Contribution To The MTD-DAQ ETL Software

Summer student project report

Student: Adnan Mohamed

Supervisors: Mehmet Ozgur Sahin, Giacomo Zecchinelli

Table of Contents

Abstract

This report serves as an overview of my contributions and insights gained during my involvement in the MTD-DAQ project, specifically focusing on the Endcap Timing Layer (ETL) software within the MIP Timing Detector (MTD) at CERN's Compact Muon Solenoid (CMS) experiment. Through a brief exploration of the ETL's hardware components, data flow, and software functionalities, this report aims to provide readers with a concise summary of my work while placing it within the broader context of the MTD-DAQ project. The report covers essential aspects of chip control, configuration, and software functionality, highlighting my journey as a summer intern.

Introduction

In particle physics experiments, conducted at CERN, the MTD will hold a crucial role in the CMS experiment during the High Luminosity LHC (HL-LHC) era [1]. This detector is designed to measure the time of arrival for MIPs. This helps identifying important signals from the background noise in events with many particle interactions happening at once [1]. The MTD's purpose is to provide accurate timing measurements for MIPs, helping us figure out where they were created and linking them to the right interactions [1]. The MTD also has two main parts: the ETL and the Barrel Timing Layer (BTL). These parts work together to measure the timing of MIPs. The ETL is like a sensor on the sides of the CMS, while the BTL is like a sensor around the central part.

The Data Acquisition System (DAQ) is another important part of the MTD, which makes sure all the relevant data from the MTD's sensors and front-end electronics are collected, organized, and sent to the central CMS data system [2]. It does this using different tools like optical links, boards, and different ASICs [1]. In the software side, the MTD-DAQ team is creating a special software using the Python programming language. This software helps control and set up the electronics in the ETL and BTL. Another core aspect of the MTD-DAQ is the EMP framework which plays an important role by providing essential firmware and software libraries to control and monitor hardware [3]. It's like a toolbox that helps create and manage specific programs, such as reconstruction and trigger related algorithms [3]. This framework abstracts the low-level details of connecting and communicating to hardware devices on the ETL and BTL which makes it easier to write high-level programs using Python as we are doing. The software we are working on is hosted on the Serenity. Serenity is an Advanced Telecommunications Computing Architecture (ATCA) prototyping platform, comprising various parts like carrier cards containing FPGAs, chips and CPU [4]. This platform is part of the MTD-DAQ infrastructure, which facilitates the collection, control, and management of data from readout chips within the MTD-DAQ [2].

This report gives an overview of the MTD-DAQ software and highlights my learning efforts and contribution during my 8-weeks internship, focusing on parts of the ETL software of the MTD-DAQ.

The ETL Electronics

The ETL detector has several electronic components with the Readout Unit (RU) building block which consists of: Versatile Transceiver Plus (VTRx+), Low-Power Gigabit Transceiver (lpGBT) chips, Slow Control Adapter (SCA) chip, and Endcap Timing Readout Chips (ETROCs) [1] (see Figure 1). The ETL RU facilitates data transmission between the ETL front-end electronics and the data acquisition system, while also providing control and monitoring functions (e.g., configuring chips and getting temperature readings) for the detector [1]. The DAQ-LpGBT chip provides the means to configure and communicate with the SCA and ETROC chips on the service hybrid via communication protocols (e.g., I2C). The SCA chip serves the purpose of providing environmental information of the electronics (e.g. temperature reading) and configuring the ETROCs. The ETROCs serve as the readout ASICs which provides the required information about the MIPs. Furthermore, the Service Hybrid (SH) is an assembly of two printed circuit boards (PCBs) that provide power and readout services to the modules. The software's responsibility is to provide configuration, control, and communication functionality for the ETL.

