CERN Summer Student Programme 2023

18 August 2023
paulojunior367@gmail.com

Development of a software package for controlling Precision Proton Spectrometer (CMS-PPS/CERN)

Paulo Roberto de Moura Júnior

Supervisor:
D. Figueiredo - *Istituto Nazionale di Fisica Nucleare (Pisa)*

Keywords: Compact Muon Solenoid (CMS), Precision Proton Spectrometer (PPS), XDAQ and TOTEM.

Abstract

The Precision Proton Spectrometer (PPS) is a sub-detector of the Compact Muon Solenoid (CMS), made of a set of near-beam movable detectors, housed in Roman Pot (RP) units, symmetrically installed along the LHC beam pipe about 200m from the CMS Interaction Point (IP5). PPS is designed for precise measurements of the position, translated in momentum loss, and the time of flight of protons scattered in a very forward region. In light of that, the combination of the forward detector acceptances and precise time measurements O(-12)s are very powerful for gamma-gamma high masses processes searches. The goal of this summer student project was to develop a software package for controlling the frontend electronics of the micro Silicon-Strips detectors, used for PPS in special LHC fills for Roman Pots alignment. The software is written in two layers, one for hardware access and another as a top layer user interface. Through a web application users should be capable of accessing specific frontend boards by a slow-control token bit protocol communication, managed by the CERN-made CCU25 ASIC. Therefore, during the stay in the project, the software package was successfully developed meeting the initial requirements and tests proved the package to be working as expected.

Contents

1 Introduction

1.1 The Precision Proton Spectrometer

The Precision Proton Spectrometer (PPS) is a subsystem of the Compact Muon Solenoid (CMS) experiment, which has been designed for measuring scattered protons. Historically, PPS was originated as a joint project between CMS and TOTal Elastic and diffractive cross section Measurement (TOTEM)[1] Collaborations.

PPS is composed by a set of mechanical near-beam movable units, which host tracking[2] or timing[3] detectors and are sealed under a secondary vacuum in respect to the beam line. In the literature, this type of detectors are known as Roman-Pots (RPOTs). PPS RPOTs detectors are symmetrically installed in stations along the LHC beam pipe and localized at about 200 m from the CMS Interaction Point (IP5), as shown in Figure 1. PPS operates with different technologies for tracking (CMS 3D pixels and silicon micro-strips for beam alignment) used for protons kinematics measurements, and timing (artificial single-crystal diamonds - scCVD), for pileup rejection.

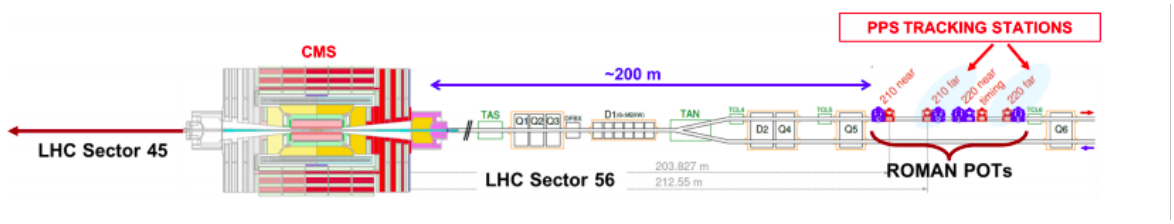

Figure 1: schematic diagram of one of the LHC sectors where PPS is installed. The PPS timing station is placed between two PPS tracking stations. See ref.[3].

The PPS detectors exploit some advantages of being installed at a very forward region due to its acceptance, such as:

- to measure high masses exotic states produced via gamma-gamma interactions, because the large suppression of quark-gluon (qg) and/or gluon-gluon (gg) cross-sections.
- to measure Anomalous Quartic Gauge Couplings (AQGCs) in Beyond Standard Model (BSM) Physics, increasing the sensitivity of the central detector.
- to measure emerging intact protons from colourless exchanges with small momentum loss ($\xi = \triangle p/p$ between 2 and 10%) and providing data for forward physics processes.
- to monitor the LHC beam optics, stability and to provide feedback for the LHC teams operations.

