DEVELOPMENT OF THE SLOW CONTROLLER OF THE RPC SYSTEM LINK FOR LS2 UPDATE OF THE CMS/HL-LHC EXPERIMENT

Fabio Marujo da Silva

Tese apresentada ao Programa de Pós-Graduação em Instrumentação e Óptica Aplicada, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, e Universidade Federal Fluminense, UFF, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica, Área de concentração em Instrumentação.

Orientador: Gilvan Augusto Alves
Coorientadora: Ana Lucia Ferreira de Barros

Rio de Janeiro,

Agosto 2023

DEVELOPMENT OF THE SLOW CONTROLLER OF THE RPC SYSTEM LINK FOR LS2 UPDATE OF THE CMS/HL-LHC EXPERIMENT

Tese apresentada ao Programa de Pós-Graduação em Instrumentação e Óptica Aplicada, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, e Universidade Federal Fluminense, UFF, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica, Área de concentração em Instrumentação.

Fabio Marujo da Silva

Banca Examinadora:

Presidente, Gilvan Augusto Alves, D.Sc. (CBPF) (Orientador)

Professora Ana Lucia Ferreira de Barros, D.Sc. (CEFET/RJ) (Coorientadora)

Professor Daduí Cordeiro Guerrieri, Ph.D. (CEFET/RJ)

Professor Diego Barreto Haddad, D.Sc. (CEFET/RJ)

Professor Douglas Mota Dias, D.Sc. (UERJ)

Professor Robert Mota Oliveira, D.Sc. (UERJ)

Rio de Janeiro,

Agosto 2023

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

S568 Silva, Fabio Marujo da
Development of the slow controller of the RPC system link for LS2 update of the CMS/HL-LHC experimente / Fabio Marujo da Silva. — 2023.
122f. : il. color. , enc.

Tese (Doutorado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2023.
Bibliografia : f. 118-122
Orientador: Gilvan Augusto Alves
Coorientador: Ricardo Alexandre Amar Aguiar

1. Sistemas eletrônicos. 2. Detectores. 3. Eletrônica digital. 4. Sistemas embarcados (Computadores). I. Alves, Gilvan Augusto. (Orient.). II. Barros, Ana Lucia Ferreira de (Coorient.). III. Título.

CDD 621.381

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

DEDICATION

A Deus, por ter me enviado forças quando me ajoelhei de coração e pedi sabedoria e fé nessa trajetória ousada e obscura do conhecimento.

A Santo Antônio de Sant'Ana Galvão, (Frei Galvão) por escutar e atender minhas orações.

Aos meus Pais Orlando Pereira da Silva e Iara Marujo da Silva.

A Michelle Zolet, que foi mais forte e me ergueu nos momentos de derrota ofertando apoio quando mais precisava.

A todas as pessoas que conviveram comigo durante este período, rindo, chorando, estudando, divertindo, vivendo e dividindo todos esses momentos, que somente com estas pessoas foi possível superar as derrotas e dividir esta vitória.

ACKNOWLEDGMENTS

Agradeço incansavelmente a Deus por sempre me levantar dos tombos e me guiar pelo caminho correto na hora certa.

Ao meu orientador Gilvan Augusto Alves onde sempre sabia me direcionar, quando não havia solução. A minha coorientadora, Ana Lucia Ferreira de Barros que mesmo entre suas inúmeras atividades, conseguia ajudar a qualquer momento. Sem vocês não haveria este trabalho e foram essenciais para sua conclusão.

A todos os professores do PPGIO-CEFET/RJ, que passaram pela minha jornada acadêmica, sempre buscando ajudar no crescimento da minha vida profissional e a Secretaria da Pós-graduação, em especial a Sheila e o Ivan Luis Galloulckydio.

A toda comunidade científica do CBPF, que sempre torceu por meu sucesso, principalmente Prof. Nami Fux e Ignácio Bediaga.

Agradecimento em especial ao ex-diretor do CBPF Ronald Cintra Shellard (*in memorian*) e o Diretor Márcio Portes de Albuquerque, que confiaram no meu trabalho e me deram a oportunidade e suporte em colaborações internacionais, como o CERN e LIP onde evolui minha capacidade de pesquisa.

A coordenação da COTEC/CBPF por toda infraestrutura, em especial o Coordenador Marcelo Portes e o Mestre Gabriel Azzi que sempre me apoiaram incondicionalmente para o sucesso desta Tese. Aos meu amigos de trabalho Pedro Russano e Mauricio Bochner pela compreensão nessa fase de especialização.

Ao Grupo do CMS da UERJ, Coordenador Luiz Mundim, Professores Dilson de Jesus Damião, Eliza Melo da Costa, Felipe Torres da Silva Araujo e em especial ao Prof. Sandro Fonseca de Souza pelas longas conversas requintadas com um toque especial em assuntos da colaboração do RPC Muon Group.

Ao Prof. Antonio Vilela Pereira da UERJ, por dividir seu conhecimento e contribuir de forma exuberante para o bem dessa Tese.

Aos meus incontáveis amigos do passado e presente, mas em especial ao Luís Miguel Domingues Mendes, Denys Pestana Viana (2005/2 UERJ), Carlos Alberto Moraes Correia e Fabio Oliveira. Vocês também são responsáveis por ter deixado mais alegre essa caminhada até aqui, e um pedaço deste trabalho pertence a vocês.

EPIGRAPH

“Nada neste mundo pode substituir a persistência. O talento não pode; nada é mais comum do que homens talentosos e fracassados. A genialidade não pode; gênios não recompensados estão em todo lugar. A educação não pode; o mundo está cheio de gente instruída que não chegou a lugar algum. Persistência e determinação, apenas elas, são onipotentes”. -*Calvin Coolidge, Presidente Americano*

RESUMO

Desenvolvimento do Slow Controller do Sistema RPC Link para o Upgrade no LS2 do Experimento CMS/HL-LHC

A atualização da Fase II do Sistema de Link RPC visa atender a todos os requisitos do HL-LHC, incluindo a capacidade de operar em um ambiente de alta radiação, aprimorar a resolução de tempo e aumentar a capacidade de taxa de entrada e a largura de banda de saída do sistema. Além disso, o novo sistema de link é gerenciado pelo design exclusivo do Slow Controller. O Conceito de Slow Controller que foi desenvolvido neste trabalho refere-se a um Subsistema de Controle e Monitoramento. Definido por seu design intrincado e orientação não comercial, essa entidade computacional tem o papel crítico de evitar falhas, supervisionar operações e facilitar comandos manuais e automatizados para o novo RPC Link System. O Slow Controller integra perfeitamente sistemas e subsistemas entre o software RPC Online até o firmware dos FPGAs. Suportado por uma combinação de firmware, hardware e software, atende a uma ampla gama de aplicações internas e externas, utilizando uma combinação de equipamentos comerciais e não comerciais para controlar tarefas críticas, como distribuição do relógio TTC e comandos de acionamento rápido, configuração de limiares FEB, reconfigurando FPGAs do sistema Link e lendo o status do sistema Link. Neste desenvolvimento, todas essas funções são realizadas através de FPGA na rotina de implementação por software. A fase inicial, desenvolvida nesta tese, que é o sistema de pequena escala, envolveu uma análise aprofundada das funções exigidas pelo novo sistema Link e a melhor solução para o hardware proposto. Os resultados da implementação do Slow Controller foram traduzidas em subprogramas e firmware que são correspondente e integradas aos FPGAs por software em alto nível para construir a comunicação de dados para monitorar e controlar o novo RPC Link System. Paralelamente, foram estudadas as rotinas de software necessárias para o programa RPC online, e uma extensa avaliação é descrita e conduzida para garantir a funcionalidade perfeita da cadeia de controle e comunicação deste subsistema.

Palavras-chave: CMS; FPGA; RPC; Slow Controller; Sistemas Digitais

ABSTRACT

Development of the Slow Controller of the RPC System Link for LS2 Update of the CMS/HL-LHC Experiment

The RPC Link System, Phase II upgrade, aims to meet all HL-LHC requirements, including operating in a high radiation environment, improving time resolution, and increasing input rate capability and bandwidth system output band. Furthermore, the unique Slow Controller design manages the New RPC Link System. The Slow Controller concept that was developed in this work refers to a control and monitoring subsystem. Defined by its intricate design and non-commercial orientation, this computational entity has the critical role of preventing failures, overseeing operations, and facilitating manual and automated commands for the New RPC Link System. The Slow Controller seamlessly integrates systems and subsystems between RPC Online Software and FPGA firmware. Supported by a combination of firmware, hardware, and software, it serves a wide range of indoor and outdoor applications, utilizing a combination of commercial and non-commercial equipment to control critical tasks such as TTC clock distribution and quick trigger commands, setting FEB Thresholds, reconfiguring Link system FPGAs, and reading Link System status. All these functions are performed through FPGA in the software implementation routine in this development. The initial phase, developed in this thesis, is the small-scale system, which involves an in-depth analysis of the functions required by the New RPC Link System and the best solution for the proposed hardware. The results of the Slow Controller implementation were translated into subprograms and firmware that are matched and integrated into the FPGAs by software at a high level to build data communication to monitor and control the new RPC Link System. At the same time, the software routines for the Online RPC Program were studied, and an extensive evaluation was described and conducted to guarantee the perfect functionality of this subsystem's control and communication chain.

Keywords: CMS; FPGA; RPC; Slow Controller; Digital Systems

LIST OF FIGURES

Figure 1 – Illustration of developing thesis timeline context. 23

Figure 2 – Hierarchy of a control system in the A Toroidal LHC ApparatuS (ATLAS) experiment. Adapted from (POY et al., 2008). 25

Figure 3 – DCS architecture in the ATLAS experiment. Adapted from (AAD et al., 2008). 26

Figure 4 – Large Hadron Collider (LHC) diagram and breakdown of main detectors. The two beams and their collision points are also visible with highlights in CMS Experiment. Adapted from (EVANS; BRYANT, 2008). 27

Figure 5 – Pre-pandemic calendar of the different periods of LHC (THE LHC EXPERIMENTS COMMITTEE, 2017). Adapted from (EVANS; BRYANT, 2008) 29

Figure 6 – FEE Printed Circuit Board. 30

Figure 7 – Brazilian Collaboration in the CMS/LHC Hadronic Calorimeter Project in 2014. 31

Figure 8 – Hardware Interface Units and Communication channels like the new Slow Controller system. Adapted from (THE LHC EXPERIMENTS COMMITTEE, 2017). 32

Figure 9 – The subsystems that Slow Controller (in light blue) acts in the Resistive Plate Chamber (RPC) Detector. 34

Figure 10 – The 3 phases foreseen for the development of Slow Controller in the European Organization for Nuclear Research (CERN) Project. Adapted from (LARREA et al., 2015). 35

Figure 11 – Initial setup prototype of the Slow Controller in the laboratory. 36

Figure 12 – Phase 2 of the Slow Controller in the laboratory with 13th AMC in MCH-2 slot (AMC13) connected at Control Board-V1 testing the monitoring and control functionalities. 37

Figure 13 – In the quadrant of the CMS experiment, the red selection indicates the region where additional RPCs will be placed to extend the muon coverage, and the purple outline indicates the region interested in updating the RPC Link System. 39

Figure 14 – The schematic sectional view of the Compact Muon Solenoid (CMS) detector with the geometric and subsection components. Looking outside the CMS at the central interaction point, we have the Silicon Tracker, the Electromagnetic Calorimeter, the Hadronic Calorimeter, the Superconducting Solenoid, and the Muon Chambers. 40

Figure 15 – Transmission to back-end and AMC13 for data acquisition to central CMS with emphasis on control (Slow Controller) in micro Telecommunications Computing Architecture (uTCA). Adapted from (COSTANZA et al., 2017). 45

Figure 16 – Block diagram of slow controller implemented with Low-Power GigaBit Transceiver (lpGBT)s. Adapted from (ANDERSON; FREEMAN; WHITMORE, 2012). 46

Figure 17 – Block diagram of slow controller implemented with lpGBTs. Adapted from (DENG et al., 2020). 48

Figure 18 – Schematic view of RPC chambers double gap design (left). Schematic view of readout and connections to and from link and control boards (right). The numbers refer to the complete RPC system after the update. 49

Figure 19 – On the left side, the current Control Board (CB)'s and Link Board (LB)'s installed in the VME Rack of the 3RE region of the barrel, and on the right side, the connection scheme of the RPC Link System. 52

Figure 20 – Assembly flow of design requirements and components of the Slow Controller system. 55

Figure 21 – Description of the Slow Controller in control between the high-level software and the hardware of the CBs and LBs for the new RPC Link System. 57

Figure 22 – Hardware configuration for the Slow Controller with the AMC13XG advanced module. 59
Figure 23 – Scheme of the Slow Controller: a user can send direct commands to the hardware using toolbox XDAQ or the command line. Adapted from (LEAL REYES, 2021). 60
Figure 24 – Control board architecture for VME pattern modified. 62
Figure 25 – The Architecture of bits in use at the Control Board for Slow Controller. 62
Figure 26 – All data frame formats with 120-bit to access by a Slow Controller System using XDAQ. 63
Figure 27 – Commands Type-A for reading/set for Control Board, Link Board and FEB parameters. 64
Figure 28 – Type-B commands for selecting LB and reading diagnostic data. 65
Figure 29 – Type-C commands in GBT-frame seen in fixed latency stream for remote programming on FPGA. 66
Figure 30 – VadaTech's uTCA Crate model VT892. 66
Figure 31 – Settings for AMC and physical ports at modules uTCA. 67
Figure 32 – The uTCA dual-star backplane is connected by MCH and MCH2 bus. 69
Figure 33 – Default NAT-MCH BASE using in Slow Controller with NAT-MCH-User's view Panel. 70
Figure 34 – The NAT-MCH module functions as a router for the IP header datagram. 71
Figure 35 – The transaction or set of transactions must fit into a single UDP packet in a TCP-IP datagram. 72
Figure 36 – A bus timeout of approximately eight microseconds in ACK until the end request. 73
Figure 37 – AMC13XG block diagram with Board Stack configuration only tongues 1, 2, and 3. 75
Figure 38 – Block diagram of the AMC13XG operating as a central data hub on the Slow Controller. 76
Figure 39 – The block diagram for the CentOS 7 Linux control internal process. 78

Figure 40 – A HyperDAQ server for monitoring simple quantity-specific of both events inside the application. Adapted from (HOLZNER, 2009). 79
Figure 41 – Configuration file XML for connecting AMC13XG in PC host by TCP-IP. 80
Figure 42 – Structure of the hierarchy tree of the slow control environment algorithm from geographical and hardware points of view. 82
Figure 43 – Software Selection for a Server Installation in CERN CentOS 7. 85
Figure 44 – Default slot configuration using MCH1 and MCH2 for starting network layout. 86
Figure 45 – Access by USB port to the discovery of IP default NAT-MCH. 87
Figure 46 – Access uTCA by IP default NAT-MCH. 88
Figure 47 – Connection successful on AMC13XG runs in terminal Linux. 95
Figure 48 – Differences between Data-frames assembly by firmware. 105
Figure 49 – Hardware status in level 9 in a halted state. 107
Figure 51 – Status generated by Slow Controller code at level 9 using the AMC13XG driver. 109
Figure 52 – Data taken from the acquisition by the Slow Controller containing the addresses and values of the registers stored in the FPGA T1. 110
Figure 53 – Register by Dump 1 from FPGA T1 Registers. 110
Figure 54 – Hyperdaq runs in PC Host for Slow Controller. 112
Figure 55 – File XML from AMC13XG configuration for FPGA T1. 113

LIST OF TABLES

Table 1 – Information on CMS components and their purposes. Modified from (DOOLING, 2015). 42

Table 2 – Component numbers of the current RPC link system. 51

Table 3 – Firmware available to AM13XG: 366 versions. 104

LIST OF ALGORITHMS

Algorithm 1 –	Basic experimental to start control system.	97
Algorithm 2 –	The command insertion process optimize.	99
Algorithm 3 –	Algorithm by data input on SFP1.	100
Algorithm 4 –	Python code to implement basic control system.	101

LIST OF LISTINGS

Listing 1 – Command to show Board configuration from NAT-MCH. 86
Listing 2 – Command to remove old versions. 88
Listing 3 – Installing new repositories to IPBus package. 88
Listing 4 – Command to instal ControlHub. 89
Listing 5 – Command to update the environment variables. 89
Listing 6 – The file /etc/yum.repos.d/cactus.repo on CentOS 7. 90
Listing 7 – Install CACTUS and AMC13 Tools on CentOS 7. 91
Listing 8 – Installing Prerequisits from GCC8 on Centos 7. 92
Listing 9 – Installation a devtoolset-8 on CentOS 7 package with repositories. 92
Listing 10 – Command to update the environment variables. 92
Listing 11 – Configuration IP in xml file for AMC13XG. 93
Listing 12 – Set the environment folder path for the links from XML tables. 94
Listing 13 – Start ControlHub service in CentOS 7. 95
Listing 14 – Set the environment variables the links from xml tables. 95
Listing 15 – General reset and counter reset commands in AMC13XG. 96
Listing 16 – The AMC13 generates fake data signals and enable Backplane Links. 97
Listing 17 – From Event Builder enables DAQLSC to output Fiber. 98

LIST OF ABBREVIATIONS

ACK	Acknowledgment
ADC	Analogic Digital Converter
ALICE	A Large Ion Collider Experiment
AMC	Advanced Mezzanine Card
AMC13	13Th AMC In MCH-2 Slot
ASIC	Application Specific Integrated Circuit
ATCA	Advanced Telecommunications Computing Architecture
ATLAS	A Toroidal LHC ApparatuS
BSM	Beyond The Standard Model
BTI	Bunch And Track Identifier
BX	Bunch Crossing
CB	Control Board
CBPF	Centro Brasileiro De Pesquisas Físicas
CCM	Crate Control Module
CCS	Clock & Control System
CCU	Communication And Control Units
CEFET/RJ	Centro Federal De Educação Tecnológica Celso Suckow Da Fonseca
CERN	European Organization For Nuclear Research
CLK1	Port FCLKA In MCH2
CMS	Compact Muon Solenoid
CMSOS	CMS Online Software
CNPQ	Conselho Nacional De Desenvolvimento Científico E Tecnológico
CPPF	Concentrator And Pre-Processor Fanout
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSC	Cathode Strip Chambers
DAC	Digital-to-analog Converter
DAQ	Data Acquisition Controller
DAQLSC	DAQ Link For Readout Over An Optical Fiber

DCS	Detector Control System
DHCP	Dynamic Host Configuration Protocol
DIO	Digital Input/Output
DSPS	Digital Signal Processors
DT	Drift Tubes
FC7	Fast Control Board With FPGA Xilinx Kintex-7
FEB	Front-End Electronic Board
FEC	Front-End Controller
FEE	Front-End Electronic
FIFO	First In, First Out
FINEP	Financiadora De Estudos E Projetos
FPGA	Field Programmable Gate Array
GB/S	Gigabyte Per Second
GBE	Gigabit Ethernet
GBP/S	Gigabits Per Second
GBT	Protocol High Speed Bidirectional ASIC
GBTX	Radiation Tolerant GBT Protocol High Speed Bidirectional ASIC
GCC	GNU Compiler Collection
GEV	Giga-electron-volt
GNU	GNU Project's Implementation Of The C Standard Library
GOL	Gigabit Optical Link
HCAL	Hadron Calorimeter
HEP	High Energy Physics
HF	Frontal Hadronic Calorimeter
HL-LHC	High Luminosity Large Hadron Collider
HPL	High Pressure Laminate
HTTP/2	Hypertext Transfer Protocol Version 2
I/O	Input/output
ID	Hardware Identification
IP	Internet Protocol
IPBUS	Protocol To Control/communicate With Ethernet-attached UTCA Hardware
IPC	Inter-process Communication
IPM	Institute For Research In Fundamental Science

IRPC	Improved RPC
ISO	International Organization For Standardization
JTAG	Joint Test Action Group
LB	Link Board
LBB	Link Board Box
LHC	Large Hadron Collider
LHCB	Large Hadron Collider Beauty
LPGBT	Low-Power GigaBit Transceiver
LS	Long Shutdown
LS2	Long Shutdown 2
LS3	Long Shutdown 3
LVDS	Low-Voltage Differential Signaling
MB/S	Megabyte Per Second
MCH	MicroTCA Carrier Hub
MCH1	First MicroTCA Carrier Hub
MCH2	Second MicroTCA Carrier Hub
MHZ	Megahertz
NAT	Network Address Translator
NAT-MCH	Double-width MicroTCA Carrier Hub
NGFEC	Next Generation Front-End Controller
NS	Nanosecond
OMTF	Overlap Region (Barrel And Endcap) Muon Track Finder
PCB	Printed Circuit Board
PPGIO	Programa De Pós-graduação Em Instrumentação E Óptica Aplicada
QPLL	Quartz Phase Locked Loop
QSPI	Quad Serial Peripheral Interface
RCMS	Run Control And Monitor System
RPC	Resistive Plate Chamber
SDRAM	Synchronous Dynamic Random Access Memory
SEU	Single Event Upset
SFP+	Small Form-Factor Pluggable Plus
SIPMS	Silicon PMTs
SM	Standard Model

SN	Serial Number
SRAM	Static Random Access Memory
TCDS	CMS Timing And Control Distribution System
TCP	Transmission Control Protocol
TDC	Time-to-Digital Converter
TDR	Technical Design Report
TEV	Tera-electron-volt
TRACO	TRAck COrrelator
TTC	Timing And Trigger Control
TTCRX	TTC Receiver ASIC
TTS	Trigger Throttling System
TWINMUX	Twin Multiplexer, Data Concentrator For Muon Barrel Track Finder
UART	Universal Asynchronous Receiver/Transmitter
UHAL	Hardware Access Layer For UTCA
UHTR	UTCA Trigger And Readout Module
USB	Universal Serial Bus
USC	Service Cavern
UTCA	Micro Telecommunications Computing Architecture
VLSI	Very-Large-Scale Integration
VME	Versa Module Eurocard Bus, ANSI/IEEE Computer Hardware Bus Standard
XAAS	XDAQ As A Service Software
XDAQ	Generic Data Acquisition Software Environment
XML	EXtensible Markup Language

CONTENTS

Introduction		**23**
Text Structure and Organization		23
Slow Controller Concept		24
The LHC and the CMS Experiment		26
The LHC update		28
Brazil Collaboration on CMS		29
Motivation		30
General Objectives		33
Specific Objective		35
Justification		37
1	**Bibliographic Revision**	**40**
1.1	The Slow Controller in Large Experiments	40
1.1.1	CMS detector at the LHC	40
1.1.1.1	CMS Sub Detectors	42
1.1.2	The Slow Controller in the CMS HCAL	44
1.1.3	The Slow Controller in ATLAS TileCal	47
1.2	The current RPC Muon Detector System	49
1.2.1	The Current RPC Link System	51
2	**Instrumentation and Methods**	**54**
2.1	Introduction	54
2.2	Design Requirements and Components	54
2.3	System Description	57
2.4	Instrumentation Methodology	58
2.5	Materials of Systems and Subsystems	59
2.5.1	The Control Board System	60

2.5.1.1 Overview of the Link System Project 60
2.5.1.2 Architecture 61
2.5.1.3 Command Types 63
2.5.2 The Hardware uTCA Crate 65
2.5.2.1 The MCH1: NAT-MCH module 69
2.5.2.1.1 The IPBus 71
2.5.2.2 The MCH2: AMC13XG module 74
2.5.3 Software Environment 77
2.5.3.1 XDAQ Toolbox 78
2.5.3.2 Setup Libraries 80
2.5.3.3 RPC Online Software: A Technical Overview 81

3 Development of the Slow Controller System 84
3.1 Initial Design 84
3.2 uTCA Infrastructure 85
3.2.1 NAT-MCH 86
3.2.1.1 Installation of IPBus and ControlHub 88
3.3 Setup AMC13XG for Slow Controller 90
3.3.1 Installation Settings to AMC13XG 90

4 Implementing Instrumentation and Control 94
4.1 Test Connections to AMC13XG 94
4.2 Local Event Builder Test 96
4.3 DAQ Outputs 98
4.4 DAQ Inputs 99
4.5 Python in Slow Controller 100

5 Results and Discussions 102
5.1 Initial Considerations 102
5.2 Advances in Firmware 103
5.3 Integration in LHC signals 105
5.4 Electronics Design 106
5.5 Application Development 111

Conclusions 112

Initial Considerations 114
The Research Successful 115
Innovations in the Theme 115
Future Works 116

References **117**

Introduction

Text Structure and Organization

For a better understanding of the structure of this Doctoral Thesis, the document is divided into an introduction about the subject and five chapters describing the problem and the instrumentation used to solve it. In the last section, the conclusion is described with all contributions to physics and research and development in the subject. Figure 1 displays the timeline of the thesis.