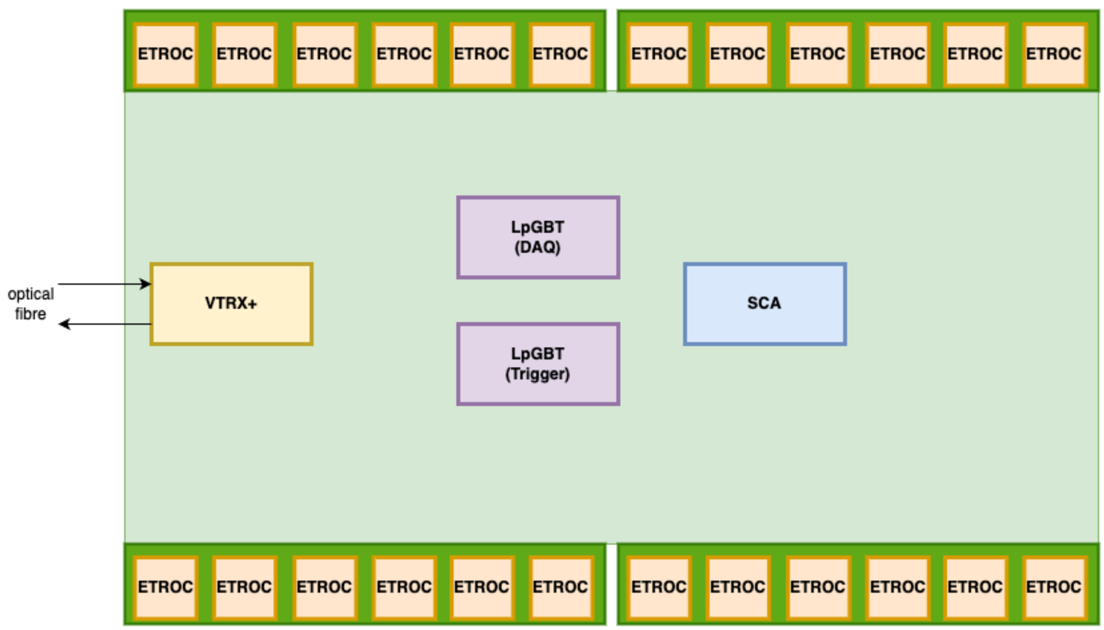

Figure 1: A diagram that illustrates the ETL Readout Unit (RU) component chips.

The ETL Data Flow Overview

The Data Flow Diagram (Figure 2) illustrates how data moves in the ETL readout units and shows the interconnections between the chips on it. Information from ETROCs flows through E-links to LpGBTs on service hybrids, then via optical fibers through VTRx+ chips to back-end electronics boards. Optical links also transmit control signals and timing information. This diagram highlights the functionality that is expected from the software to provide, including:

- Initializing chips with accurate configurations.
- Facilitating communication between the DAQ LpGBT (Transceiver) and other chips through EC and I2C channels
- Managing E-links and ensuring proper frequencies for the ECLKs, as denoted by CLK40.
- Enabling the transmission of reset commands to chips (indicated by RESET arrows).
- Extracting ADC values from the SCA and converting them into meaningful temperature readings.

This software functionality works together with the hardware components to help in data movement, chip configuration and initialization.