1.2 PPS and TOTEM Data Acquisition Systems (DAQ)

PPS readout is fully integrated with CMS for a common data-taking. For such purposes, a set of software tools have been developed for automatically control and configure the detector finite state machines (FSMs), based on a CMS online C++ framework, named XDAQ[4]. PPS DAQ downstream is composed by the so-called *partitions* which handle the interface between CMS and PPS communications: one for 3D pixels and another for the timing and legacy TOTEM micro silicon-strips (Si-Strips) detectors. At the hardware level, PPS (or TOTEM) sends triggered events upstream CMS FEROL readout units.

The PPS 3D pixels partition mimics the CMS pixels and it is based on μTCA standard electronics. At the back-end side, there is one board using the slow control (CCU25) token ring protocol to configure the integrated circuits of PPS portcard[1] and another back-end board to deliver the I^2C fast-commands at 400 MHz clock speed, which aims to configure the readout ASICS PROC600 and token-bit-manager (TBM10) in each plane, as well as to provide LHC clock and CMS trigger commands. Both back-end boards are exactly the same, named FC7, but they differ by the firmware uploaded on them. Finally, the readout from the detector is performed synchronizing triggered events which are sent by the *POH4s* optotransceivers mezzanines connected in the front-end board (PPS portcard). The events are later sent to CMS FEROL by the 10 Gb/s S-LINK opto-links.

The PPS timing partition, which also handles the readout of the Si-Strips for alignment or TOTEM runs[2], uses the CCU25 token ring to configure the field programmable gate array (FPGAs) soldered in a specific designed timing front-end board, named *Digitizer Board*. The Digitizer board[5][6] is an interface card hosting a Microsemi SmartFusion2 M2S150-FC1152 FPGA which codes 32-bit word packets and delivers data upstream CMS DAQ with 40 MHz clock synchronization. The I^2C lines of the FPGAs are distributing commands for the embedded devices, such as phase lock loop (QPLL) or Giga Optical Hybrids (GOH), while the JTAG lines are specifically used for the readout ASICs communication. Finally, for the back-end side, the slow control ring is interfacing a front-end Control (FEC) board based on the VME standards. Due its flexibility allowing customized mezzanines hosting different readout ASICs, the digitizer board has been used in PPS (and TOTEM) for different data-taking layouts.

The triggered events are received by a upstream PPS SRS (scalable readout system[3]), legacy from TOTEM Collaboration, delivered from the digitizer's board GOHs to CMS ferol using the 5Gb/s S-LINK copper links. The SRS control software sends UDP packets to prepare the PPS timing partition for the readout.

[1] Front-end board where the 6 detector planes are connected. Each plane is composed by a flex-hybrid board where the readout chips are wire-bounded together with the sensors.

[2] Or any other detector technology designed for TOTEM

[3] Based on RD51 Collaboration design.

2 Summer Student Activities

2.1 Si-Strips Control Software

The FSMs of the CMS subsystems are controlled by a scalable software framework, which is able to broadcast states for all the CMS subsystems using SOAP commands[7]. The XDAQ tool enables connectivity and communication between different applications layers distributed in all the CMS data acquisition online network.

The slow control is vastly used within the CMS collaboration since the beginning of Run-1, relying on the CCU25 radiation hardness communication ASIC that distributes a token ring for all the front-end boards that can be electrically chained in the same optical link.

The design of the control software is written with two layers. The first layer is the interface library between the hardware and the back-end. This library has the methods to create the slow control objects as well as the methods for reading or writing in specific registers, when provided the CCU25 address and I^2C channels. The second one is a top layer user interface that enables users to access the methods for each CCU25 through a web application.