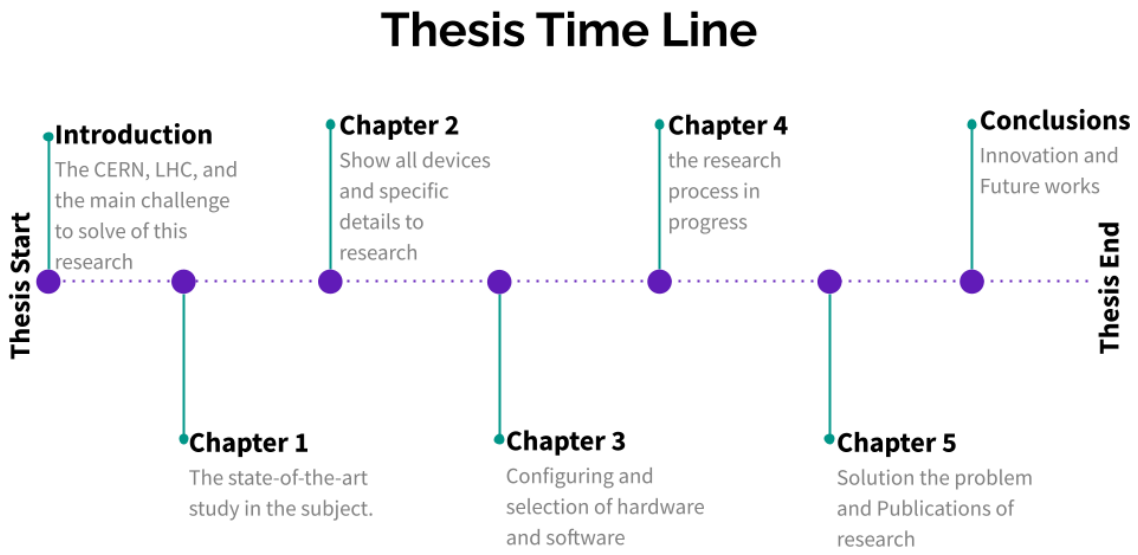


Figure 1 – Illustration of developing thesis timeline context.

The initial chapter, Introduction, opens the reference regarding high energy physics and information about the CERN and the Experiment CMS in LHC with the relationship between the Brazilian scientific collaboration project and the needs that justify the development of this work.

Chapter 1 shows theoretical and experimental concepts of instrumentation in High Energy Physics (HEP) experiments as a solution for monitoring and control, showing the iteration of this type of technique and its use with detectors in the most diverse experiments

in HEP.

In addition to this review of the literature on the subject, a summary of the current system of the CMS Experiment is introduced in its methods and subsystems, highlighting the current RPC Link System with its characteristics and the limitations and relationship with the developed Slow Controller System.

Chapter 2 refers to all the electronics involved in the instrumentation of the systems and subsystems that make up the project in addition to describing the assembly methodology and its specific configurations, as well as the subsystems that are integrated into the central control by a solution using software and hardware in the proposed system.

Chapter 3 refers to building and assembling the Slow Control System. The description covers the assembly and specific configurations for each hardware system, the testing and commissioning of each hardware, and finally, the connections between them to build a functional Slow Controller System.

In Chapter 4, an experimental setup is shown to insert monitoring and control routines in the Control Board of the New RPC Link System, using several available ways and connections and its options at the software level for sending and receiving through experimental laboratory procedures with the equipment used in the project.

In Chapter 5, the results obtained from the assembly and acquisition of the variables monitored and controlled by the Slow Controller are shown, with discussions through explanations and details of the results obtained in experimental laboratory procedures;

Finally, the chapter of conclusion discussing with the system's feasibility, its replication capacity to satisfy the New RPC Link System's needs, and the qualitative and quantitative evaluation of the developed system's performance and functionality. The final considerations describe the obtained results and suggest proposals for the continuation of future work.

Slow Controller Concept

The Slow Controller Concept revolves around Control and Monitoring Systems or Subsystems, characterized by their high complexity and non-commercial nature. As a computational system, the Slow Controller is pivotal in failure prevention, monitoring, and

user-directed and automated controls.

It ensures a harmonious and safe interplay between the leading systems and their underlying subsystems, offering a unified interface across various management levels. This controller integrates hardware and software services, facilitating functionality across internal or external applications. These services encompass a mix of commercial and non-commercial controls as in the ATLAS Experiment (RAUSCH; TAVLET, 1999), shown in Figure 2.

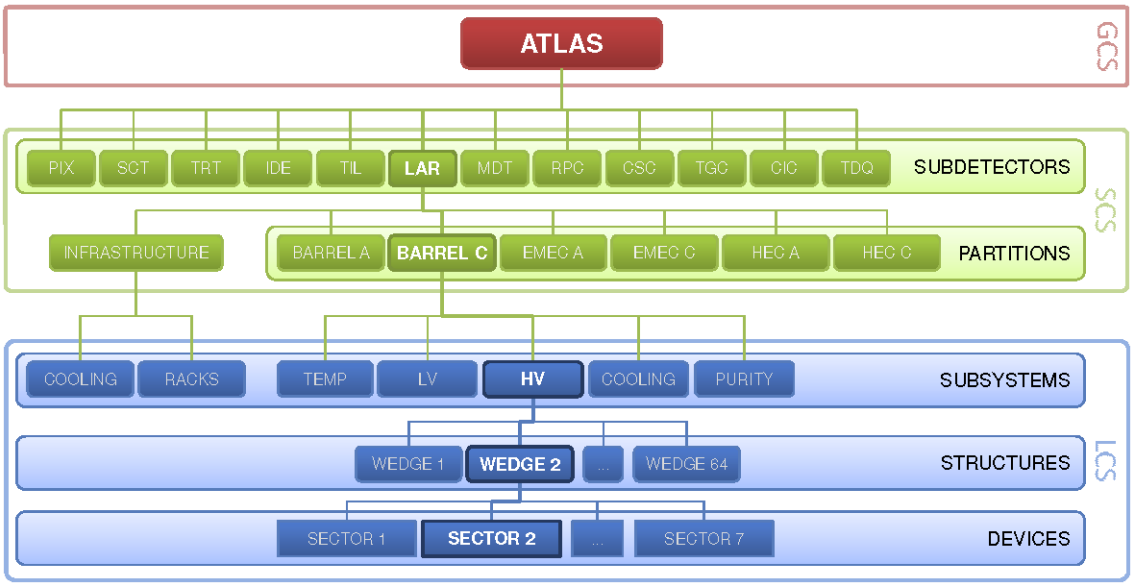


Figure 2 – Hierarchy of a control system in the ATLAS experiment. Adapted from (POY et al., 2008).

In any experiment or system, the Slow Controller must demonstrate the capability to implement control over its desired operating state, continuously monitor and archive operational parameters, identify any abnormal behavior for the operator, and facilitate both manual and automatic actions.

For this type of system, it is essential to establish standards for hardware usage and select a commercially controlled software product to serve as the foundation for all Slow Control applications.

A comprehensive framework of components, including software modules for implementation and their respective usage policies and licenses, significantly extended these software packages. Consequently, the hardware platforms employed in Slow Controller systems can vary from a simple embedded system to an industrial computer mounted on racks.

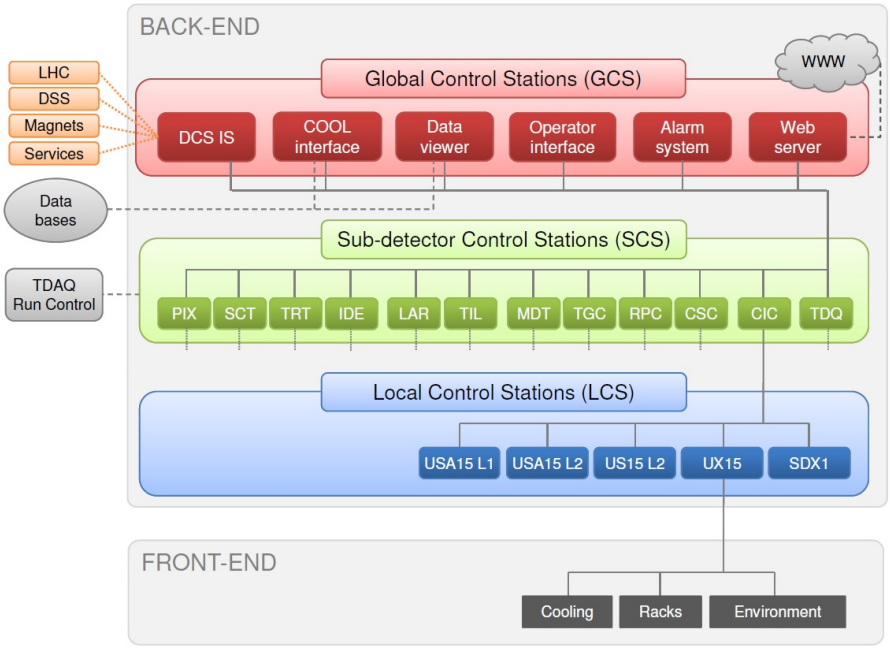


Figure 3 – DCS architecture in the ATLAS experiment. Adapted from (AAD et al., 2008).

In ATLAS, this system is called Detector Control System (DCS) due to its component systems and subsystems and common hardware standards, being extended throughout the design and implementation of the software and hardware for all of the DCS, as well as the operational aspects. Figure 3 shows a Slow Controller model implemented by DCS.

The LHC and the CMS Experiment

The LHC – Large Hadron Collider – of the European Organization for Nuclear Research, CERN (THE CMS COLLABORATION, 2008) is a pioneer in the study of Elementary Particle Physics at the highest energies obtained in the laboratory.

CERN has already produced excellent scientific results as in mid-2017, the CMS experiment published more than 600 articles in refereed journals. The highest point was the discovery of the Higgs boson by the experiments at ATLAS and CMS in 2012 (AAD et al., 2015).

The collisions in the LHC it produce unstable particles of high mass and they can

be observed in the detectors. These observations, for example, led to the discovery of the Brout-Englert-Higgs boson by the CMS and the ATLAS (COLLABORATION et al., 2012) detectors.

The LHC is the largest and most powerful particle collider in the world. Inside the LHC, two beams of protons or lead ions are accelerated in opposite directions. These beams collide in four particle detectors: ATLAS, CMS, A Large Ion Collider Experiment (ALICE), and Large Hadron Collider beauty (LHCb), seen in Figure 4.

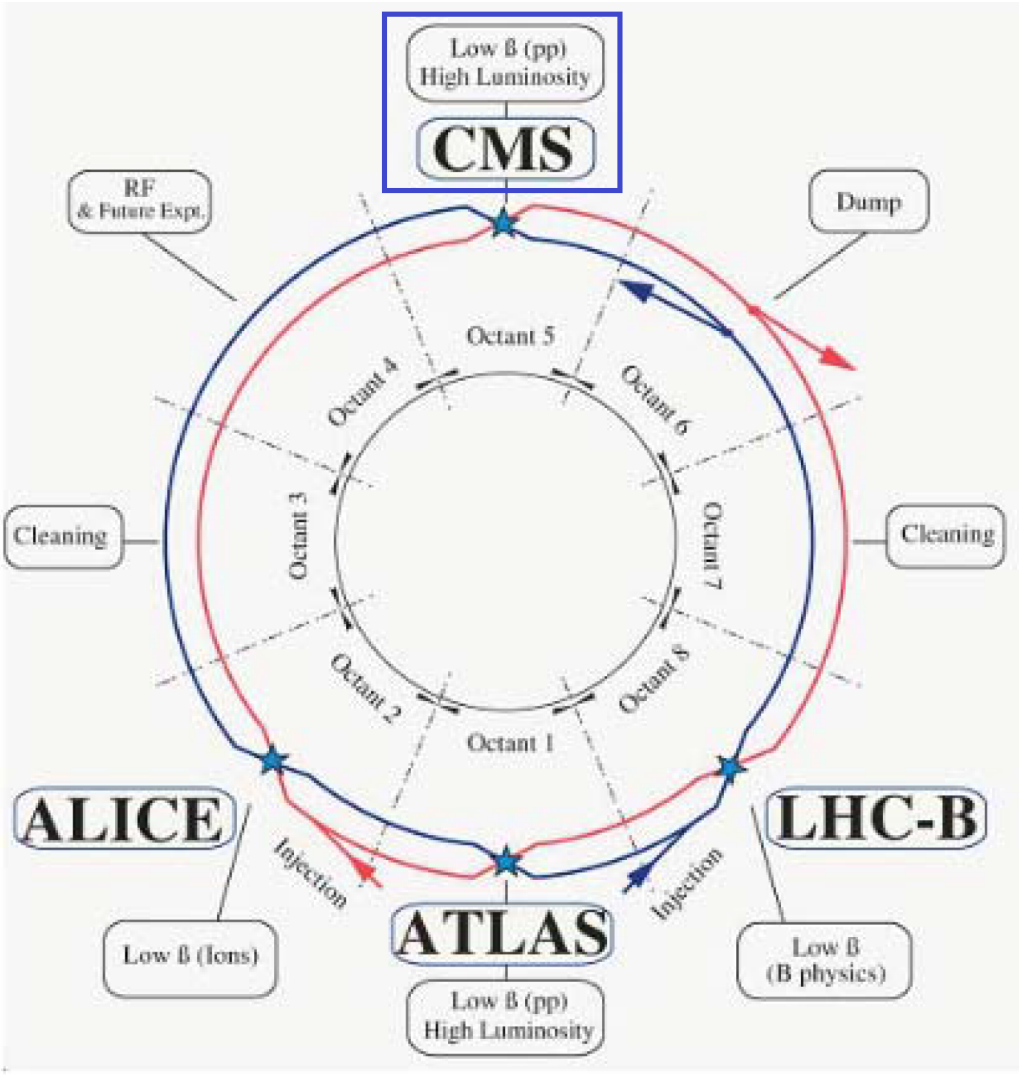


Figure 4 – LHC diagram and breakdown of main detectors. The two beams and their collision points are also visible with highlights in CMS Experiment. Adapted from (EVANS; BRYANT, 2008).

In this great particle accelerator, CMS is looking for answers to fundamental questions in particle physics: Are all the properties of the 125 giga-electron-volt (GeV) Higgs Boson consistent with the Standard Model of Particle Physics (the theory describing

electroweak and Strong interactions – Standard Model (SM)). Is nature supersymmetric? Can we explain dark matter in the universe by particles that can be produced at the LHC?

The CMS experiment has been very successful. The detector is working without problems and the energies of the center of mass with proton-proton, have a world record up to 13,6 tera-electron-volt (TeV), with an LHC luminosity of 10^{34} cm^{-2} s^{-1}. The SM has been tested in an unprecedented regime and with high accuracy.

The key missing SM particle, the Higgs boson, was discovered through its decay modes, two photons and four leptons (AGUILAR-SAAVEDRA et al., 2012; CHATRCHYAN et al., 2012). New Beyond the Standard Model (BSM) particles – in addition to the standard model – have been researched in many channels, and strict limits have been set, but so far no new physics has been found.

The LHC experiments, and in particular the CMS, allows the study of particle physics to be carried out at energies never before reached in terrestrial accelerators.

All of the results derived from this class of experiments are extremely important, not only for the understanding of elementary particles but also for other areas, such as cosmology and astroparticles.

As usual, the importance of this project is associated not only with the extraction of scientific results, which alone justify the investment made but also with the technological development in several areas, like computing and electronics.

One of the characteristics of this project is the development of cutting-edge technology to be used in the improvements of data acquisition and selection electronics (trigger) of the CMS Detector.

The LHC update

From its beginning in 2010 to its predicted end of life in 2035, the LHC alternates between data acquisition periods and prolonged outages, known as Long Shutdown (LS). Engineers use these extended shutdowns to perform repairs, update the system, and incorporate new features into the LHC.

Figure 5 displays a calendar of these periods, Long Shutdown 2 (LS2) and Long Shutdown 3 (LS3), regarding the project and presents the estimated increases in both

instantaneous and integrated luminosities. The calendar of LS2 concentrates on Phase 2 runs from 2017 to 2024, while Phase 3 in LS3 spans from 2024 to 2027.

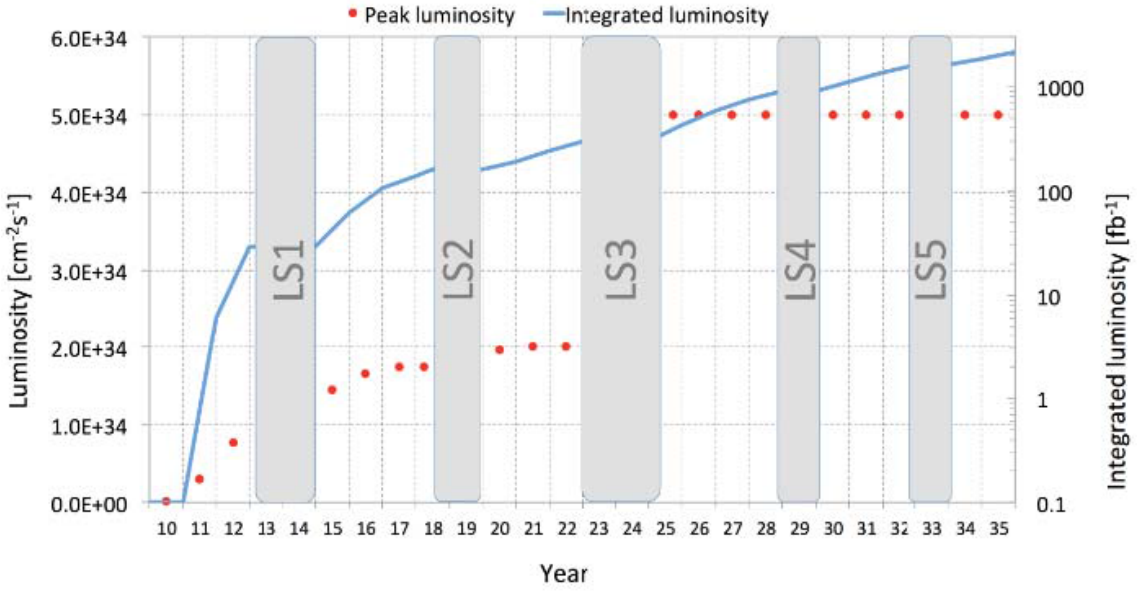


Figure 5 – Pre-pandemic calendar of the different periods of LHC (THE LHC EXPERIMENTS COMMITTEE, 2017). Adapted from (EVANS; BRYANT, 2008)

✓ Phase 2

Phase 2 has received some date changes and runs from 2017 to 2024. During this phase, the luminosity along with the collision energy, will be progressively increased. It will be divided into one period, each time separated by shutdown periods. During the writing of this Doctoral Thesis, LHC is at the end of its second period, called LS2. The CMS Muon System Detectors update project is expected to be integrated into the entire CMS as soon as this phase of LS2 is finished, between 2021 and 2023.

• Phase 3

Phase 3 will last from 2025 to 2027. It will start with a very long shutdown phase of more than 2 years and is also made up of 3 periods separated by shutdowns. During this phase, LHC will enter the High Luminosity Large Hadron Collider (HL-LHC) phase. Brightness will be increased by a factor of 2.5 over the last periods of phases 1 and 2 and will be held constant during phase 3.