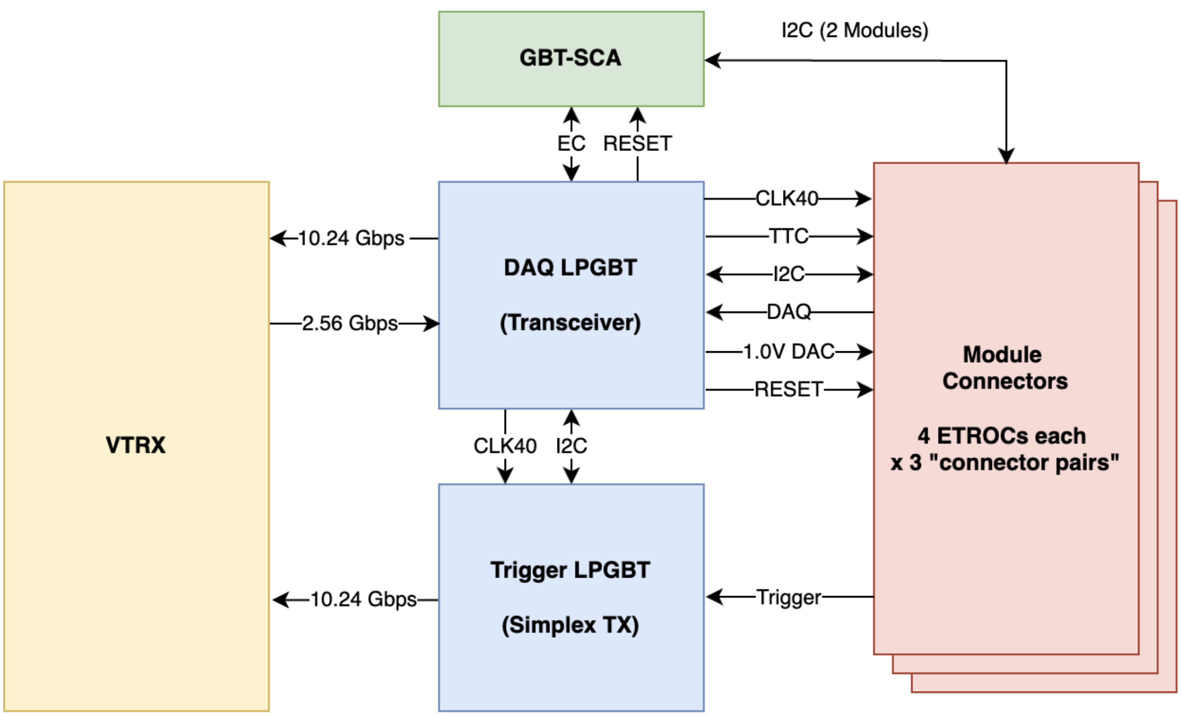

Figure 2: The Data Flow Diagram illustrating the interconnections between chips in the ETL modules and what role each chip is serving.

About My Project

During my internship, I had the privilege of actively working on the MTD-DAQ project, and specifically working on the ETL software part of it. This section highlights the three key areas of my involvement: Learning Efforts, Configuration Files, and Service Hybrid Initialization and Testing Scripts.

Learning Efforts

My journey began with a dedicated effort to comprehend the various parts of the MTD system and specifically the ETL and BTL subdetectors and the existing code-base of the BTL. Through a process of reading manuals, studying documentation, reviewing code, and asking my supervisors and other team members about the different parts of the system, I aimed to have a good understanding of the system. This endeavor included exploring technical resources such as the ETROC, LpGBT, and SCA manuals, and the MTD technical design report. This learning journey equipped me with the essential knowledge needed to contribute to the software for the ETL.

Configuration Files

For the chips on the service hybrid to work as expected, we must write some configuration files to ensure proper operation of the chips. An aspect of my contributions to the MTD-DAQ project involved creating configuration files. These files can be thought of as detailed instructions that guide each chip on how to perform its tasks effectively. This approach was influenced by the structure used for the BTL, aiming for compatibility and consistency across the system.

As you can see in Figure 3, the configuration is a tree-like structure: there's a main starting point called 'SH-config' which leads to specific paths labeled 'LpGBT0-config,' 'LpGBT1-config,' and 'SCA-config.' Each path is like a connection to an individual chip within the readout system.

Each chip has its own set of specific instructions. Additionally, each chip points us to important files like 'ADC' and 'GPIO mapping' that provides mapping information of the ADC and GPIO ports of the chips.

In collaboration with my supervisors, I created and verified configuration files for the service hybrid. This involved aligning the configuration structure with that of the BTL, ensuring compatibility and a consistent approach.