At the LHC tunnel, each sector (45 and 56) has two pairs of top/bottom vertical Roman Pots Si-Strips detectors, controlled by a single slow control loop (CCU25). Thus one loop in sector 45 or 56 is controlling four CCU25 chips. The main goal of this project was to develop a XDAQ application which controls customized microelectronics devices (GOHs, PLLs) of the frontend board of the Roman Pot Si-Strips. The final package developed will have then two XDAQ instances running, one per sector.

The XDAQ software is loaded with a text file containing the addresses of the CCU chips (with ring and slot addresses as well) line by line to further perform PLL or GOH actions, such as reset, reading status or setting theirs internal registers.

2.2 Application development

All the work made in the project was performed using C++ language and the user interface is a web application implemented through GNU Cgicc library [8], which is a C++ Common Gateway Interface (CGI) class library that can be used to access the HTML elements, such as forms through the XDAQ instances. The use of CGI in the project is justified because it's a very straightforward method for implementing simple web-based tasks, such as form processing.

The distinguished feature of the developed application is the implementation of an algorithm to load the text file and to dynamically creates the hardware interface for different CCU25 chips. This enables the XDAQ software layer to be more flexible as the user can, at any time, change the addresses before running the application.

Once the application is running, it will create accesses to each CCU25 to interface all the methods implemented for the back and front-end. Then whenever the web interface is accessed, tabs will be created for each configured device, as shown in the Figure 2. In each created tab there will be forms to be filled with values for performing a desired action and submit buttons that run a specific method implemented in the XDAQ instances, such as reading or setting up specific registers on the board of interest.