Brazil Collaboration on CMS

In the area of hardware development, we can highlight the Brazilian collaboration in the CMS experiment: Work was carried out on the improvement (upgrade) of the CMS detector, in particular, on the Frontal Hadronic Calorimeter (HF) of this experiment, to greater precision to energy measurements in this region.

In this sense, we developed a new calibration board for the HF Front-End Electronic (FEE). *Micropress Circuitos Impressos Ltda* developed this board and assembled it by *Produza S/A*. Therefore, it can show in Figure 6.

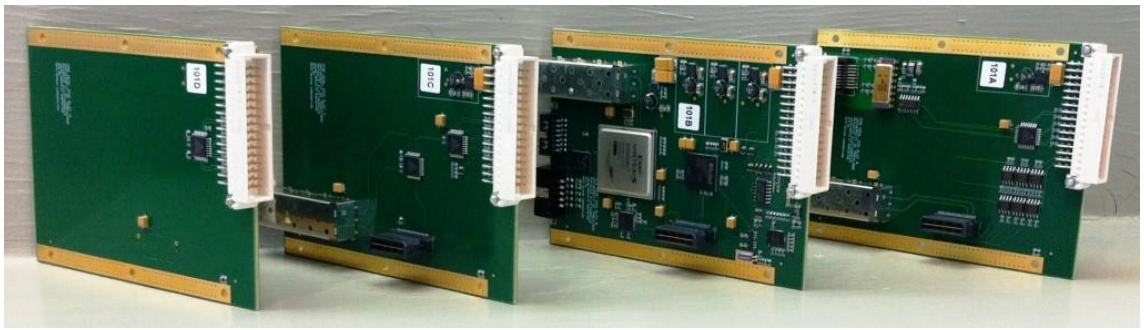


Figure 6 – FEE Printed Circuit Board.

In addition to the development, in Brazil, of this Printed Circuit Board (PCB), the firmware associated with the FEE, and the software for the database of the HF were also developed.

Cooperation between Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ), Centro Brasileiro de Pesquisas Físicas (CBPF), and CERN was signed to carry out the development of this doctoral project that has a vital component in scientific instrumentation and possible mobilization of Brazilian industry.

All the investment made so far in the CMS experiment by Brazilian agencies, notably Financiadora de Estudos e Projetos (FINEP) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), was carried out in component development, together with national companies, as mentioned above.

Figure 7 demonstrates the importance of Brazil's collaboration with CERN, as it depicts one of many Brazilian researchers holding important positions in a large project like the HF of the CMS/LHC.

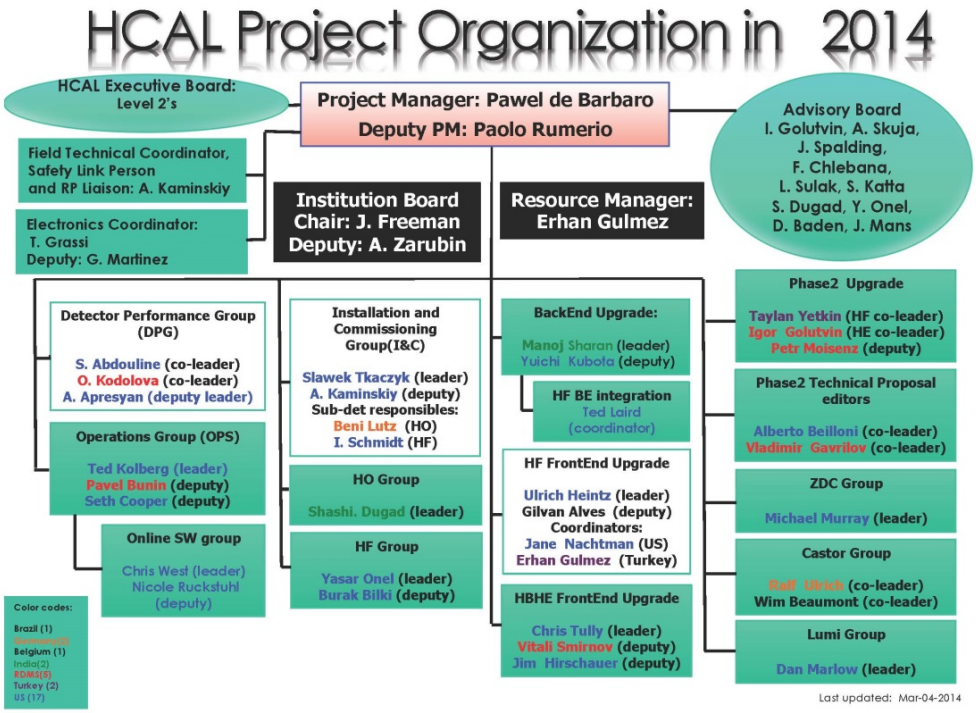


Figure 7 – Brazilian Collaboration in the CMS/LHC Hadronic Calorimeter Project in 2014.

Motivation

The CBPF collaborates scientifically with CERN, driven by the ability to generate experimental data from the study of increasingly precise particle collisions produced by the LHC. This collaboration also stems from the necessity to derive new findings for diverse types of particle detection, particularly in the Muon System in the CMS Experiment.

In 2018, the Brazilian scientific collaboration began work together developing the firmware related to Front-End Electronic Board (FEB)s for the project components of enhanced improved RPC (iRPC) type detectors (ASILAR, 2022) at CERN. Following the development, they concluded the necessity for a new slow controller system to progress further (SHCHABLO et al., 2021).

The forthcoming FEBs will transmit raw data at nearly 10 Gigabits per second (Gbp/s) to the updated RPC Link System in the new Back-End system within the Muon System. The existing Link system maintains only hardware-level control for data transmission between the LB and the CB.

This subsystem must align with the new Back-End system and remain functional with the existing technology in the CMS Detector. Figure 8 depicts the projected development of a dedicated subsystem.

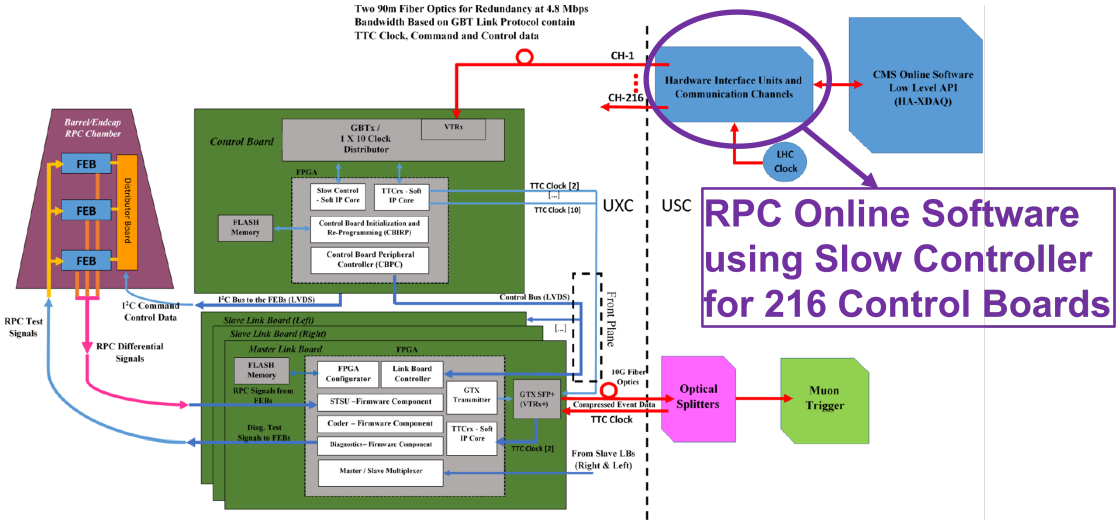


Figure 8 – Hardware Interface Units and Communication channels like the new Slow Controller system. Adapted from (THE LHC EXPERIMENTS COMMITTEE, 2017).

This work's research revolves around these novel FEBs and Link Systems, which necessitate Slow Control at both hardware and software levels. This requirement arises from the significant update for enhancing data acquisition in the CMS Experiment.

The emergent slow control system will influence the data systems of the FEBs, the new link system, and the new Back-End system (GOLA, 2020). This System ensures compatibility across all these stages in acquiring RPC data.

Moreover, all these systems should communicate with both the existing and new detectors in the muon system, considering future upgrades in the subsystems of the new RPC Link System (BOGHRATI et al., 2021).

The new Slow Controller in the updated Muon System, within the RPC Link System, will contribute to reaching the CMS Detector's data communication in this subsystem speed of up to 10 Gbp/s, thereby fulfilling all the requirements of the High Luminosity HL-LHC.

Establishing a Slow Controller subsystem to oversee and regulate the New RPC Link System is vital. Experimental work conducted in CERN's laboratories regarding the new Back-End system (CAO et al., 2021) offered substantial information to commence the Slow Controller project.

The Slow Controller's development involves connecting 216 links of the Hardware Interface Units and Communication channels on the CB in its final form. This execution

employs uTCA-type racks for swift data communication, supported by their Advanced Mezzanine Card (AMC) modules.

This work pioneers in its field, as no system of this magnitude exists presently. It primarily motivated the start of a technological research for a Slow Controller in a new RPC Link System, and represents a unique, innovative research endeavor in instrumentation and control.

General Objectives

To meet the main purposes of the upgrade in LS2, the Muon Detector System will also be responsible for receiving the signals and sending them to the next stage at a higher level, using the RPC Link System. The prototype in the progress of the subsystem Slow Controller can be seen in Figure 9.

This current RPC Link System does not have a Slow Controller system by software and there is no kind of monitor agent in real-time. Currently, this control is only by hardware with minimal functions that reaches a data transmission rate of 1.6 Gbp/s and has no feedback to keep working or in case of any problems.

This maximum current data transmission rate of 1.6 Gbp/s is over cables and wires in Low-Voltage Differential Signaling (LVDS) and will be replaced by optical fibers so that the high data transmission rate is compatible with HL-LHC.

By meeting these specifications for the HL-LHC, the collision data transmissions from the RPC detectors will be immune to electromagnetic interference due to high radiation at the experiment site.

The new Slow Controller will contribute to the RPC Link System reaching a speed of 10 Gbp/s, high continuous data transmission in time without interruptions, and offering a secure data transmission without losing information. Furthermore, following the specification of the RPC Link System, the Slow Controller will also be able to work in a high-radiation environment.

Increased data transmission speed and improvement of the data acquisition time resolution will increase the capacity of the data input rate in the system, and with that, it will increase the data output bandwidth of the system, but for this, it is necessary to have a

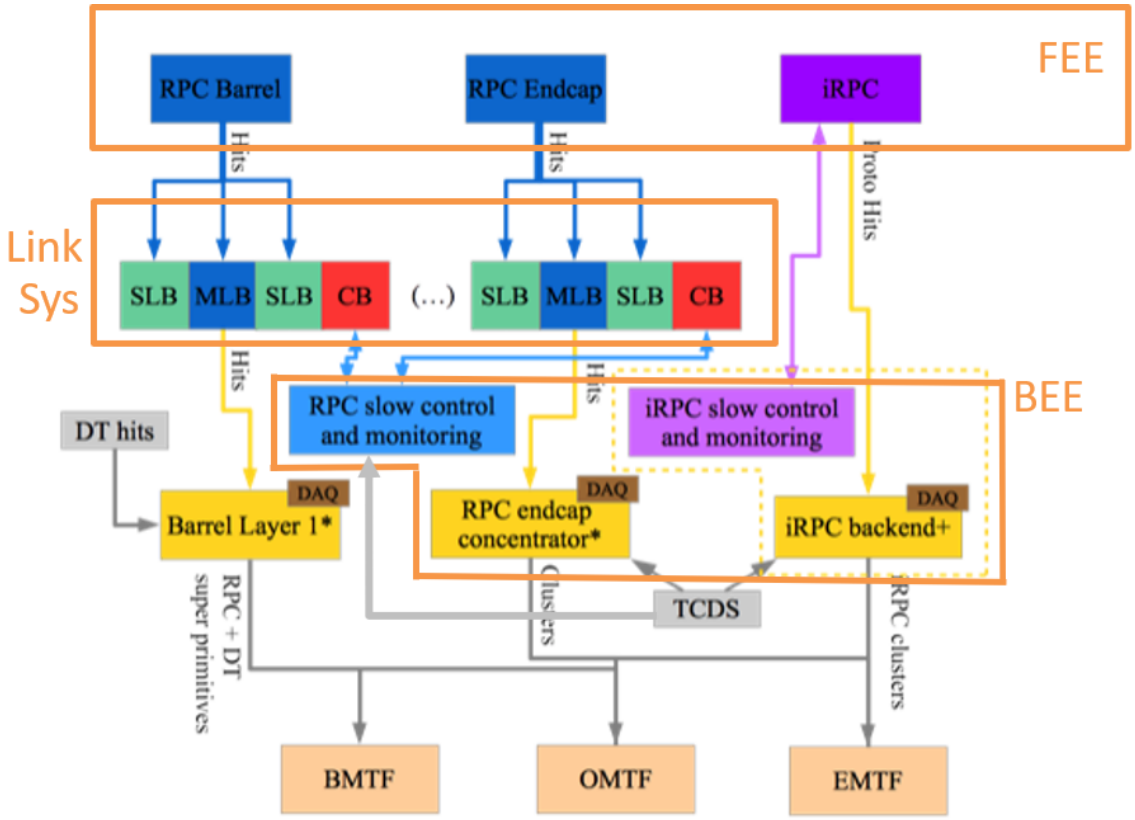


Figure 9 – The subsystems that Slow Controller (in light blue) acts in the RPC Detector.

new Slow Controller more efficient and with the ability to control and manage these new features of this new RPC Link System.

This subsystem will be of fundamental importance to managing and monitoring this large flow of data received in this new version of the RPC Link System.

This management of the RPC Link System also includes another no less important function, which is the development of the hardware for reading and monitoring the status from CB, LB, and FEBs, generating a general status report, which also has these features of the new Slow Controller.

The Slow Controller will have as one of the main tasks in its development to control the data coming from the FEEs through the CBs through their optical links, which will be implemented with technology Radiation tolerant GBT protocol high speed bidirectional ASIC (GBTx).

To adhere to CERN standards for implementing monitoring and control, the experimental prototype of the Slow Controller system must possess the capability of seamless integration throughout all phases of development. This ensures the proper alignment with CERN guidelines and requirements.

The first version of Slow Controller (small-scale system), following phase 2 (medium-scale system), and version final (large-scale system) all of the phases for that system project at CERN can be seen in Figure 10.

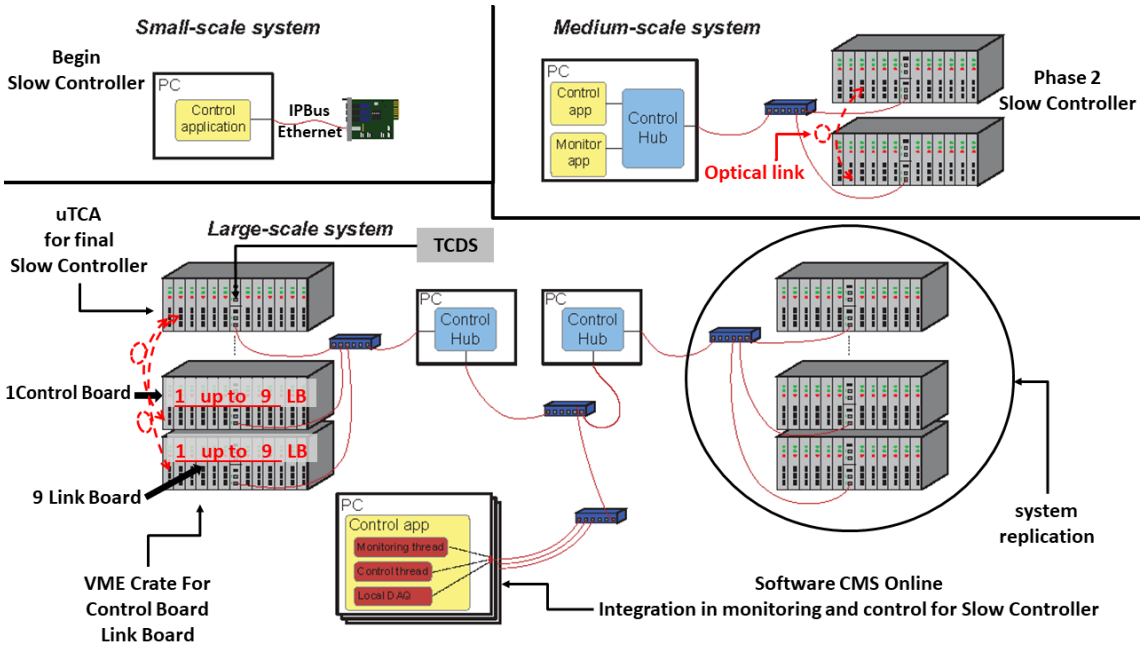


Figure 10 – The 3 phases foreseen for the development of Slow Controller in the CERN Project. Adapted from (LARREA et al., 2015).

Another main development point is the monitoring of the general state of all hardware where the final version of Slow Controller operates, in 216 CB units and 1376 LB units, all of them FEEs distribute for 480 chambers in Barrel and finally 648 in End-caps.

Furthermore, the final prototype system will allow the reconfiguration of the Field Programmable Gate Array (FPGA)s in the FEB remotely without interrupting the service, using optical communication links. In addition, all these electronics in use pass a challenging test for high radiation.

Also, in this thesis project, the implementation of the integration of the Back-End system and CMS Timing and Control Distribution System (TCDS) with the final version of Slow Controller will be addressed.

This new Slow Controller system will be integrated into the Back-End System. It will improve the angular resolution of the trajectories increasing by a factor of about 6, correlating more information from Bunch and Track Identifier (BTI) and TRAck COrrelator (TRACO).

Specific Objective

The specific objective of this Ph.D. project centers on creating a streamlined experimental system that embodies both hardware and firmware components for the RPC Link System. This system fosters a more sophisticated and practical approach to link system research and application.

Figure 11 visually represents the preliminary Slow Controller system. This system illustrates its potential to facilitate effective data transfer command exchanges between the PC Host and Optical Links. Doing so allows for the rigorous testing of AMC13XG Network Settings within the Data Acquisition Controller (DAQ) Link Modes of Operation.

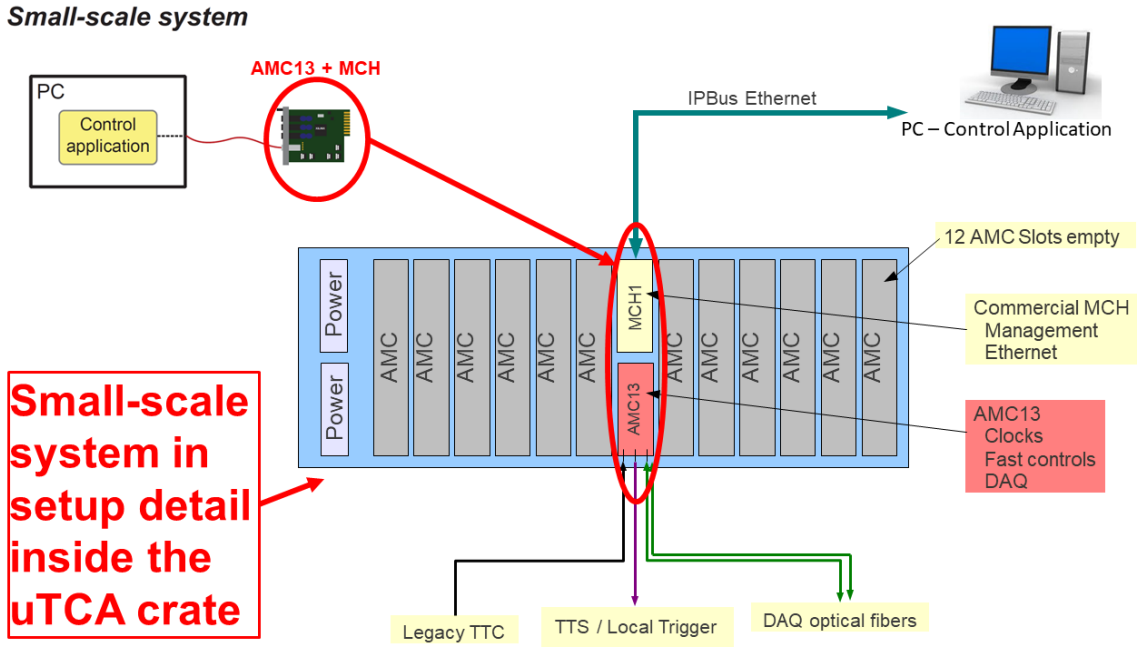


Figure 11 – Initial setup prototype of the Slow Controller in the laboratory.

In addition, the system brings latency and backpressure capabilities to the fore (to use SRAM memory if it exceeds its temporary storage capacity), with speeds extending from 5 Gbp/s up to 10 Gbp/s. This capacity offers valuable insights into the system's performance at different rates but also aids in assessing its compatibility with the setup's developmental requirements.

Following the precise adjustment of AMC13XG's optical data transmission settings, it becomes necessary to implement additional software configurations within the system. This modification is essential for the system to facilitate efficient data transmission to and

from the control board.

Figure 12 shows the Control Board's connection setup within AMC13XG, further cementing the understanding of its operational dynamics.