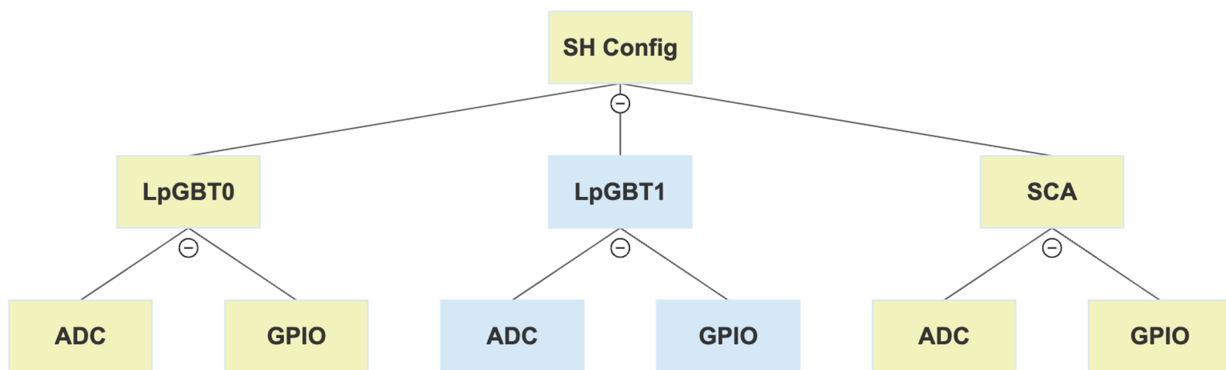

Figure 3: Part of the configuration structure for the ETL which is consistent with the BTL configuration structure.

Service Hybrid initialization and testing scripts

Another aspect of my contributions involved the development of scripts for the initialization and testing of the SH chips. These scripts ensure the proper start-up and functioning of the SH chips within the ETL readout system.

The initialization script is designed to properly configure each chip. This involves setting specific parameters and modes that enable the chips to communicate with the other chips on the service hybrid. Although we faced some difficulties in initializing the first chip, we managed eventually to initialize the DAQ-LpGBT. Furthermore, I worked on testing scripts that assess various functionalities of the DAQ-LpGBT chip of the Service Hybrid. One script focuses on reading ADC values. This is important to provide temperature information. Another script is dedicated to testing the Electronic Clock (Eclk) configurations. It ensures that the Eclk frequencies are correctly set. I used unittest, a built-in testing library, to write some of these scripts.

Conclusion

This report has explored the role of the ETL in the bigger MTD-DAQ system, showing how both hardware and software work together. It also sheds light on the various aspects of my work as an intern. We made configuration files and developed testing scripts to make sure things run correctly. Along with that, we made some progress in initializing the service hybrid DAQ-LpGBT chip and adapting the BTL code.

Even though we are a small part of the MTD-DAQ project, I believe that the project I was working on is meaningful and challenging which I highly enjoyed. I gained new insights in particle physics and software development through this great 8-week period.

References

[1] CMS Collaboration, "A MIP Timing Detector for the CMS Phase-2 Upgrade," CERN-LHCC-2019-003 ; CMS-TDR-020, ISBN: 978-92-9083-524-0, 2019. [Online]. Available: https://cds.cern.ch/record/2667167?ln=en

[2] P. Simkina and CMS Collaboration, "The DAQ and clock distribution system of CMS MIP Timing Detector," CMS-CR-2022-099, [Online]. Available: https://cds.cern.ch/record/2824604

[3] "EMP framework guide" CERN, [Online]. Available: https://serenity.web.cern.ch/emp-fwk/

[4] A. Rose, "Serenity - An ATCA prototyping platform for CMS Phase-2," CMS-CR-2018-327, Oct. 31, 2018. [Online]. Available: https://cds.cern.ch/record/2646388/files/CR2018_327.pdf