



Development of online software tools for Precision Proton Spectrometer (CMS-PPS/CERN)

Nadia Mariana Leal Reyes

Supervisors:

D. Figueiredo - *Istituto Nazionale di Fisica Nucleare, Sezione di Pisa*

I. Pedraza - *Benemérita Universidad Autónoma de Puebla*

Keywords: Compact Muon Solenoid (CMS), Precision Proton Spectrometer (PPS) and XDAQ
CMS Online software

Abstract

The Precision Proton Spectrometer (PPS) is a subdetector 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 $\mathcal{O}(-12)$ s are very powerful for gamma-gamma high masses processes searches.

This summer student project focus on the PPS data acquisition system (DAQ) and online software. PPS DAQ is fully integrated with the CMS DAQ online environment, which relies on software tools for communication to configure the detector Final State Machines (FSM). These FSMs are mostly detector related to prepare each CMS subsystem for the readout in an automatized process and they are controlled by a scalable software framework, referred as XDAQ.

Finally, the development of a new online software utility, written in an object-oriented programming language, which controls the PPS detectors through a slow control optical link will be presented.

Contents

1	Introduction	3
2	Electronics and Online System	4
2.1	Hardware	4
2.2	Setup for Testing and Validation	5
2.3	Control Software	5
2.4	Virtual Machine Configuration	6
2.5	Application	6
2.6	Remote Access Configuration	7
2.7	Software Repository	7
3	Conclusion	7
4	Acknowledgements	7

1 Introduction

The Precision Proton Spectrometer (PPS) is a sub-system 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 a set of mechanical near-beam movable units, which house tracking[2] or timing[3] detectors sealed under a secondary vacuum in respect to the LHC primary vacuum. In the literature, this type of detectors are known as Roman-Pots (RPs). PPS RP detectors are symmetrically installed in stations along the LHC beam pipe at about 200 m from the CMS Interaction Point (IP5), as shown in Figure 1. In LHC Run-2, PPS operated for the first time commissioned with different technologies for tracking (CMS 3D pixels and micro-strips) used for protons kinematics measurements, and timing (artificial single-crystal diamonds - scCVD and Ultra Fast Silicon Detectors - UFSD), for pile-up rejection.

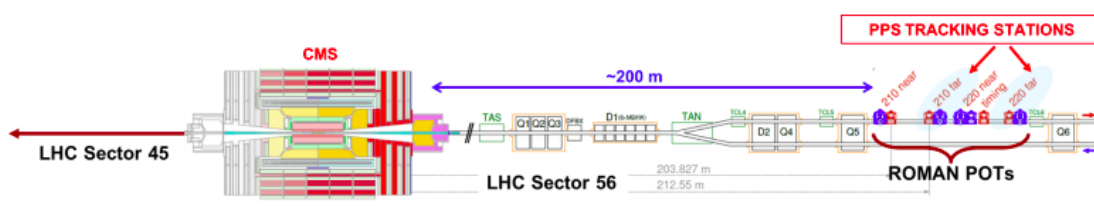


Figure 1: Schematic diagram of one of the LHC sectors where PPS is installed. The timing station is placed between two tracking stations. See ref.[3].

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 ($\xi = \Delta p/p$ 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.

Finally, PPS readout is fully integrated with CMS for a common data-taking. For such purposes, a set of software tools have been developed for control, configuring and preparing the detector final state machines (FSMs), based on a CMS online C++ framework, named XDAQ[4].

2 Electronics and Online System

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[5]. The XDAQ tool enables connectivity and communication between different applications layers distributed in all the CMS data acquisition online network.

The aim of this project is the development of a new online software utility, written in an object-oriented programming language and able to control a new design mezzanine (new T2) for the PPS (TOTEM) front-end board through a slow control optical link [6]. 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.

2.1 Hardware

Part of the PPS DAQ sub-system uses the CCU25 token ring to configure the field programmable gate array (FPGAs) soldered in a specific designed front-end board, named *Digitizer Board*, Figure 2. The Digitizer board[7][8] 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 I²C 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 read-out 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 read-out ASICs, the digitizer board has been used in PPS (and TOTEM) for different data-taking layouts.