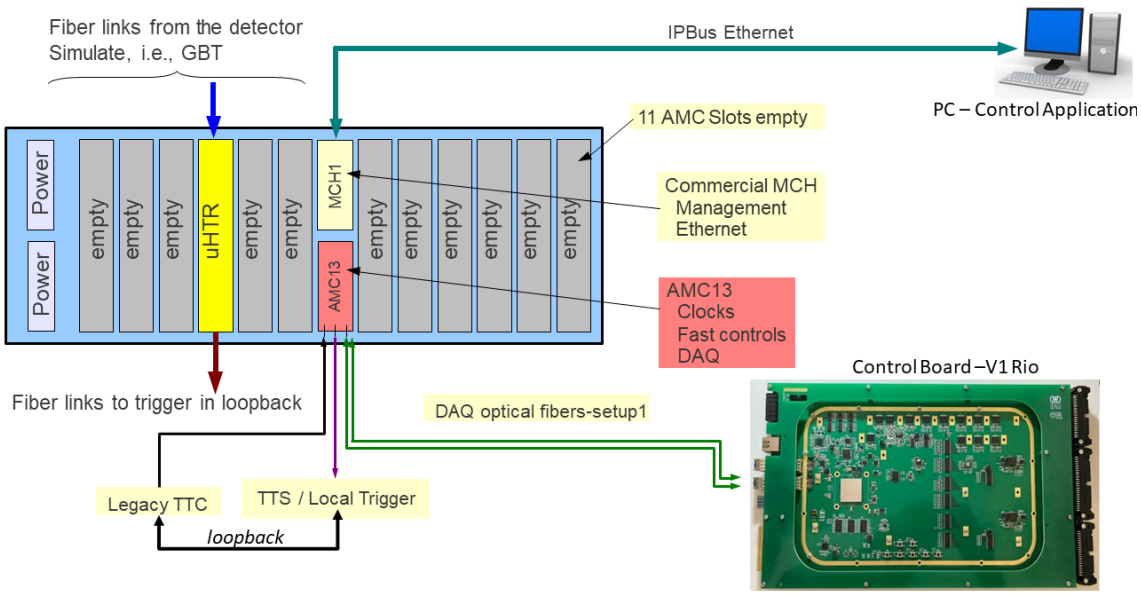


Figure 12 – Phase 2 of the Slow Controller in the laboratory with AMC13 connected at Control Board-V1 testing the monitoring and control functionalities.

Furthermore, assessing any firmware updates in alignment with the set requirements is vital, ensuring optimal control over the system's development.

The careful and regular evaluation of these updates is critical in effectively guiding the system development, thus ensuring the highest performance and functionality.

Justification

One of the areas of focus of the Programa de Pós-graduação em Instrumentação e Óptica Aplicada (PPGIO) at CEFET/RJ is the area of Scientific and Technological Instrumentation. This area ranges from theoretical research in systems modeling and simulation to developing methods and technologies related to optical devices and systems and scientific instrumentation applied to physics experiments.

In this doctoral project, we propose the development of processes and systems

using interfaces based on scientific instrumentation. Our collaborative efforts with the CERN aims to generate nationally focused solutions for problems in applied physics within electronics and data acquisition systems.

This thesis will present how the RPC Link System works and its need to develop a new Slow Controller system that can read and process detailed information from RPC's detectors that are extremely fast, based on FPGAs and interfaces used for the CMS particle detector.

During HL-LHC operation, the expected conditions in terms of the use of physical space, radiation, and the likely aging of current detectors will make muon identification and correction of the pT assignment challenging for the muon system.

To ensure the security and redundancy of data originating from the RPC Detectors and received by the RPC Link System, the new Slow Controller actively monitors the data under the new conditions of HL-LHC.

In addition, two significant relevant updates will separately complement this development: the newRPC Link System and the New Back-End system.

The detectors' high-speed data communication electronics, called the RPC Link System, will undergo replacement for the RPCs, and the RPCs' coverage will be extended in angular coverage by η = 1.9 to 2.4. Figure 13 displays the region for new RPC placement to expand coverage (inside the red selection) and the region impacted by the Link System update (purple selection). The RPC Link System has the function of processing, synchronizing, and zeroing the signals coming from the FEB in the RPCs with a time resolution of 25 nanosecond (ns);

With the new Back-End system, a suitable identification of Bunch crossing (BX) will be obtained with BTI when four aligned hits are in the collision reconstruction process. The Back-End will provide higher quality reconstruction in the CMS Muon System.

One of the main justifications of this research is seeking more effective participation in the transfer of cutting-edge technology developed in international cooperation projects.

Based on previous examples, we expect national companies in Brazil with sustainability certificates to partner in designing and manufacturing components for this New Slow Controller System as part of this development.

With all of the additional improvements, the new Slow Controller will support new functionality to collect information on the expected full load, including an additional safety factor of three (hardware), and to provide more details in the database for the

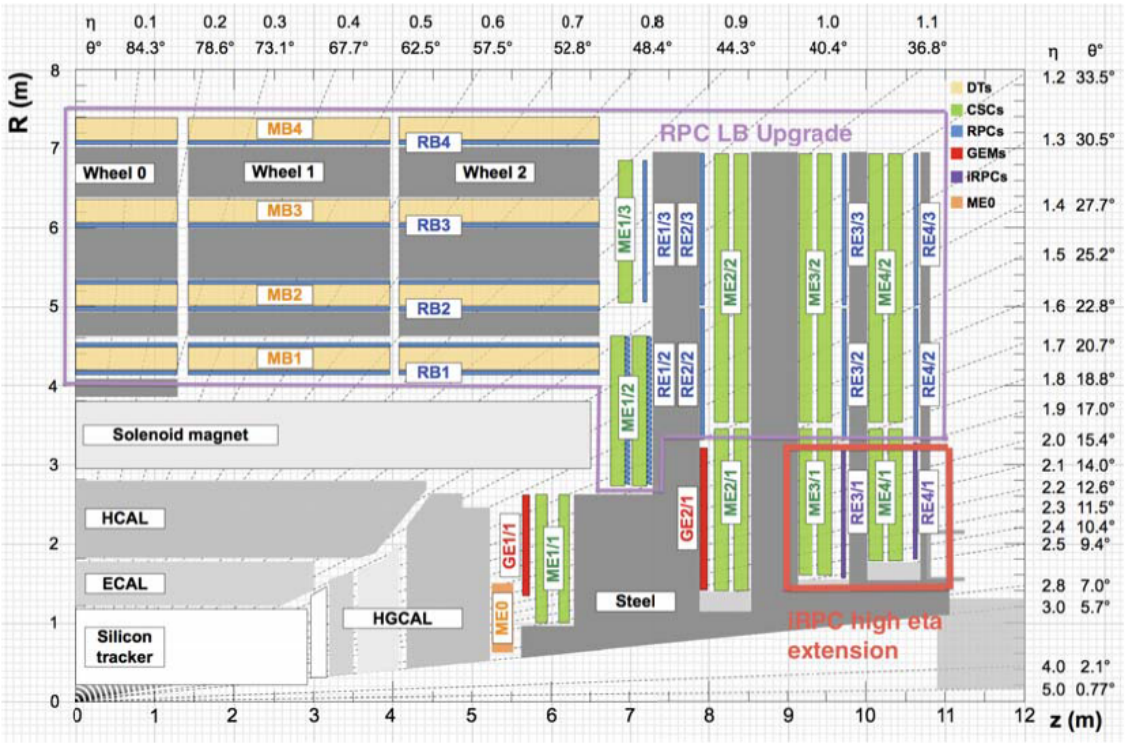


Figure 13 – In the quadrant of the CMS experiment, the red selection indicates the region where additional RPCs will be placed to extend the muon coverage, and the purple outline indicates the region interested in updating the RPC Link System.

reconstruction of more accurately trajectories in the longitudinal projection that are under-explored so far.

The New Slow Controller System also helps to get better precision and improved time resolution in other subsystems described here. That work will open the window to explore new physics channels with long-lived particles or new kinds of particles.

1- Bibliographic Revision

1.1- The Slow Controller in Large Experiments

1.1.1 CMS detector at the LHC

The CMS is one of two multipurpose detectors at the LHC, where it is the second largest international scientific collaboration, currently involving 4,300 particle physicists, engineers, technicians, students, and support staff from 182 institutes in 42 countries.

The experiment's construction aimed to measure the Higgs boson, electroweak symmetry breaking, and potentially discover new physics.

Colliding particle detectors in high energy physics experiments, such as CMS, have an ”onion”-like structure with each layer consisting of a sub-detector, as seen in Figure 14 (CMS, 2020; THE CMS COLLABORATION, 2008).

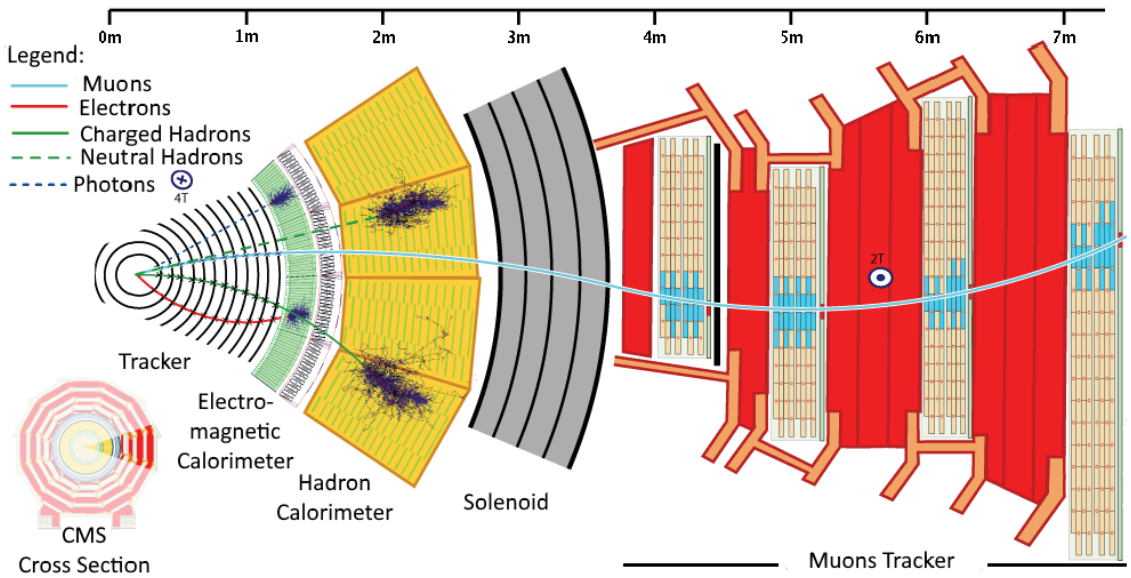


Figure 14 – The schematic sectional view of the CMS detector with the geometric and subsection components. Looking outside the CMS at the central interaction point, we have the Silicon Tracker, the Electromagnetic Calorimeter, the Hadronic Calorimeter, the Superconducting Solenoid, and the Muon Chambers.

CMS subdetectors are designed to measure the various characteristics of particles created in proton-proton collisions.

The CMS Cartesian coordinate system has its origin at the Interaction Point: The x axis points to the center of the LHC ring, and the y axis indicates perpendicularly upward to the plane LHC and the z axis points in the counterclockwise beam direction.

The azimuthal angle and the polar angle define the corresponding cylindrical coordinates. The azimuthal rise ϕ is measured in the plane xy starting at x, while the polar angle θ is measured in the plane rz starting at z.

In high energy physics, instead of the angle θ, η (DOOLING, 2015), what is called *pseudorapidity* is used. The equation 1 defines the function of the production angle concerning a beam axis.

$$\eta = -\log\left(\tan\frac{\theta}{2}\right) \tag{1}$$

The CMS detector consists of a colossal solenoid magnet with cylindrical coils. It is 21m long, 15m wide, and 15m high.

Unlike the other detectors in the LHC experiment that was built *in situ*, the CMS was built at surface level in 15 sections and taken in parts to the underground complex at point 5 at Cessy in France, and then mounted. The information of CMS subdetectors is listed in Table 1.

Table 1 – Information on CMS components and their purposes. Modified from (DOOLING, 2015).

CMS subdetectors	Parts	Coverage in η	Purpose
Superconducting Solenoid Magnet	4 layers	-	Bending of charged particle trajectories
inner Tracker	pixel strip	< 2.5 < 2.5	Measurement of charge and momentum
ECAL	barrel endcap preshower	< 1.479 1.479 - 3.0 1.653 - 2.6	Energy measurement electrons and photons
HCAL	barrel endcap forward outer	< 1.4 1.3 - 3.0 up to = 5.0 < 1.3	energy measurements of hadrons
Muon System	barrel endcap	1.2 0.9 - 2.4	Identification and measurement of muons

Researchers can find new physics in this experiment by discovering new particles or considering deviations from the Standard Model. The detector's design enables it to identify high-momentum resolution photons, electrons, and muons.

1.1.1.1 CMS Sub Detectors

The CMS Experiment has five sub-detectors that constitute this experiment. These sub-detectors are:

- ***Silicon Tracker***: The silicon detector can reconstruct the path of muons, electrons, and charged hadrons. By being closer to the collisions, the decays of particles of short duration (BARNEY, 2015) can be detected. It makes very accurate measurements of the location of particles, with an accuracy of up to 10 μm.

- ***Electromagnetic Calorimeter***: The Electromagnetic Calorimeter (ECAL) consists of 75,848 lead tungstate crystals (PbWO4) weighing approximately 100 tons, designed to halt and measure electrons and photons' energy. These crystals emit light as electromagnetic photon showers, which photodetectors (avalanche photodiodes or vacuum photo triodes) measure. The crystals exhibit radiation resistance and can recover from radiation damage when maintained at room temperature; they are cooled to 0.1°C during beam time.

- ***Hadronic Calorimeter***: The Hadronic Calorimeter (HCAL) is specifically designed to detect hadrons, particles produced from quarks and gluons. This detector part is organized into dense absorbing and scintillating layers, combining steel and scintillators. The scintillators produce a pulse of light when a particle flows through it. The fact that Hadron Calorimeter (HCAL) is housed inside the solenoid is one of the most significant differences between ATLAS and the CMS experiment.

- ***Superconducting Solenoid Magnet***: To equip the CMS with a sizeable superconducting solenoid capable of generating a 4T magnetic field was necessary to complete this high magnetic field. Allows for very accurate momentum readings by analyzing the trajectories of charged particles. Additionally, the solenoid provides enough return flux outside the solenoid to be used by the muon chambers, with about a 2T magnetic field in that region.

- ***Muon Chambers***: Muons and neutrinos are the only particles that can pass through all detector sections without losing most, if not all, of their energy. Bethe's Equation 2 describes the power loss a particle experiences as it travels through matter. Due to the characteristics of muons, this equation states that muons will not release much of their energy as they pass through the material. Combined with these particles having a high mass, they are complicated to stop.

$$-\left\langle \frac{dE}{dx} \right\rangle = \frac{4\pi}{m_e c^2} \cdot \frac{nz^2}{\beta^2} \cdot \left(\frac{e^2}{4\pi\varepsilon_0} \right)^2 \cdot \left[\ln\left(\frac{2m_e c^2 \beta^2}{I \cdot (1-\beta^2)} \right) - \beta^2 \right] \tag{2}$$

Muon chambers, unable to stop muons quickly, utilize the strong magnetic field to measure the curvature of these charged particles and, consequently, determine their momentum. The muon chambers are composed of three components: the RPCs, the Drift Tubes (DT)s, and the Cathode Strip Chambers (CSC).

This thesis focuses on the Muon System as a critical consolidation point in this phase, as the Slow Controller must develop within this part of the CMS experiment. In addition to the RPC detectors, the DT and CSC detectors complement the complex particle detection system of the muon system to identify and measure muons with more precision needed to fulfill the physics program in the CMS update.

Consequently, maintaining Slow Controller performance in high luminosity conditions must be ensured. Several upgrades of detector electronics are necessary, specifically focusing on technology and application to RPC detectors. This thesis project will concentrate on the muon detector, primarily its data transmission system, the RPC Link System.

1.1.2 The Slow Controller in the CMS HCAL

In the article published by (COSTANZA et al., 2017), the next generation Front-End Controller (ngFEC) is the system responsible for the Slow Controller and the tight control within the Phase update I of the HCAL of the CMS. This job is based on Fast Control Board with FPGA Xilinx Kintex-7 (FC7), an AMC compliant uTCA developed at CERN and built around the Xilinx Kintex 7 FPGA.

ngFEC decodes the 40.0788 MHz LHC clock and synchronization signals received from the *backplane* and distributes them to the FEEs via six protocol high speed bidirectional ASIC (GBT) links. The block diagram of Slow Controller and ngFEC in uTCA is seen in Figure 15.

The latency of tight control signals is fixed in clock cycles. Even if the direct link to a front-end module is broken, a redundancy scheme ensures successful communication using the link to the neighboring front-end module.

Thanks to ngFEC, all front-end modules can be programmed remotely using the standard Joint Test Action Group (JTAG) protocol. Furthermore, the Crate Control Module (CCM) server software interfaces ngFEC with the Detector Control System, which constantly monitors voltages and temperatures in the front-end electronics.

This other article by (ANDERSON; FREEMAN; WHITMORE, 2012) shows a schematic of the update of the front-end in the HCAL of the CMS, in the period of LS2 seen in Figure 16.

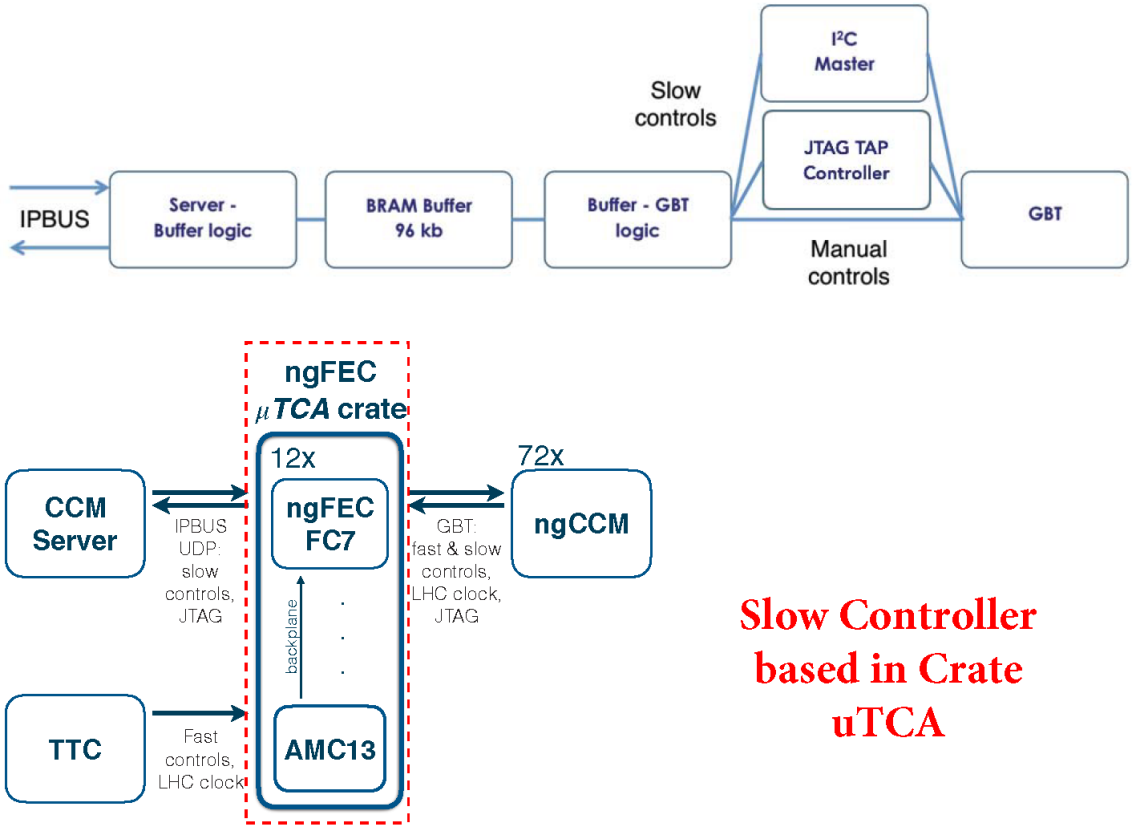


Figure 15 – Transmission to back-end and AMC13 for data acquisition to central CMS with emphasis on control (Slow Controller) in uTCA. Adapted from (COSTANZA et al., 2017).

The electronics update of HCAL is required to handle the large instantaneous increase in luminosity $\gg$ that x10^{34} cm^{-2} s^{-1} and an expected integrated luminosity of $\sim$ 3000 fb^{-1}.

A key aspect of updating HCAL is reading the segmentation information longitudinal to improve background rejection, energy resolution, and electron isolation at the L1 trigger level.

As a solution to this front-end electronics upgrade challenge, a new Application Specific Integrated Circuit (ASIC) and Analogic Digital Converter (ADC) was developed for high-speed data transmitters to handle the increased data volume with commercial FPGAs, improving thermal design and general readout architecture improvements.

We can highlight some aspects of this update, such as the Peltier system reading module with a feedback loop through the Slow Controller, which presented stability in a range of 0.1°C in the monitoring and control of the Silicon PMTs (SiPMs) cooling.

