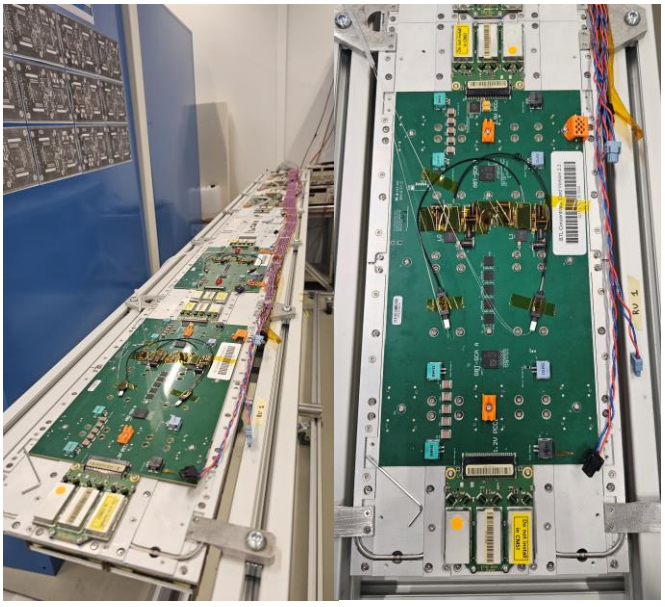


FIGURE 7

On the left (See Figure 7) you can see a section of the MIP Timing Detector (MTD) that features multiple concentrator cards. These cards are crucial components in the MTD's data acquisition system, as they aggregate and process signals from various sensors within the detector.

Each concentrator card houses multiple chips and associated wiring, which manage the data flow from the sensors, converting analog signals into digital data, and transmitting this data to the central DAQ system. The chips are likely responsible for tasks such as data conversion, signal processing, and communication with other parts of the DAQ system. The concentrator cards ensure that the data collected from the detector is efficiently processed and relayed for further analysis, playing a key role in the overall operation of the MTD.

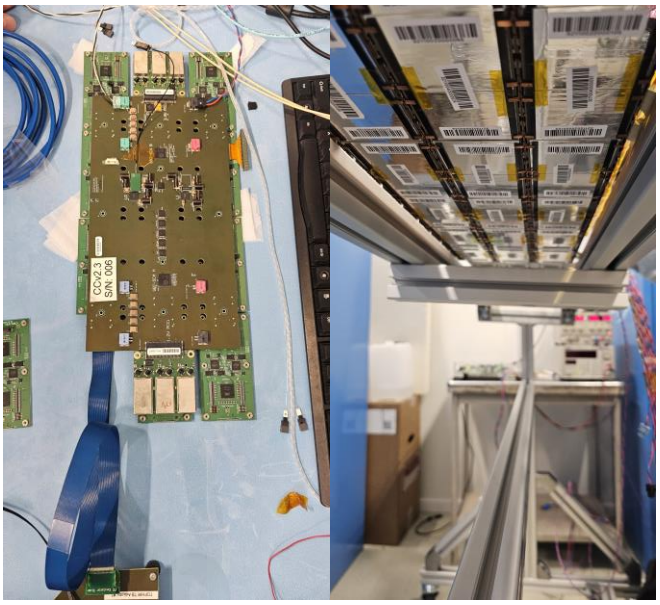


FIGURE 8

Beneath those cards, we have another set of concentrator cards shown in Figure 8. These cards are equipped with TOFHIR (Time-of-Flight High Resolution) chips and other essential components. The TOFHIR chips are specifically designed for precise time measurement, playing a crucial role in the MTD's ability to achieve its target timing resolution of 30-35 picoseconds. This lower layer of concentrator cards is integral to the system's function, as it further refines the timing data, ensuring accurate event reconstruction and particle identification within the CMS detector.

Then, there is a crystal-like structure at the bottom. This part is likely composed of crystals or similar materials that are used to detect particles passing through the MTD. They emit light when struck by high-energy particles, and the emitted light is then collected and converted into electrical signals by the detectors on the concentrator cards above.

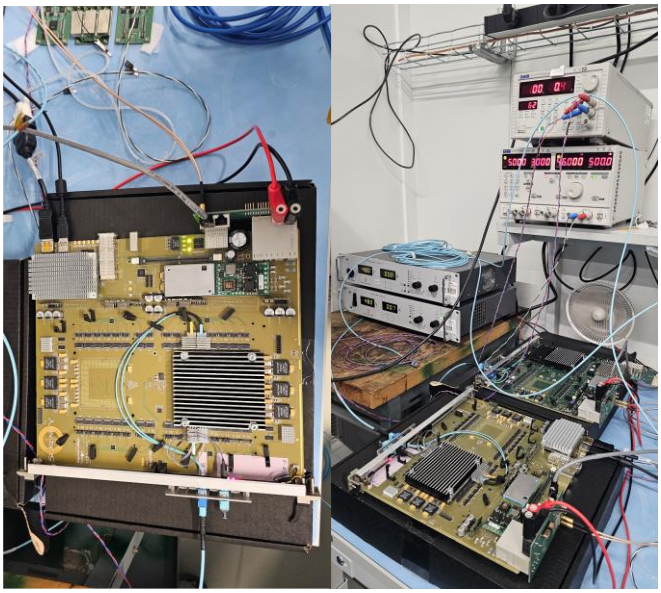
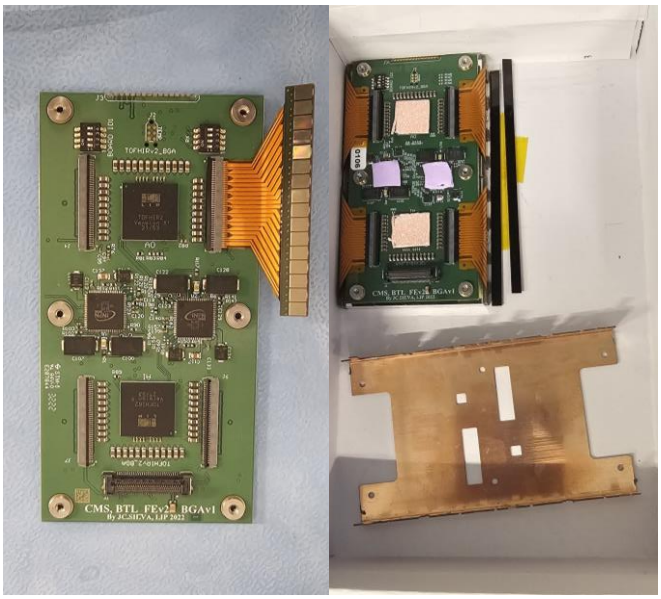


FIGURE 9

Connected to the tray we have what we call a “serenity board” (see Figure 9). The Serenity board is a crucial component in the MIP Timing Detector (MTD) system, used in high-energy physics experiments like the CMS experiment. It features a central FPGA for data processing, multiple heatsinks for thermal management, and various connectivity options for high-speed data transfer and power management. The board is responsible for managing real-time operations within the MTD, executing commands, and controlling muon detection modules to ensure precise timing and data acquisition.



The image in Figure 10 showcases a Front-End Card used in the MIP Timing Detector (MTD) system. This green Printed Circuit Board (PCB) is equipped with two TOFHIR chips and is attached to crystals via silicone photomultipliers, which are the golden-orange components. The board is connected to other components through flex circuits (PCDs) and is protected by a copper case. Thermal plates and pads are used to manage heat, ensuring optimal performance.

FIGURE 10

FIGURE 11: A PICTURE OF ME IN THE LAB