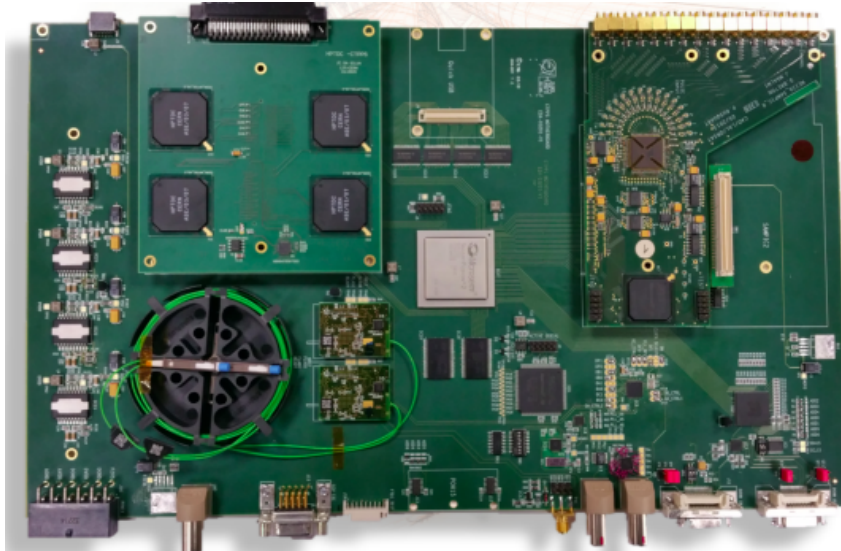


Figure 2: Digitizer board hosting a Microsemi SmartFusion2 FPGA in the center, two mezzanine slots, GOH and POH connectors, CCU25 ASIC, LHC clock and trigger inputs. The system features a remote FPGA firmware upgrade by optical links (Credit: G. Antchev).

2.2 Setup for Testing and Validation

In the scope of this project, Figure 3 is showing a scheme of the setup mounted at CERN for the new T2 mezzanine software development: a computer is interfacing the VME crate controller, where the FEC board is placed. From the FEC board mezzanine (i.e. ring 1 to 12) an optical link is connected to an optical to electrical converter (DOHM), which can interconnect in a control loop multiple front-end boards. The computer is interfacing the VME controller (Optical Link Bridge, V3718) by a PCI CONET board (A2818).