When monitoring and controlling the cooling system of the SiPMs, the data will be sent to the GBT (MOREIRA, 2007) chips that will drive 82 bits from the front-end of HCAL

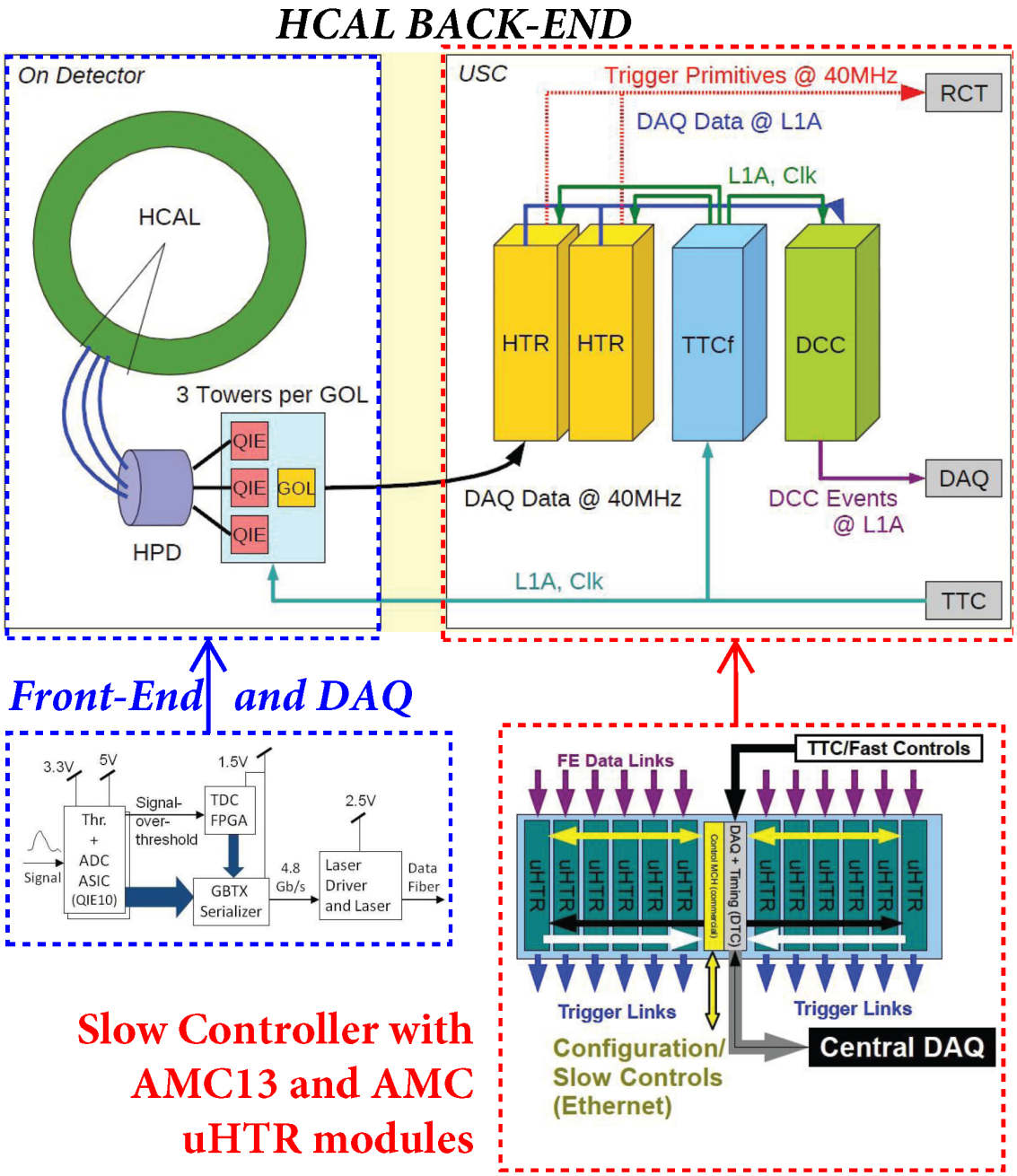


Figure 16 – Block diagram of slow controller implemented with lpGBTs. Adapted from (ANDERSON; FREEMAN; WHITMORE, 2012).

every 25 ns on a 4.8 Gbp/s serial link.

As the number of installed optical links is limited, most fibers will be up-links.

Still in Slow Controller, will also be done via GBT Opto-links, one uplink, and one downlink for front-end communication.

The clock, SiPMs voltages, Peltier cooler voltages, SiPMs leakage current, face-plate voltage readout, SiPMs faceplate temperatures, faceplate voltages, and integration

phase adjustment (ASIC/ADC) will be sent on the uplink and monitored/forwarded to the back-end.

The back-end collects data through the Slow Controller using a uTCA-based crate system, which receives data at 4.8 Gbp/s, applies thresholds and sums energies, produces primary triggers passed to the L1 level trigger and maintains the data pipeline for potential reading by the DAQ.

New trigger data modules in uTCA such as AMC13 and uTCA Trigger and Readout Module (uHTR) as data hubs are currently in use in HCAL (MANS; FRAHM, 2010; ANDERSON; FREEMAN; WHITMORE, 2012).

1.1.3 The Slow Controller in ATLAS TileCal

In (DENG et al., 2020) developed, an optical link system in a separate study to upgrade the Liquid Argon Calorimeter of the ATLAS experiment to its LS2 configuration.

The optical link system transmits data from more than 182,000 detector channels housed within 1524 FEBs through 26 optical fibers per FEB, spanning 150 meters to the designated room. Furthermore, the system facilitates the transfer of clock signals, beam crossing reset signals, and Slow Controller-monitor signals from the designated room back to the FEBs.

In addition to the I^2C configuration of the ASICs, the Slow Controller is responsible for the control links. Figure 17 shows the Slow Controller block diagram.

The Slow Controller has the ability to turn on or off all power supply modules and reset the ASICs. This feature is necessary in the event of a device crash that requires a reboot. The implementation of this feature is achieved through the utilization of Digital Input/Output (DIO) ports of the control lpGBT.

Each DIO contains on-board pull-up or pull-down resistors that ensure default configurations and regular board operations during the initialization of the lpGBT control.

It is important to note that a control lpGBT is only capable of cycling the control lpGBT, which is located on the other half of the board. Additionally, temperatures are monitored at various positions, which is implemented through the built-in ADCs converters of the control lpGBT.

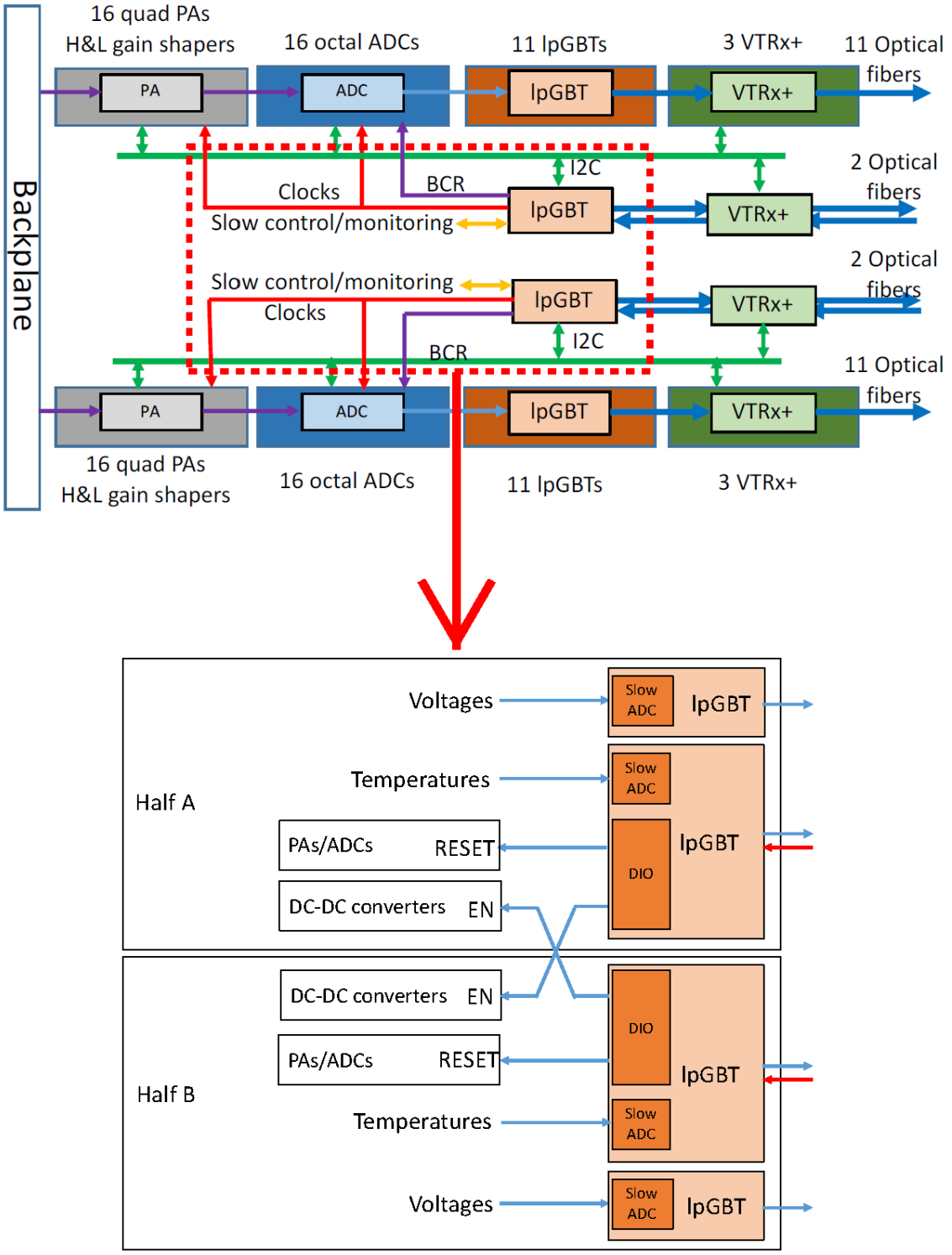


Figure 17 – Block diagram of slow controller implemented with lpGBTs. Adapted from (DENG et al., 2020).

lpGBT is unable to monitor all voltages from the power supply.

1.2- The current RPC Muon Detector System

The current RPC system consists of 480 chambers in the barrel, and 576 chambers in the endcap, organized into four stations called RB1 to RB4 in the barrel region and RE1 to RE4 in the endcap region.

The innermost barrel stations, RB1 and RB2, are instrumented with RPCs on the inner and outer faces of the corresponding DT chambers. All other RPC stations on the barrel and end caps have one RPC chamber per station.

The RPC of the CMS consists of a central double chamber operating in avalanche mode, with reading strips between the two gas gaps midway. Figure 18, on the left, shows a schematic representation.

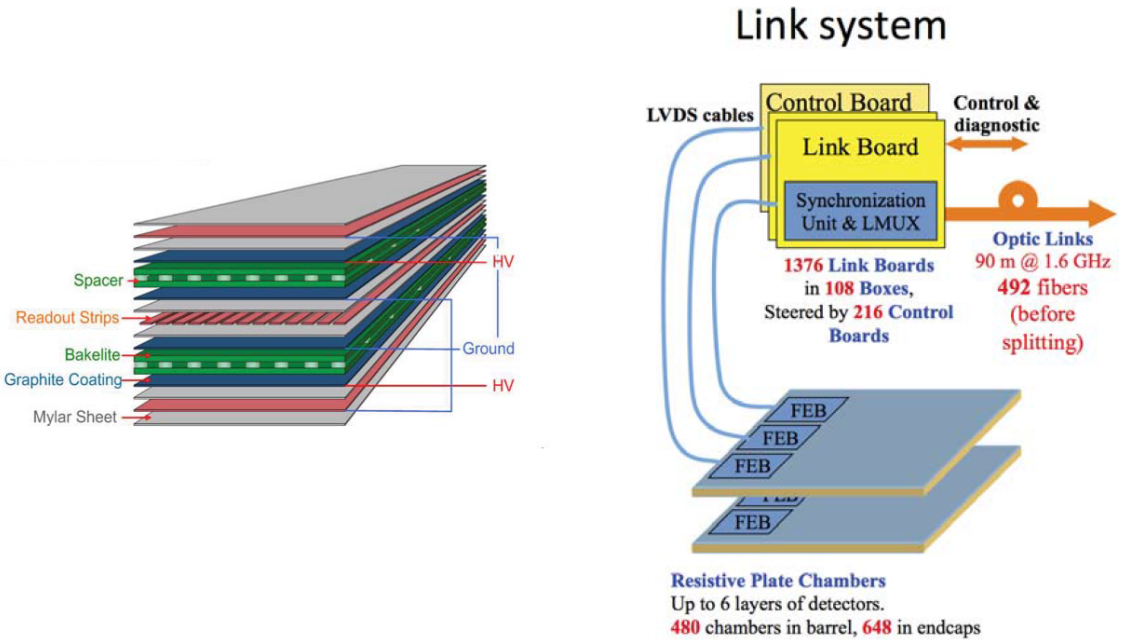


Figure 18 – Schematic view of RPC chambers double gap design (left). Schematic view of readout and connections to and from link and control boards (right). The numbers refer to the complete RPC system after the update.

Each two mm-thick gaps in the RPC comprises two high-pressure resistive electrode plates made of High Pressure Laminate (HPL), commonly called Bakelite, separated by a 2 mm distance, allowing gas to flow between them. A thin graphite layer coats the outer surface of the HPL plates, to which a high voltage is applied.

The reading plan is constructed using copper strips on a Mylar sheet, which is centrally positioned between the gap as the gas circulates. The strips are aligned at a

value of η, with variations between 2.28 and 4.10 cm in the barrel and 1.74 and 3.63 cm in the endcap, based on their location within the experiment.

The RPCs operate in avalanche mode, utilizing a gas mixture consisting of 95.2% Freon ($C_2H_2F_4$), 4.5% isobutane ($i - C_4H_{10}$), and 0.3% sulfur hexafluoride (SF_6), with relative humidity ranging from 40% to 50%.

Specific details can be found in Collaboration (COLLABORATION, 2013). The signals originating from the RPCs are asynchronously transmitted to the FEBs, which are located within the chambers. Here, the signals are amplified, discriminated, and digitized.

The LVDS signals from the FEB is conveyed through 10 to 20-m-long twisted pair cables to the link system near the CMS Detector. Figure 18 on the right displays a schematic view of the reading.

The link system acquires data from the FEB, synchronizes and compresses it, then transmits it via optical links to the trigger processors: the concentrator boards in the barrel Twin Multiplexer, data concentrator for muon barrel track finder (TwinMux), the Overlap region (barrel and endcap) Muon Track finder (OMTF) in the overlap region, and the Concentrator and Pre-Processor Fanout (CPPF) in the endcap.

The current Link System comprises two primary types of boards: Link Boards and Control Boards.

- **Link Boards**: Each LB contains 96 input channels, with one channel corresponding to a copper strip within the RPC. The initial step for the LB involves synchronizing and assigning the relevant collision signals received from the BX within a 25 ns time-frame. Subsequently, the data undergoes compression using a basic zero-suppression algorithm. Input channels are organized into 8-bit partitions, and only those with at least one non-zero bit are chosen for each BX. These non-empty partitions are time-multiplexed, meaning that if multiple partitions exist within a given BX, they will be transmitted one after another in consecutive BXs. Data from three adjacent LBs is consolidated by the central LB, which houses the optical transmitter responsible for sending the data to the Service Cavern (USC) via a fiber optic cable, achieving a transfer rate of up to 1.6 Gbp/s.

- **Control Boards**: The CB facilitate communication between the control software and the LBs through Front-End Controller (FEC)s and the Communication and Communication and Control Units (CCU) system. CBs are connected in ”token”

rings, with each ring comprising 12 CBs from a crate and a mezzanine plate FEC positioned on the Clock & Control System (CCS) plate, which is housed within a Versa Module Eurocard bus, ANSI/IEEE computer hardware bus standard (VME) box at theUSC. LB and CB boards are placed in vertical racks, specifically within crates. These crates adhere to the standard 6U height and feature 20 slots for two CBs and eighteen LBs. In addition, each crate contains a custom backplane for connecting camera cables, cables that supply power to the LBs and CBs, and FEB control cables that utilize an I^2C protocol via CB. Table 2 displays the quantity of link system components.

Table 2 – Component numbers of the current RPC link system.

Item name	Quantity
Link Boards	1376
Control Boards	216
Link boxes	108
Front planes	216
Optical links	492
12-channel fibers (from CBs to FECs)	18
TTC fibers	216

1.2.1 The Current RPC Link System

While RPC detectors possess an intrinsic time resolution of approximately 1.5 ns, as cited in paper (ABBRESCIA et al., 1999), the current link system records RPC information at a BX of 25 ns, resulting in a loss of resolution and, consequently, crucial detector data, this resolution deficiency in the current system's hardware is a weakness that will be addressed in the HL-LHC upgrade.

The existing RPC Link System, depicted in Figure 19, was manufactured between 2006 and 2007 and has been continuously operational since its installation in 2008.

The current system comprises custom Very-Large-Scale Integration (VLSI) and VME-specific boards containing a variety of devices, including both the LB and CB, which are based on Xilinx Spartan-3 FPGAs.

The Rack utilized in the current system adheres to the custom VME-specific and is designed to receive and transmit all signals to the RPC detectors. These detectors connect to the bottom of the Rack via flat cables. However, these connections may be vulnerable to communication failures under the new HL-LHC specifications in the upcoming years.

Furthermore, the current CB and LB feature radiation tolerance is based on an Actel ProAsicPlus FPGA. Nevertheless, no guarantees or studies confirm their ability to withstand the elevated radiation levels expected in future updates to the CMS Experiment.

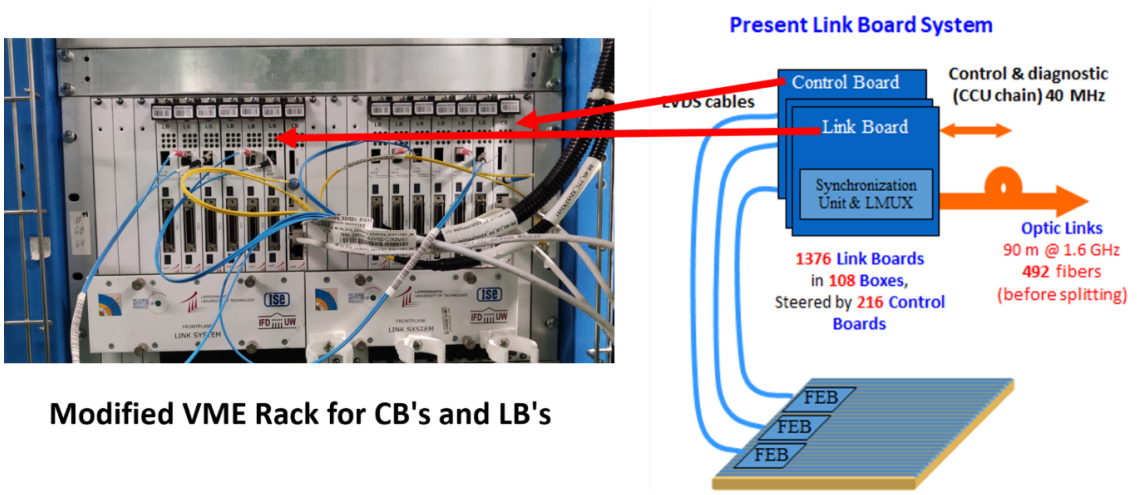


Figure 19 – On the left side, the current CB's and LB's installed in the VME Rack of the 3RE region of the barrel, and on the right side, the connection scheme of the RPC Link System.

The system, installed in LHC to LS2, performed well, causing only minor luminosity loss. However, operating at total luminosity post the HL-LHC upgrade presents potential inefficiencies.

These include the susceptibility of electronic devices using memories Static Random Access Memory (SRAM) and based FPGA to Single Event Upset (SEU) due to low ionizing radiation levels, issues detected by CMS Online Software (CMSOS) similar to SEU, and potential problems arising from an increase in electronics component's failures as the system ages.

Further challenges include the limited availability of replacement boards, especially the new RE4 boards, the scarcity of spare parts, and the potential for failures due to excessive temperature, current, or voltage.

Additionally, while the ASIC receiver (model TTCrx) it is building to designed to handle optical signals, its noise level sometimes exceeds the maximum allowable noise, requiring a Quartz Phase Locked Loop (QPLL) chip to reduce noise.

The TTCrx chip's internal issues in the current system can lead to output clock imbalance, decreasing RPC detector efficiency. Current CMSOS can correct this, but locking the corrupted data is complicated.

Lastly, the CCU25 chip imposes another constraint on the CB board, potentially leading to significant system loss if one CB fails. In addition, the current CB and LB firmware need to be updated, causing compatibility issues with new compilers and old chips.

2- Instrumentation and Methods

2.1- Introduction

When establishing an instrumentation system, a crucial initial step is to precisely determine the specific objectives of the measurements and understand the inherent complexities associated with the problem at hand. This involves crafting a realistic model that encompasses the specifications of all variables relevant to the instrumentation system, encompassing measurement, functionality, and implementation aspects.

To ensure successful development and achieve the intended outcomes, it becomes imperative to ascertain the acceptable tolerance levels for the system variables. This process involves carefully considering how to handle these variables to achieve the desired results proposed for the instrumentation system effectively.

The instrumentation system can be designed and implemented by defining the purposes, comprehending complexities, and setting acceptable tolerances to align with the project's objectives and yield meaningful outcomes.

Based on (GREGORY, 1981)'s statement, this work is defined in an instrumentation subsystem using devices based on high-speed data communication equipment such as a uTCA system, fast electronics using FPGA technology and advanced modules AMCs.

These modules are necessary to provide an acceptable solution within the resources and minimum conditions in time resolution to solve the integration problem between the different types of hardware involved in this development.

2.2- Design Requirements and Components

The assembly flow of the Slow Controller system in this project is a critical component in the control and regulation of processes involved due to the organized integration of software and hardware elements.

This formal technical exposition delineates the step-by-step assembly process described at CMS in the Technical Design Report (TDR) (COLLABORATION, C., 1997; EVANS; BRYANT, 2008; THE LHC EXPERIMENTS COMMITTEE, 2017) focusing on design requirements, component integration, reliability, no interference, and methodologies, summarized in Figure 20.