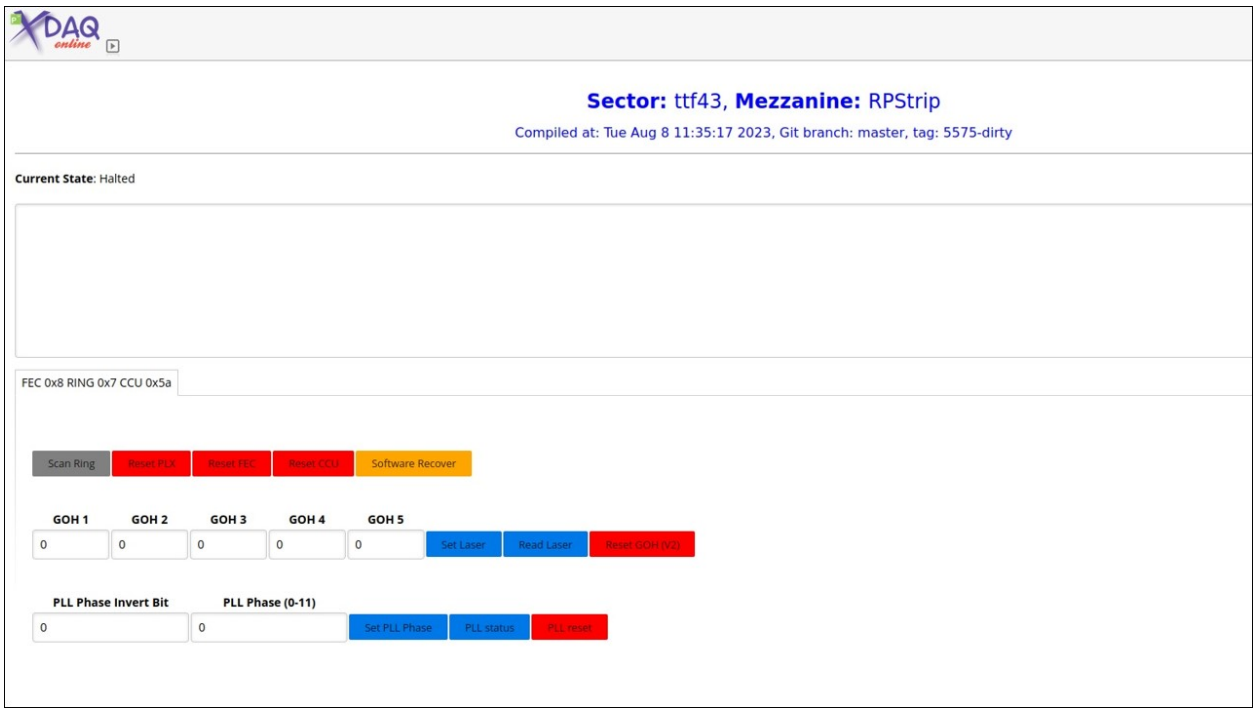

Figure 2: created tab for CCU25 device with addresses FEC 0x8 RING 0x7 and CCU 0x5a provided from text file.

2.3 Application testing

For verifying the proper functioning of the package, some basic tests were performed such as filling up all GOH and PLL values and trying to send them to the accessed device through "Set Laser" and "Set PLL Phase" buttons and testing all the other submit buttons, accessing the same CCU device in two different tabs to see if the buttons and methods would work as expected and trying to run the application without filling up or without creating the required text file to see if the protections implemented would work correctly. Finally, the software package worked properly in all those tests that have been performed. The application is robust against typo errors and hardware access issues.

3 Conclusion

As the main desired features for the software development were successfully implemented and the tests performed proved the package to be functional, the goal of this project of developing a control system to help the operations of the Si-Strips detectors was successfully achieved. There are still upgrades for the future of the project that haven't been performed because of the short stay in the project, such as the implementation of more tabs for checking the status and performing debug actions in the chips. These extra features are not essential but could be implemented to make the control system more complete.

4 Acknowledgements

I would like to thank every person that has made this journey possible for me especially my family who taught me to believe in my dreams and my former Brazilian supervisor Prof. Dr. Gustavo Gil da Silveira. I couldn't thank enough my supervisor Dr. Diego Figueiredo for being patient, always ready to help and very helpful during my stay in the project. At CERN Summer Student Programme I had the perfect opportunity to grow personally meeting different cultures and learning from them and also professionally because I had to learn completely new things such as C++, HTML and CGI programming.

References

[1] M. Albrow, M. Arneodo, V. Avati *et al.*, 'CMS-TOTEM Precision Proton Spectrometer,' Tech. Rep., 2014. [Online]. Available: https://cds.cern.ch/record/1753795.

[2] M. M. Obertino, 'The PPS tracking system: performance in LHC Run2 and prospects for LHC Run3,' *JINST*, vol. 15, C05049. 11 p, 2020. DOI: 10.1088/1748-0221/15/05/C05049. [Online]. Available: https://cds.cern.ch/record/2747949.

[3] E. Bossini, 'The CMS Precision Proton Spectrometer timing system: performance in Run 2, future upgrades and sensor radiation hardness studies,' *JINST*, vol. 15, C05054. 12 p, May 2020, Proceedings of IPRD2019 Conference. DOI: 10.1088/1748-0221/15/05/C05054. arXiv: 2004.11068. [Online]. Available: https://cds.cern.ch/record/2715995.

[4] V. Brigljevic *et al.*, 'Using XDAQ in application scenarios of the CMS experiment,' *eConf*, vol. C0303241, MOGT008, 2003. arXiv: hep-ex/0305076.

[5] TOTEM Collaboration, 'Upgrade of the TOTEM T2 Telescope, Technical Design Report,' CERN, Geneva, Tech. Rep., Jun. 2019. [Online]. Available: https://cds.cern.ch/record/2677468.

[6] E. Bossini, 'The Proton Timing System of the TOTEM experiment at LHC,' *PoS*, vol. TWEPP2018, 137. 5 p, 2019. DOI: 10.22323/1.343.0137. [Online]. Available: https://cds.cern.ch/record/2710220.

[7] *XDAQ Soap messaging*. [Online]. Available: https://twiki.cern.ch/twiki/bin/view/XdaqWiki/SOAPMessaging.

[8] *GNU cgicc library*. [Online]. Available: https://www.gnu.org/software/cgicc/.