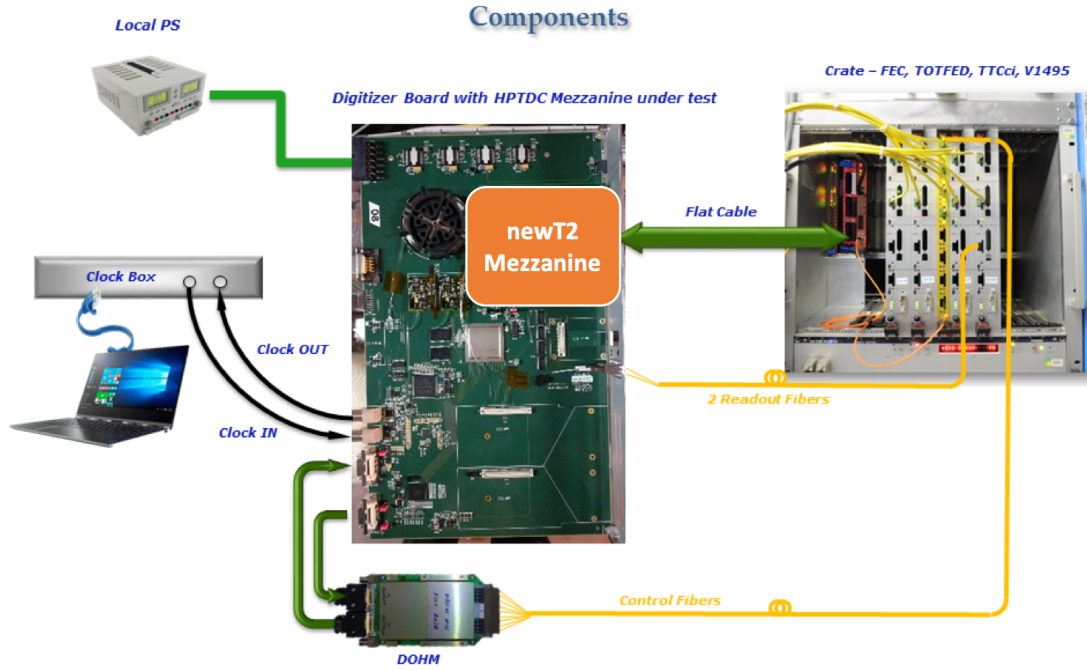


Figure 3: Scheme of the batch setup for the new T2 software development. The digitizer board is hosting the new T2 mezzanine. The digitizer board is controlled through the slow control (an optical link between the FEC VME backend and the DOHM). The commands are sent by a computer to the VME crate controller (Credit: G. Antchev).

2.3 Control Software

The design of the control software is based on 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 I²C channels. Furthermore, in that layer the sequence algorithm for configuring specific embedded devices (readout ASICs, QPLL, GOH etc.) are hard-coded. The second top layer is made to translate the CMS DAQ commands into FSM changes in the detector. In the same layer, there is a command line used for debugging and testing the hardware library methods, as shown in Figure 4.

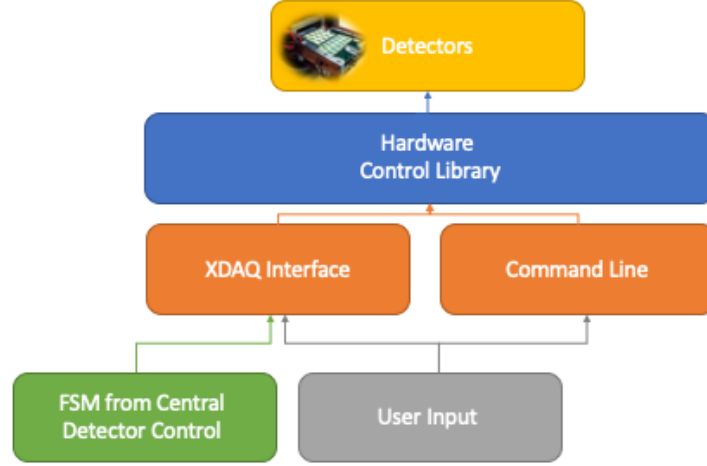


Figure 4: Scheme of the PPS control software: a user can send direct commands to the hardware using buttons (XDAQ) or thought command line. In addition, the software receives SOAP commands from the CMS DAQ shifter. Another layer of the software, is the interface between the computer and the detector (hardware library) using the slow control protocol.

2.4 Virtual Machine Configuration

The first approach for this project was based on the configuration of a virtual machine (VM), in which all the XDAQ dependencies libraries and packages were installed on Cent OS 7 unix system. The new T2 software[9] mimics a validated control software used during run-2 operations. The configured VM has been accessible through a terminal, by a local CERN network connection¹.

2.5 Application

The main task performed for the project was focused on programming in C++ language. Mainly coding with the use of a parsing method for commands and instructions made for controlling the hardware components. The application developed allows parsing parameters by the command line, as well as parsing parameters from a JSON (JavaScript Object Notation) file. At first glance, this application uses high-level predefined hardware sequences for a given FSM action, i.e. configuring embedded digitizer board devices. Finally, features to retrieve feedback from the digitizer board were added. Once the command line was validated and tested, the same methods were migrated for the XDAQ control software. As mentioned before, the XDAQ framework is used for communication between central CMS DAQ and the subsystems partitions, or the subsystem's local DAQ. Figure 5 is showing the new T2 XDAQ user interface which is accessible by an internet browser.