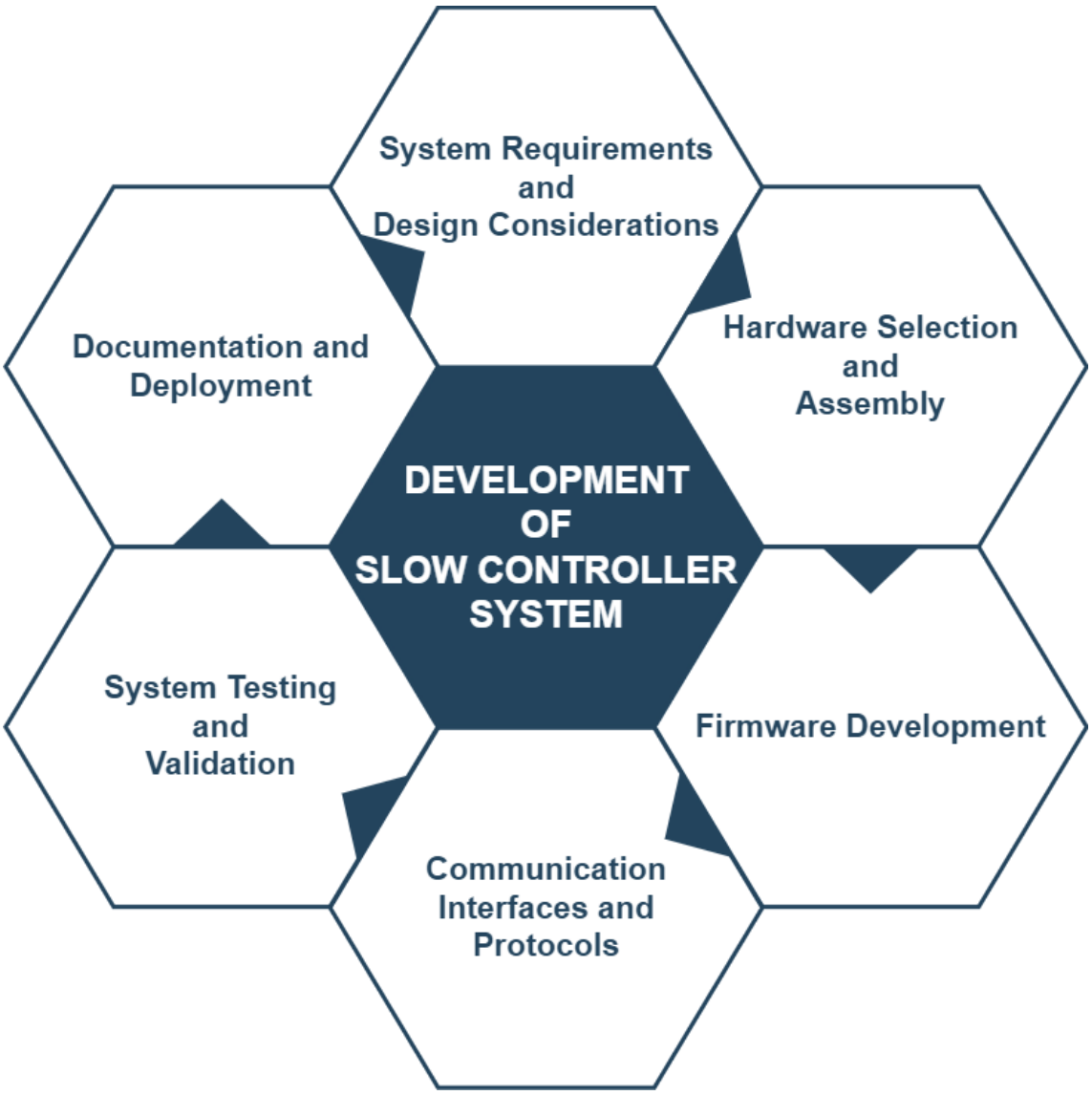


Figure 20 – Assembly flow of design requirements and components of the Slow Controller system.

The process begins with thoroughly analyzing system requirements to determine the desired performance characteristics, such as the control loop response, input/output (I/O) interfaces, and communication protocols. Based on this analysis, the System Requirements and Design Considerations are established.

Finally, all modules AMCs are selected due to the qualities of hardware and software components that adhere to the established specifications and conform to CMS and CERN standards.

The hardware assembly process involves integrating multiple components, such as AMC's modules, fiber optics, network hardware, and communication interfaces. The assembly begins with the uTCA crate configuration and all the items to connect, like cables and connectors for in/out accessing setup utilities.

The firmware, responsible for executing the control algorithm and handling I/O operations, was developed for CMS development users. The firmware pattern for the experimental setup follows the CMS standards. The firmware can then be flashed updated in FPGAs in case of change for new configurations or accept other functionalities for allowing the Slow Controller system to perform its designated design functions.

The Slow Controller system is often required to communicate with external devices or networks to exchange data, transmit control commands, or facilitate remote monitoring by RPC Software Online. Thus, appropriate communication interfaces (Universal Serial Bus (USB) 2.0 or Ethernet) and protocols (such as protocol to control/communicate with Ethernet-attached uTCA hardware (IPBus), Transmission Control Protocol (TCP)-Internet Protocol (IP), and others) are integrated into the system design. This integration ensures seamless communication between the Slow Controller system, other devices or networks, and PC Host Controller.

Another part that is not less critical is System Testing and Validation. Once the hardware and firmware components are assembled and integrated, the Slow Controller system undergoes rigorous testing to verify its functionality and performance. This phase includes functional testing, connection testing, and data transfer velocity.

Moreover, the last one, documentation and deployment, addresses comprehensive documentation detailing the system's design, assembly, and operation. This thesis about the Slow Controller system is the primary documentation for that.

2.3- System Description

The brief description of this advanced instrumentation system consists of telecommunications equipment and its advanced modules with last generation FPGAs that support large doses of radiation and have compatibility with integrated optical transceivers and high technology data transfer according to Figure 21.

The minimal hardware requirements necessary for assembling the system are defined to support high radiation conditions and default connections in HEP for efficient utilization.

In this advanced instrumentation system, the acquisition commands for the CBs and LBs will be controlled, as well as receiving monitoring data in the process of continuous surveillance of the variables of temperature, humidity, voltages of the FEBs, and diagnostics of detector muon acquisitions.

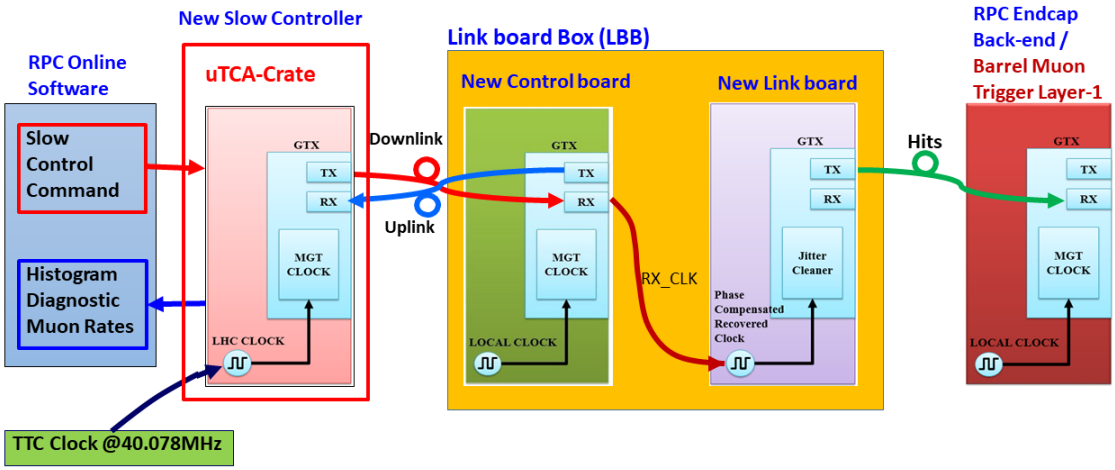


Figure 21 – Description of the Slow Controller in control between the high-level software and the hardware of the CBs and LBs for the new RPC Link System.

This system description presents all stages of the monitoring and control project implementation based on a fast electronics system using FPGAs, consolidated with advanced module AMC13 that cover all the necessary technology for the thesis.

Following a logical organization, according to Figure 21, in detail with its components, functions, and internal structure of each system involved that works together with the Slow Controller, each highlighted part can be summarized in:

- **RPC Online Software**: That toolbox is based on Generic Data Acquisition

Software Environment (XDAQ), a software platform designed to develop distributed data acquisition systems (BERTI et al., 2003) that ease designing, programming, and managing data acquisition applications by providing a simple, consistent, and integrated distributed programming environment for testing each hardware in the development at Slow Controller;

• **New Slow Controller**: Instrumentation subsystem using fast electronics based on FPGAs and advanced modules like AMCs for monitoring and control of CBs and LBs of the new RPC Link System, which is the central theme of this instrumentation work;

• **Link Board Box (LBB)**: New set that will replace the CBs and LBs seen and described in chapter 3, where the new boards of the new RPC Link System are based on racks/crates with specialized VME technology and are responsible for sending and receiving data and other signals at high speed from RPC detectors;

• **RPC End-cap Back-End/Barrel Muon Trigger Layer-1**: The new Slow Controller will contribute to this system specialized in data analysis offline environment of the CMS that will receive data at high speed with minimal error from the new RPC Link System;

The hardware and software resources that make up this innovative work, which only exists so far in this area, use state-of-the-art technology in instrumentation in the scientific area of large detectors in particle physics for the development of the Slow Controller.

2.4- Instrumentation Methodology

This section explains every stage involved in implementing this thesis, which centers on the design of an advanced instrumentation system. The system employs advanced modules reliant on radiation-resistant FPGAs, combined with optical links and high data transfer technology.

The thesis successively details the construction, assembly, and features of each virtual device, along with the hardware and software components that make up this work of scientific instrumentation.

The New Slow Controller's development methodology follows a two-step process. The first step involves modifying the primary hardware, AMC13XG, in an experimental setup. Then, these modifications undergo compatibility tests with the CB's requirements.

The Slow Controller's design operates on a two-layer approach, featuring an inter-

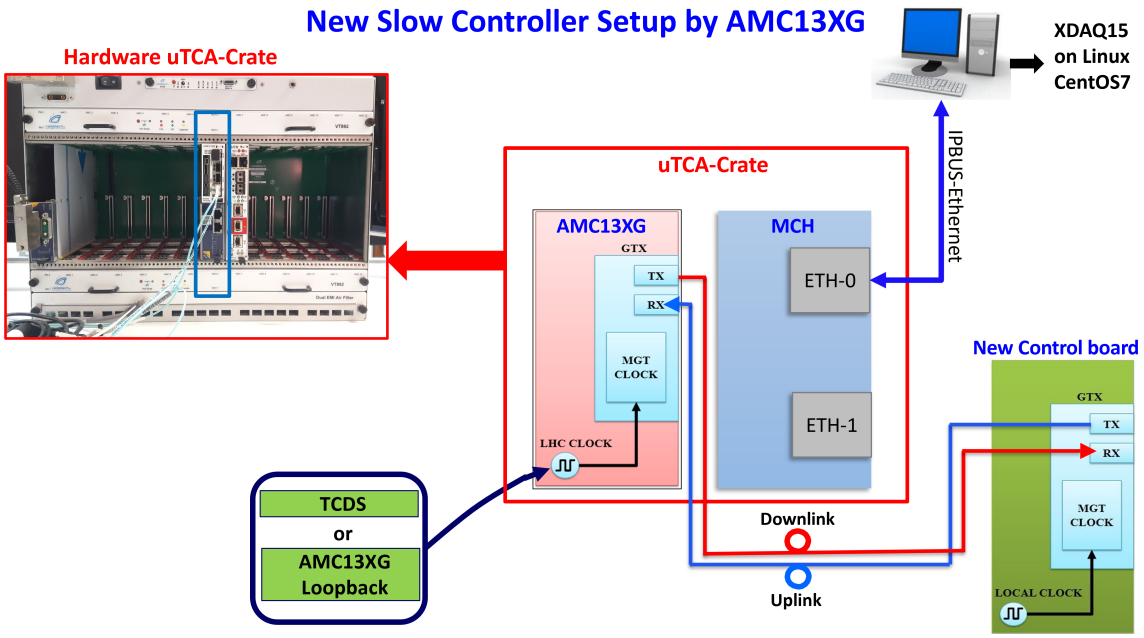


Figure 22 – Hardware configuration for the Slow Controller with the AMC13XG advanced module.

face library between the hardware and access libraries and a layer translating emulated commands from the CMSOS (LARREA et al., 2015, p. 4) into changes in the CB in Figure 22.

The TCDS, a comprehensive hardware system, communicates timing, control (synchronization) data, and status information with detector systems, dictating triggers for other subsystems.

In this Slow Controller, the AMC13XG module will emulate the TCDS shown schematically in Figure 23 hardware system's complex signals as part of the experimental project. The system will distribute key signals over the backplane from the uTCA to the XDAQ (HEGEMAN et al., 2015).

2.5- Materials of Systems and Subsystems

This chapter introduces the vital resources required to execute the thesis for this monitoring and control project, utilizing the uTCA Crate technology, as stated in item 2.4 throughout this document.

We will formally display the essential materials needed for developing this Slow

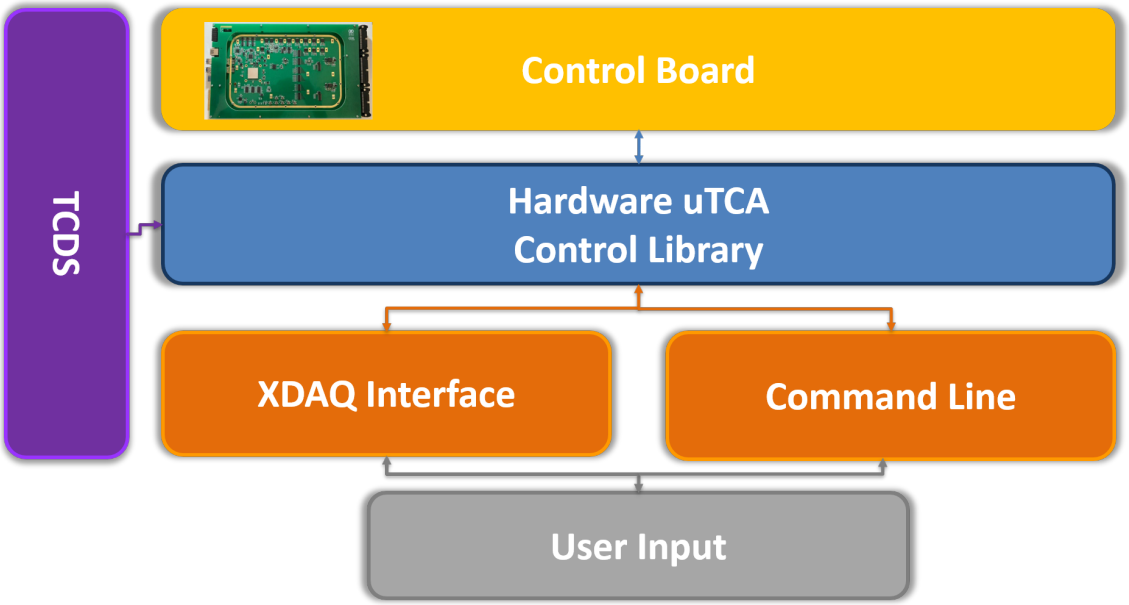


Figure 23 – Scheme of the Slow Controller: a user can send direct commands to the hardware using toolbox XDAQ or the command line. Adapted from (LEAL REYES, 2021).

Controller and its capabilities, detailed in this chapter. The descriptive presentation elucidates the conditioning of its configurations, setting the stage for understanding the environment's architecture and the effort put into its construction and assembly.

In this section 2.5, we will meticulously organize and present a series of materials for systems and subsystems. This organization offers a detailed view of its components, functions, and internal hardware and software structure, ready for deployment in this design.

2.5.1 The Control Board System

2.5.1.1 Overview of the Link System Project

For the context of Slow Controller development and concerning the CB provided by the Institute for Research in Fundamental Science (IPM) group, an overview of the link system project is formed with a general focus on this hardware.

Upgrading the link system only involves replacing LBs and CBs. The modified

standard VME racks will remain the same, as well as the backplane connectors, connector pins, and all RPC and FEB signals from the current system.

The technology of the new CB allows absorbing functionalities of the majority of the chips of the current system in use as CCU, TTC receiver ASIC (TTCrx), Gigabit Optical Link (GOL), and QPLL ASICs, replacing them with FPGAs, which in addition, these FPGAs can be reprogrammed for future updates in the algorithms.

The CB needs to communicate with the Monitoring and Execution Control System, Run Control and Monitor System (RCMS) in the USC control room over its optical links.

The Slow Controller is the primary means of communication between the new set of hardware developed in this update for the HL-LHC and the RCMS room in the USC room, where the CB will be responsible for sending and receiving these commands via RPC Online Software.

2.5.1.2 Architecture

The CB architecture is highly complex. However, the main tasks related to the Slow Controller system are highlighted.

The Small Form-Factor Pluggable Plus (SFP+)0 and SFP+1 connection provides a robust bi-directional communication channel against noise and interference based on the GBT link protocol for diagnostics, slow control command, and Timing and Trigger Control (TTC) clock, as well as providing a secure channel for remote programming of FPGAs.

The Ethernet connection accesses the internal settings via PC, and the USB JTAG/Universal Asynchronous Receiver/Transmitter (UART) connection provides the communication port for recording and rewriting firmware on the CB. This architecture can be shown in Figure 24.

The internal control from CB has a state machine for reading and writing the First In, First Out (FIFO) buffers and Sending and receiving data to/from the external interfaces.

The external interfaces, SFP+ and GBT-FPGA, transmit and receive commands and data to and from the CB. The bitstream FIFO buffers handle the reception of FPGA configuration bits from the LB or CB through the external interface.

The CB even has a mode of use by Desktop in a way where it uses the Card buffer

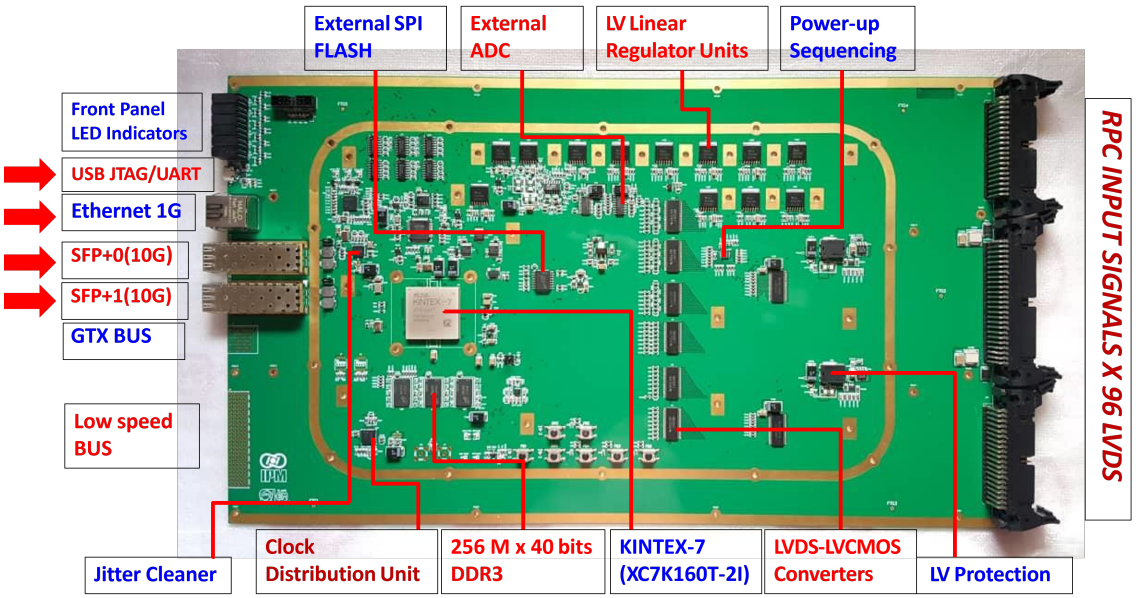


Figure 24 – Control board architecture for VME pattern modified.

to store incoming Commands from the PC interface, and card to Desktop FIFO to keep the result and status of the execution of commands or the returned diagnostics data from the Control Board by GBT-FPGA data transceiver.

Figure 25 shows the bitstream FIFO in the hardware architecture for use in data format commands at CB.

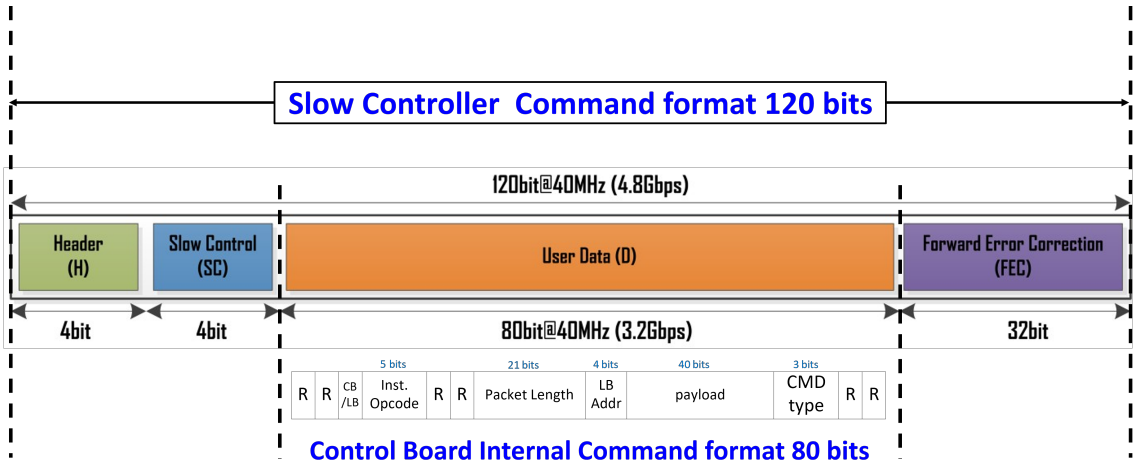


Figure 25 – The Architecture of bits in use at the Control Board for Slow Controller.

Based on KC705 Evaluation Board Kintex-7 XC7K325T FPGA, the CB uses the structure State machine handles FIFO buffers and GBTx data transmission engine.

This GBT-FPGA is set to the fixed latency, and the Data frame format is 80-bit internal at CB, with fixed latency at the data rate of 3.2 Gigabyte per second (GB/s), and

the Data frame format is 120-bit, with fixed latency at the data rate of 4.8 GB/s for Slow Controller.

2.5.1.3 Command Types

The CB has numerous features and commands as standard. For this thesis's development, the Slow Controller's primary function is to operate the three main types of external interface commands.

These three types of commands are highlighted for the Slow Controller, and they are the standard general form commands received by the Kintex-7 XC7K325T FPGA firmware. These commands can be defined in Type-A, Type-B, and Type-C.

- Type-A: Link Board/ Control Board read/write parameters;
- Type-B: Physics, Test, Calibration, Diagnostic, and Monitoring;
- Type-C: Remote Programming of the Link Boards and Control Board;

The Slow Controller selects which type of external command will be used in the request, together with the internal command block or not in this request.

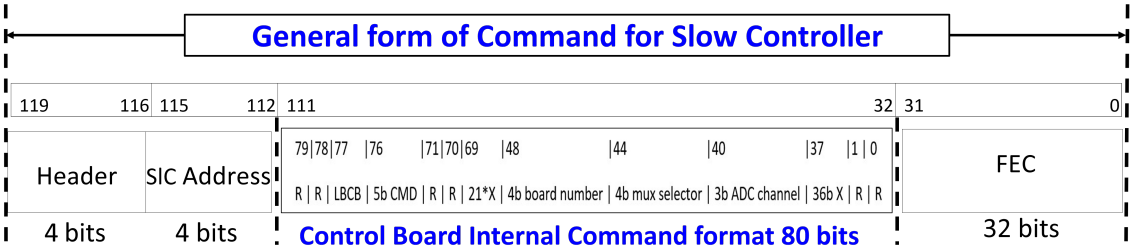


Figure 26 – All data frame formats with 120-bit to access by a Slow Controller System using XDAQ.

The general form of commands for three types, A, B, and C, is shown in Figure 26 in the mode used by Slow Controller implemented with XDAQ toolbox.

- Commands Type-A:

For the Type-A command request, some flexible commands of this type were defined, implemented, and tested to send an 80-bit command frame and read back only an 80-bit frame response in the internal controls.

In development, the Slow controller test sends an inner 80-bit command frame with

an outer 40-bit frame and reads back all 120-bit frames.

Type-A commands monitor ADC temperature, FPGA Core Voltages and total board Current, Link Board and Control Board Hardware Identification (ID)s, Link Board and Control Board FLASH IDs, FEB ADC temperature, ADC/Digital-to-analog converter (DAC), channels Time-to-Digital Converter (TDC), synchronization window, Phas,e and Latency compensation;

On each of the control boards and link boards, several operating parameters can be periodically read and monitored by the Slow Controller:

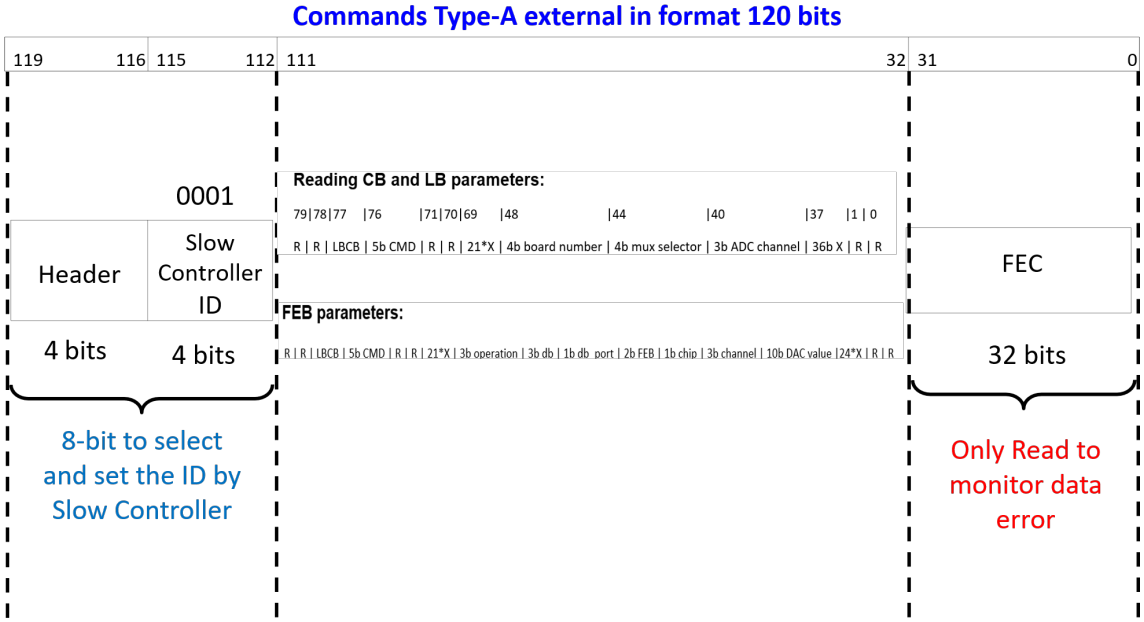


Figure 27 – Commands Type-A for reading/set for Control Board, Link Board and FEB parameters.

• Commands Type-B:

The same technique is used for Type-B commands to send a 120-bit command frame, i.e., 80-bit inner and 40 outers, and read back a packet with a 120-bit frame.

The 120-bit response frame holds general CB and LB diagnostics readout data and monitoring and data for analysis in Histograms or dashboards.

Figure 28 shows an example of a frame with a 120-bit response for this command type.

• Commands Type-C:

In particular, Type-C controls are used for remote programming via Optical GBT Link. This external command type aims to keep the GBT-frame at a fixed latency, not to

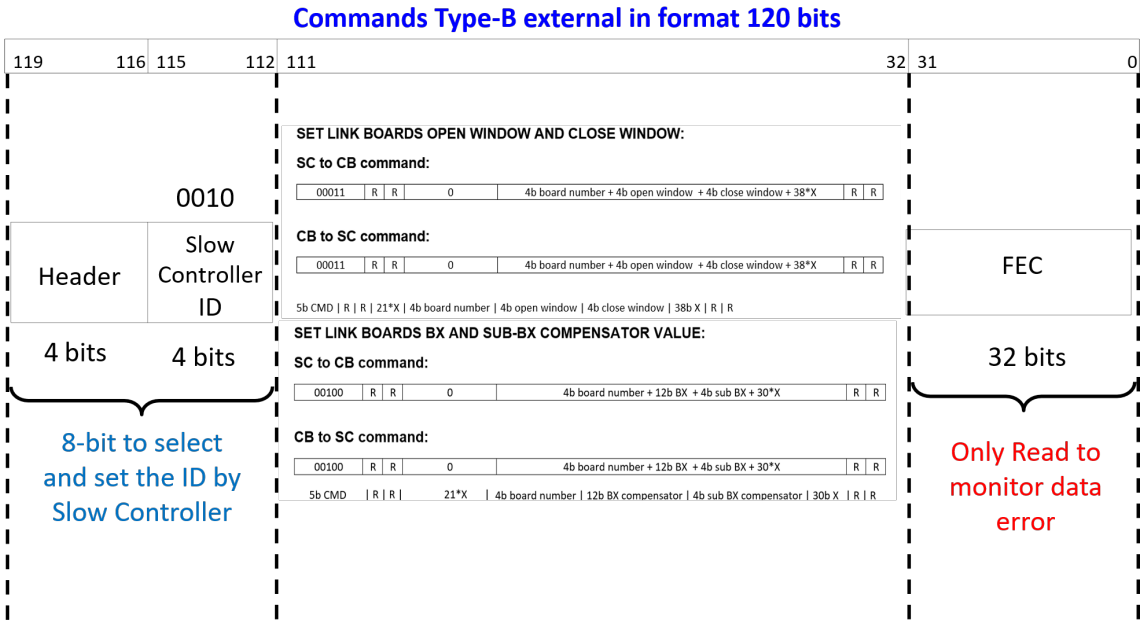


Figure 28 – Type-B commands for selecting LB and reading diagnostic data.

generate errors in the firmware recordings in the FPGA.

Link cards and control cards have 128 Megabit Quad Serial Peripheral Interface (QSPI) Flash memory for Golden Bitstream FPGA configuration and Bitstream updates.

The main idea for remote programming is: secure data transmission using GBT-frame encoding protocol in fixed latency mode and Cyclic Redundancy Check (CRC) checks, with system stability after reloading and power on and off, ensuring automatic system restart, offering an ability to revert to golden Bitstream in case of any remote reload failure type.

Figure 29 shows the 120-bit frame required for this type of command and data stream for Golden Bitstream FPGA configuration and Bitstream updates.

2.5.2 The Hardware uTCA Crate

The uTCA computing architecture standard is used in the area of HEP experiments (ZHANG; LI; JIANG, 2019). In CMS updates since 2012, many existing VME-based electronic subsystems have begun to be replaced with a new uTCA or Advanced Telecommunications Computing Architecture (ATCA) standard.

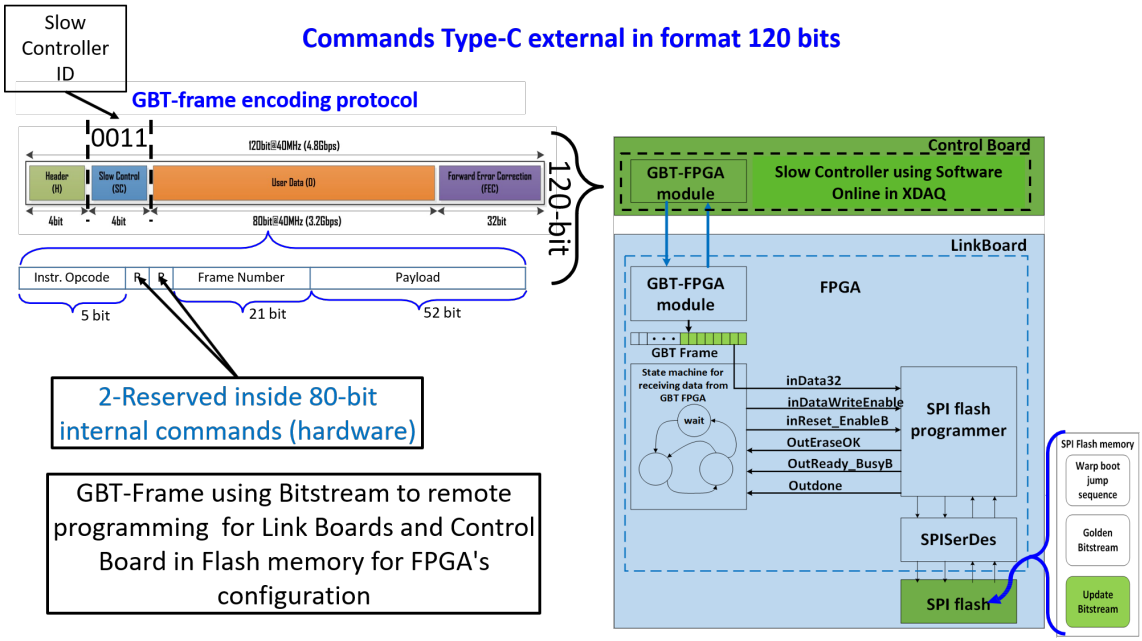


Figure 29 – Type-C commands in GBT-frame seen in fixed latency stream for remote programming on FPGA.

This Slow Controller project maintains this effort in the standardization integrated by the CMS, using VadaTech's uTCA Crate model VT892, seen in Figure 30. This uTCA offers a flexible, high-density, high-performance backplane based on the current serial standards (ILES et al., 2011).

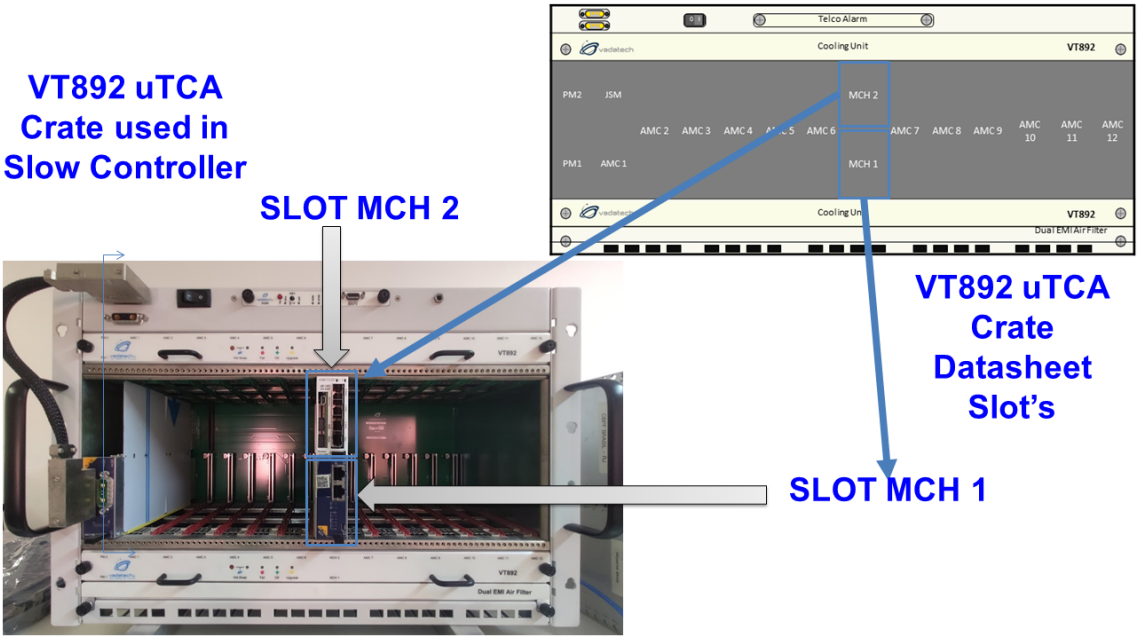


Figure 30 – VadaTech's uTCA Crate model VT892.

The uTCA is based on the AMC standard and holds up to 12 modules that can

be inserted directly into your backplane. In addition, a uTCA Crate provides connectivity between two, especially the First MicroTCA Carrier Hub (MCH1) and Second MicroTCA Carrier Hub (MCH2) bus slots in your backplane, although direct connections between different places are also allowed.

As with CMS, this project used the slot MCH1 feature in a ready-to-use module that provides management and communication functions described in the uTCA and AMC standards, while the slot MCH2 is customized to provide LHC timing, and DAQ functions specific to CMS requirements.

This slot MCH2 is called the 13th module, which has described its use and optimization of hardware, firmware, and software in chapter 2.5.2.2.

The uTCA-based design centralizes all complex logic in Slow Controller development for this new subsystem. In addition, it maximizes assembly due to reusing components, including hardware, firmware, and software.

For the Slow Controller's specific configuration, uTCA uses the physical ports standardized by the CMS and thus reuses all its electronics, software, and firmware updates. Figure 31 shows the default ports used in the uTCA for the CMS.

Fabric	AMC Port	MCH	AMC13	Category	MCH Finger	CMS Use
A	0	Yes		Common Options	1	GbE
	1		Yes			DAQ
B	2	Yes			2	Spare
	3		Yes			Fast controls (TTC)
Clock	TCLKA	CLK1/2		Clocks		Spare
	FCLKA		CLK1/2			LHC Clock
D-G	4-7	Yes		Fat Pipes	3, 4	User
	8-11		Yes[2]			
H-K	12-15			Extended Fat Pipes		
	16-19					

AMC Ports critical features for the configuration in use

AMC Ports extended

Figure 31 – Settings for AMC and physical ports at modules uTCA.

Fabric A (Port 0) is used for Gigabit Ethernet (GbE) communication with all modules provided by a GbE switch on the commercial MicroTCA Carrier Hub (MCH). In addition, a cross-over GbE connection also provides service to the AMC13 in the MCH2 site.

Fabric A (Port 1) already collects the DAQ link and will be operated at a multiple of the 125 Megahertz (MHz) reference clock (2.5, 3.125, 5.0 GB/s) in the MCH2 reference firmware. This project designer is using by pattern to use the LHC clock as a reference for port one default.

Fabric B (Port 3) is used to transmit timing and control signals from AMC13 to the AMC cards using the TTC protocol. The return channel from AMC to AMC13 currently needs to be used.

One of the advantages of the uTCA dual-star backplane is Port FCLKA in MCH2 (CLK1) carries the LHC machine clock (40.079 MHz) from the MCH2 (AMC13) to the AMC modules.

Fat pipes fabrics D-G are routed to the T3/T4 connectors of the MCH2, but the standard slot MCH2 (AMC13) does not connect to these tongues. The backplane links are connected to 12 point-to-point links for AMC Modules.

MCH is a Commercial module standard in slot MCH1, and MCH2 (AMC13) is a custom design for Slow Controller, according to signals, firmware, and software defined by patterns CMS.

As an advantage, the module connected in slot MCH2 is Bi-directional serial up to 10 GB/s point-to-point with links redundant to each AMC used at uTCA crate connected by your backplane.

These internal connections between bus MCH1 and MCH2 connected in the backplane can be shown in Figure 32.

In this development, AMC modules compatible with VadaTech's uTCA Crate model VT892 are used to control the New Link System via CB.

This set of generic telecommunications hardware, adequately configured with a type of DAQ, in this case, XDAQ, in HEP experiments, is the fundamental basis of this project.

Through this kind of technology, the Slow controller can provide additional clock signals, clock, trigger function, and user-specific I/O at the back-end with high-speed transceivers that can be used to extend the connectivity and bandwidth of other modules supported in future expansions and updates.

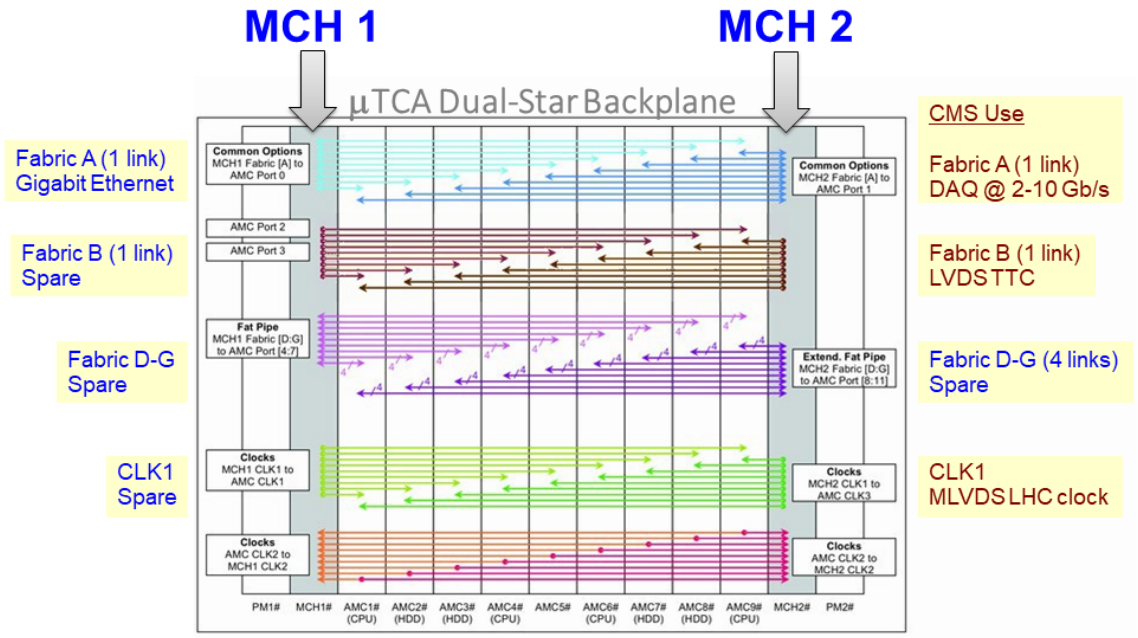


Figure 32 – The uTCA dual-star backplane is connected by MCH and MCH2 bus.

2.5.2.1 The MCH1: NAT-MCH module

The MCH Module, technically called the uTCA Carrier Hub (GMBH, 2023), is the central switching and data management mechanism for all uTCA systems.

In addition, the double-width MicroTCA Carrier Hub (NAT-MCH) is a commercial module and is designed to work on all uTCA specifications, operating up to a maximum of 13 AMC's.

Due to its scalable and flexible design, one of the main features of this NAT-MCH module is that it can also provide a centralized clock distribution for all AMCs in the system.

This hardware can be used in any uTCA system, supporting telecommunications and non-telecommunications environments and redundant and non-redundant architectures.

The base module has a Central Processing Unit (CPU) and memory on its operating baseboard, where it is equipped with a CPU from the Freescale ColdFire family of processors at a core frequency of 266 MHz operating with a real-time kernel. The internal block diagram of the CPU in the NAT-MCH and its front view is seen in Figure 33.

Another feature explored in this work is the ability to switch data between the backplane and the AMC's modules, functioning as a data concentrator hub for the various

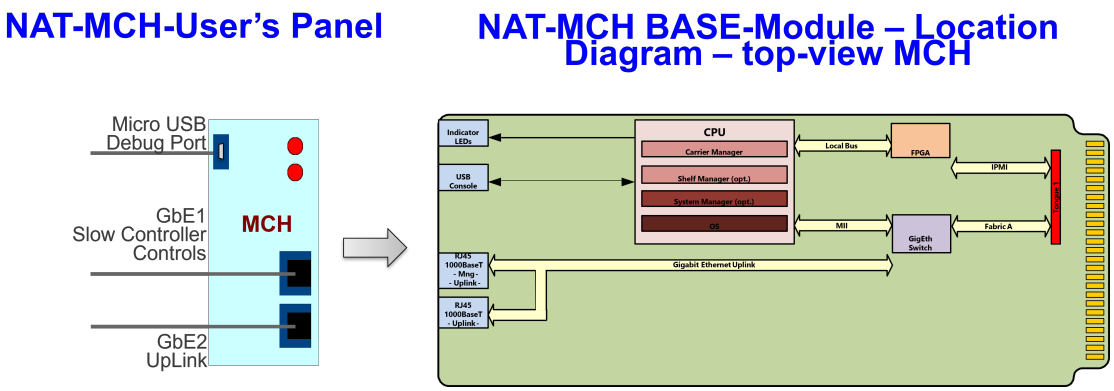


Figure 33 – Default NAT-MCH BASE using in Slow Controller with NAT-MCH-User's view Panel.

fabrics of the system, as defined for the AMC standard series and its other 1 GbE port.

One of the embedded technologies explored in The NAT-MCH is the ability to communicate between hosts in different physical address domains, acting with a router for all AMC modules connected in the uTCA, producing a gateway performing a Network Address Translator (NAT) via the backplane.