¹Secure Shell (SSH) connection to lxplus.cern.ch

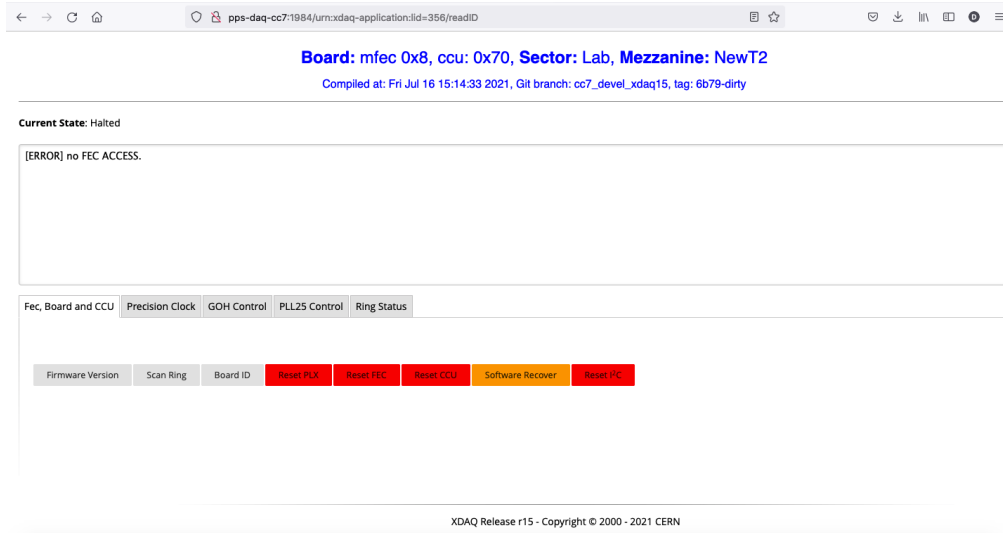


Figure 5: new T2 XDAQ interface running in a VM located within CERN network. This application is accessible by an internet browser. In order to open it outside CERN network, a SSH tunnel is needed.

2.6 Remote Access Configuration

A Secure Shell (SSH) tunnel was configured to access from outside CERN general network (GN). When running the XDAQ on a local machine another terminal is needed with a set of commands and has to be kept open all the time, if closed the network is lost. Moreover, the browser also needs to be configured using the port number defined for the *localhost* (or local machine).

2.7 Software Repository

The software repository for the project can be found on GitLab[9].

3 Conclusion

Implementation of control software based on the CMS XDAQ framework represent an advantage as it optimizes many tasks. Also, being everything online the software can be developed, maintained, tested and improved from anywhere in the world which allows to have better collaborations and updates.

The development of the software tools was not an easy task, specially having no actual experience in this field, but a lot of knowledge was gained and the goal was achieved.

4 Acknowledgements

I am really thankful for been given the opportunity to take part in the CERN Summer Student Programme as it was an invaluable experience. I would like to thank every person

that has made the programme possible. It made me grow both as an applied physicist and as a person. Lastly, I can't thank enough my supervisor Diego Figueiredo for being so patient, always willing to answer any questions and helping me overcome the obstacles. As well as Isabel Pedraza, who has been my mentor for a few years now and also supported me throughout this project.

References

- [1] M. Albrow, M. Arneodo, V. Avati, J. Baechler, N. Cartiglia, M. Deile, M. Gallinaro, J. Hollar, M. Lo Vetere, K. Oesterberg, N. Turini, J. Varela, D. Wright and C. CMS-TOTEM, "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](https://doi.org/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, 2020.
- [4] V. Brigljevic *et al.*, "Using XDAQ in application scenarios of the CMS experiment," *eConf*, vol. C0303241, MOGT008, 2003. arXiv: [hep-ex/0305076](https://arxiv.org/abs/hep-ex/0305076).
- [5] *XDAQ Soap messaging*. [Online]. Available: <https://twiki.cern.ch/twiki/bin/view/XdaqWiki/SOAPMessaging>.
- [6] C. Paillard, C. Ljuslin and A. Marchioro, "The CCU25: a network oriented communication and control unit integrated circuit in a 0.25 μm CMOS technology," 2002. DOI: [10.5170/CERN-2002-003.174](https://doi.org/10.5170/CERN-2002-003.174). [Online]. Available: <https://cds.cern.ch/record/593914>.
- [7] 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>.
- [8] 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](https://doi.org/10.22323/1.343.0137). [Online]. Available: <https://cds.cern.ch/record/2710220>.
- [9] *PPS Summer student repository*. [Online]. Available: <https://gitlab.cern.ch/pps-summer-students/controlsoftware>.