The NAT-MCH hardware configured as a router translates from an address of its subnet and provides transparent IP-level access to each AMC device connected to the uTCA to a global IP address from a host created by the NAT-MCH, which has its address acting as a gateway.

With this, NAT-MCH allows clients on the Desktop on another GbE Ethernet connection to access data communication, control, and monitoring services through their internal subnet routed to the external host. Figure 34 shows the NAT subnet created by the NAT-MCH and its connection with the Desktop for the Slow Controller.

On its other GbE ethernet connection, the desktop computer can access this subnet for project development. So each NAT-MCH receives a dynamic or static IP address for access at the specific IP address or port in the datagram data stream.

In the Slow Controller, this IP datagram routing technology is used between AMC modules that solve the general problem of providing IP-level access between hosts connected to the NAT-MCH in a Crate uTCA for modules without requiring each host in the backplane to have a specific configuration valid globally on the central system.

Another feature explored in this work is the ability to switch data between the backplane and the AMC's modules, functioning as a data concentrator hub for the various fabrics of the system, as defined for the AMC standard series and its other external GbE

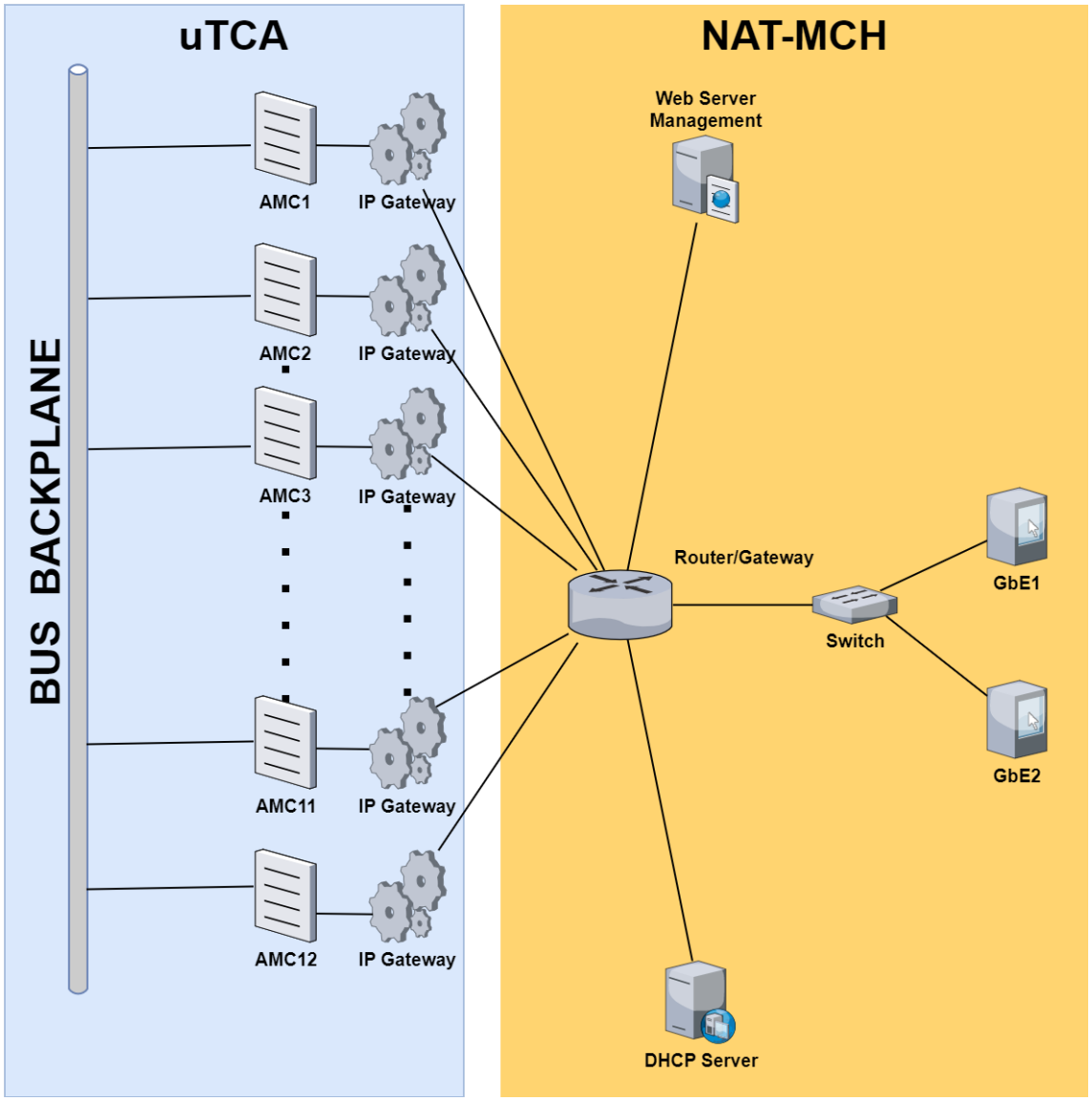


Figure 34 – The NAT-MCH module functions as a router for the IP header datagram.

ethernet connection port.

2.5.2.1.1- The IPBus

IPBus is a communication protocol used in the field of HEP for the exchange of data between hardware devices such as FPGA, Digital Signal Processors (DSPs), and microprocessors. The protocol uses a packet-based communication model to enable reliable data transfer over various interfaces.

The IPBus protocol is a control protocol designed for use with the ATCA/uTCA standards. It is based on the IP and provides a standardized way for controlling and communicating with devices in a distributed system (LARREA et al., 2015).

IPBus allows for the easy configuration, monitoring, and control of multiple devices within a system using a single network connection. It uses a simple packet structure and a set of predefined commands to communicate between devices, making it easy to implement and use.

One of the key benefits of the IPBus protocol is its flexibility. It can be used with a wide range of devices, from simple sensors to complex control systems, and can be adapted to meet the needs of specific applications, and this project is used in developing the Slow Controller.

Overall, the IPBus protocol is a powerful tool for controlling and communicating with devices in a distributed system, making it an essential component for uTCA standard in the Slow Controller.

In physics experiments, the IPBus protocol is commonly used to control and communicate with distributed systems devices. The IPBus packet structure is shown in Figure 35.

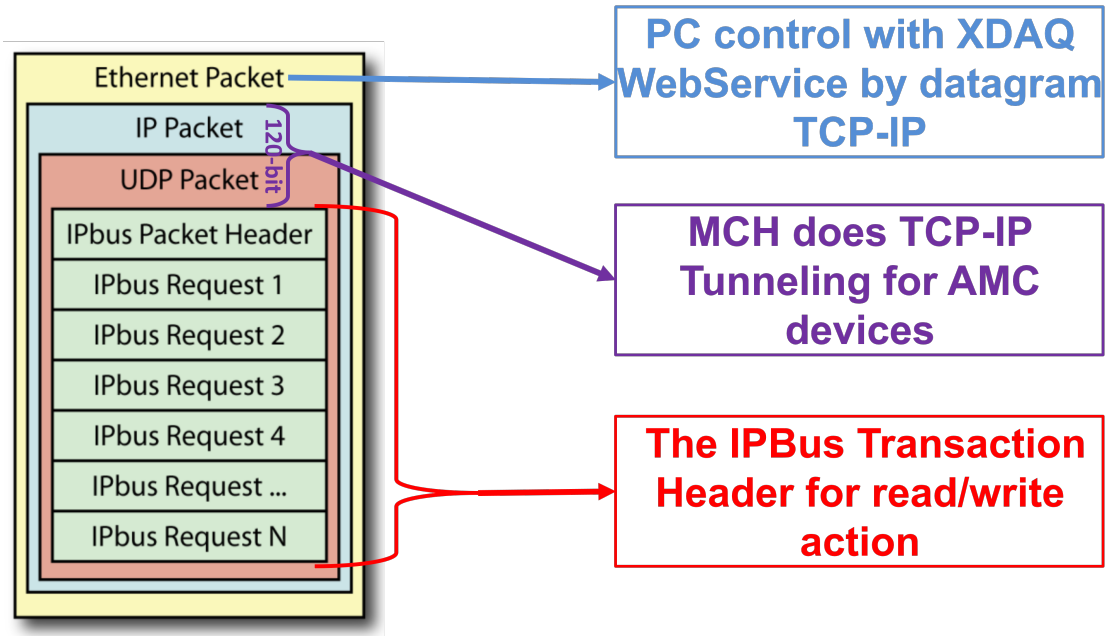


Figure 35 – The transaction or set of transactions must fit into a single UDP packet in a TCP-IP datagram.

The packet structure consists of a header and a payload. The header contains

information such as the packet length, address, and command, while the payload includes the transmitted data. The payload can be of varying lengths and contain configuration settings, status information, and measurement data.

One of the critical challenges in designing Slow Controller systems is ensuring reliable communication, mainly when multiple packets are in flight simultaneously (FRAZIER, Robert et al., 2012). To address this challenge, the IPBus protocol incorporates a reliability mechanism that ensures packets' correct transmission and reception.

The reliability mechanism in IPBus is based on sequence numbers and acknowledgments. Each packet is assigned a sequence number incremented for each subsequent packet sent. The receiver then sends an Acknowledgment (ACK) message back to the sender to confirm the successful reception of the packet. If an ACK is not received within a specific time, the sender will retransmit the packet.

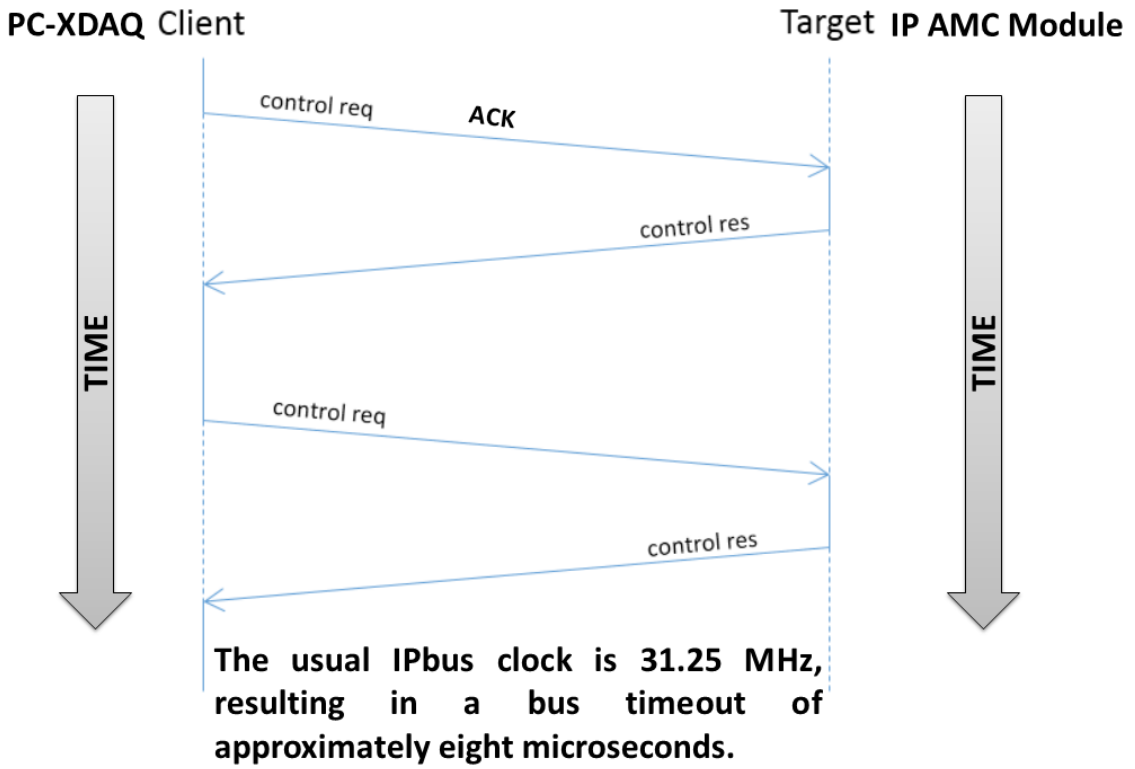


Figure 36 – A bus timeout of approximately eight microseconds in ACK until the end request.

Figure 36 shows the ACK request in the simplest use case when the transport protocol does not fail and there is only one packet in flight. In this case, only control packet requests and responses are sent over the network.

To further improve reliability, IPBus also includes a timeout mechanism. If an ACK

is not received within a predetermined time, the sender will assume the packet was lost and retransmit it. The timeout value is typically set based on the network latency and can be adjusted to optimize performance.

Overall, the reliability mechanism in IPBus is a critical component for ensuring the correct transmission and reception of packets in a distributed system. By incorporating sequence numbers, acknowledgments, and a timeout (COLLABORATION, T. C., 2011; COMMITTEE, 2011).

The IPBus protocol defines four types of packets: the IPBus Packet Header, Control Packet, Status Packet, and Re-send Packet. Each of these packets serves a different purpose in the protocol and has its unique structure and set of fields.

The Packet Header contains information about the packet, such as its length, type, and destination. Next, the Control Packet sends commands and requests from a client to a server, such as read and write requests. Next, the Status Packet is used to respond to a Control Packet, indicating the success or failure of the request. Finally, the Re-send Packet requests that a packet be re-sent if it was not received the first time correctly.

Understanding the technical details of each packet is essential for developing software and hardware that communicate using the IPBus protocol in Slow Controller. Proper protocol implementation ensures reliable data transfer and efficient communication between devices in HEP systems and can be accessed in the technical report (FRAZIER et al., 2013).

2.5.2.2 The MCH2: AMC13XG module

With particular functions, this 13th module, technically called AMC13, can distribute LHC clock and timing functions to implement controls to other AMCs and also collects DAQ data from AMCs connected to the uTCA bus.

The third version of this module is used, which is called AMC13XG, whose main modifications concerning the previous version are the remodeling of internal Tongue 1 to accept 10 GB/s optical links.

The AMC13XG is a powerful and versatile AMC designed to provide various applications with high-speed data processing and communication capabilities. For example,

the AMC13XG is used in high-energy physics research (HAZEN et al., 2013), where it is used to interface with detectors and provide data transfer capabilities between different parts of the experiment, in this specific case, between the RPC Link System and the Slow Controller for the CMS Experiment.

The AMC13XG hardware consists of a compact, low-power board that fits into a mezzanine in an MCH2 slot on a carrier board of uTCA Crate. The board is equipped with a powerful FPGA Kintex 7 XC7K325T and an FPGA Spartan 6 XC6SLX45T, which can be programmed to perform a wide range of data processing tasks.

The FPGA Kintex 7 is connected to 3 SFP+ high-speed serial interfaces for DAQ0, DAQ1, and DAQ2, providing data transfer rates of up to 10 GB/s to interface with external devices, such as detectors and data acquisition systems. In addition, the board has multiple front-panel connectors, one of them an SFP+, connected to FPGA Spartan 6 working TTC synchronization for the LHC clock. Figure 37 shows block diagram AMC13XG with tongue board sections.

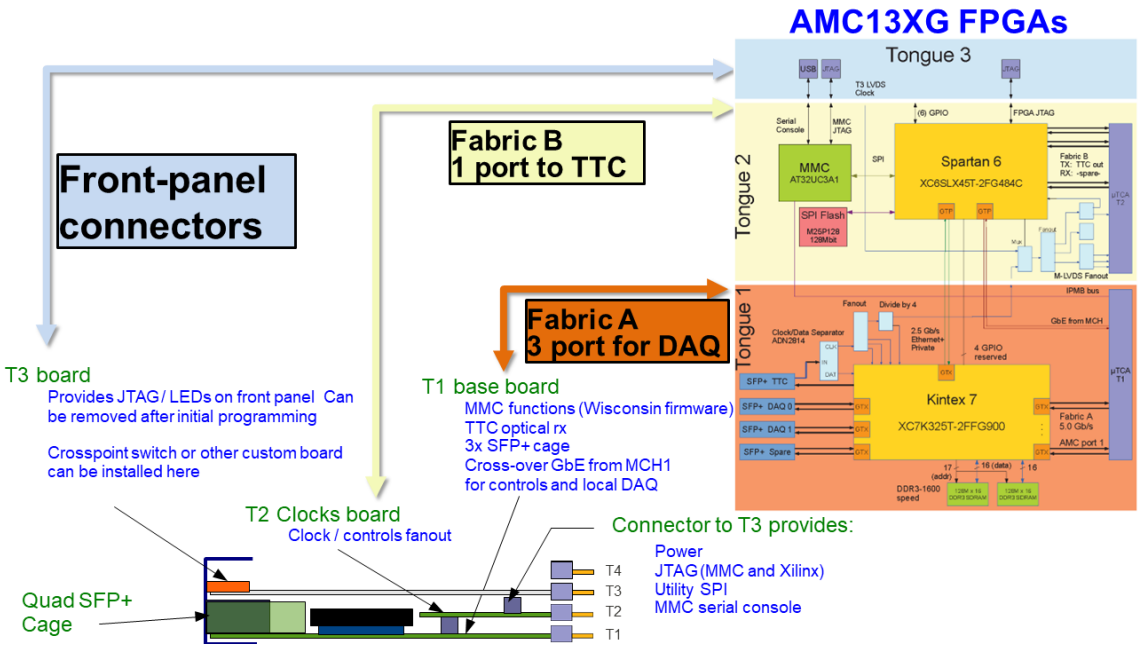


Figure 37 – AMC13XG block diagram with Board Stack configuration only tongues 1, 2, and 3.

For the Slow Controller, the two buses, MCH1 and MCH2, are configured in dual-star topology in the uTCA crate. This topology of a specific backplane dual-star was chosen for use in the CMS as a standard to reinforce the hardware specifications in this project with AMC13XG.

Another critical function implemented automatically in the AMC13XG is the CRC, which ensures that data transmitted through the TTC and Trigger Throttling System (TTS) is error-free.

The TTC system is a crucial component of the TCDS, responsible for distributing timing and trigger signals to the various subsystems. The AMC13XG provides a crucial link between the TTC and the TCDS by serving as a TTC receiver and a TTS transmitter and sharing these signals with of uTCA backplane.

One of the critical functions of the AMC13XG is to perform CRC computations to ensure data integrity. The schematic shown in Figure 38 describes the data flux started in request from run XDAQ, crossing IPBus transaction and finalized process in AMC13XG.

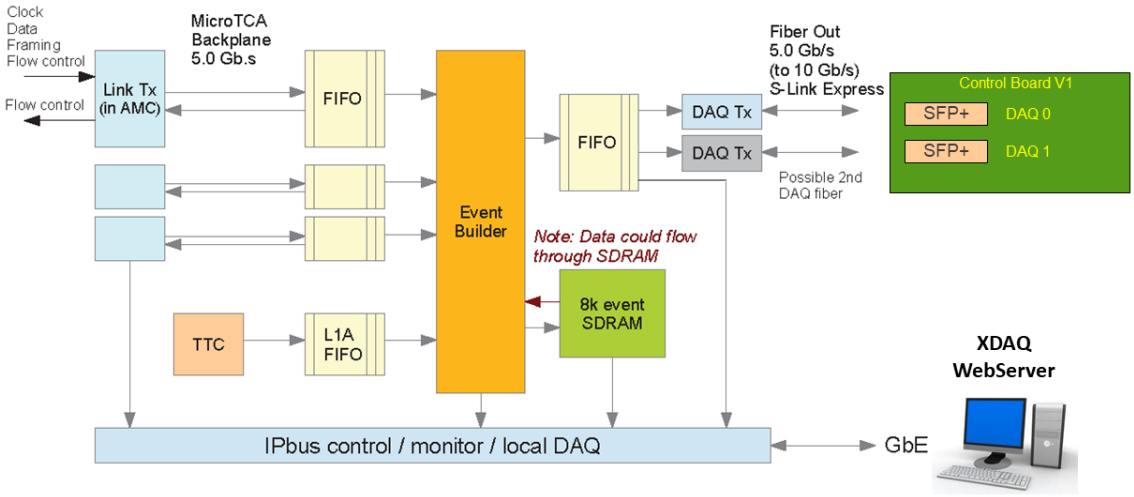


Figure 38 – Block diagram of the AMC13XG operating as a central data hub on the Slow Controller.

Another critical function implemented automatically in the AMC13XG is the CRC, which ensures that data transmitted through the TTC and TTS is error-free.

The CRC computation is performed on each data packet received by the AMC13XG, and the resulting checksum is compared to the one received with the packet. If the checksums match, the data is deemed error-free and forwarded to the TCDS. The data is discarded if the checksums do not match and the AMC13XG requests retransmission.

Overall, the AMC13XG plays a critical role in ensuring the integrity of the data transmitted through the TTC and TCDS systems. It also keeps integrity for data communication between IPBus protocol encapsulates in TCP-IP datagram.

In addition, the AMC13XG is designed to be highly flexible, allowing users to modify and customize its firmware to meet their specific needs. For hardware safety, the firmware

is the version default.

2.5.3 Software Environment

CentOS 7 (ALIBI; ROY, 2016) is a widespread open-source distribution based on Red Hat Enterprise Linux sources. This operating system is widely used in server environments for its stability, security, and reliability. All of the processes necessary for this project run in CentOS 7 libraries. The main ones are the Inter-process communication (IPC) mechanism for IPBus, eXtensible Markup Language (XML) files, and compiling C++ files.

The Libraries used in CentOS 7 come with a rich set of libraries that can be used to develop and run various applications. For example, libraries used for the Slow Controller are collections of precompiled routines that a program can use. Figure 39 highlights the environment embedded in CentOS 7 Linux using libraries to run the internal process at kernel Linux.

CentOS 7 utilizes various libraries, including the glibc from the GNU Project's implementation of the C standard library (GNU) Project and the GNU Standard C++ Library, contributing essential C and C++ functionalities. It also uses the Boost libraries for advanced C++ capabilities.

The environment supports IPBus, an efficient IPC mechanism that enhances communication among processes. XML files are another key resource, storing and transporting data in a format accessible to programmers and machines.

C++ developers can use libraries such as TinyXML, Xerces-C++, or PugiXML for parsing and manipulating XML files. These resources contribute to CentOS 7's robust environment, allowing for the developing of high-performance applications for Linux-based systems.

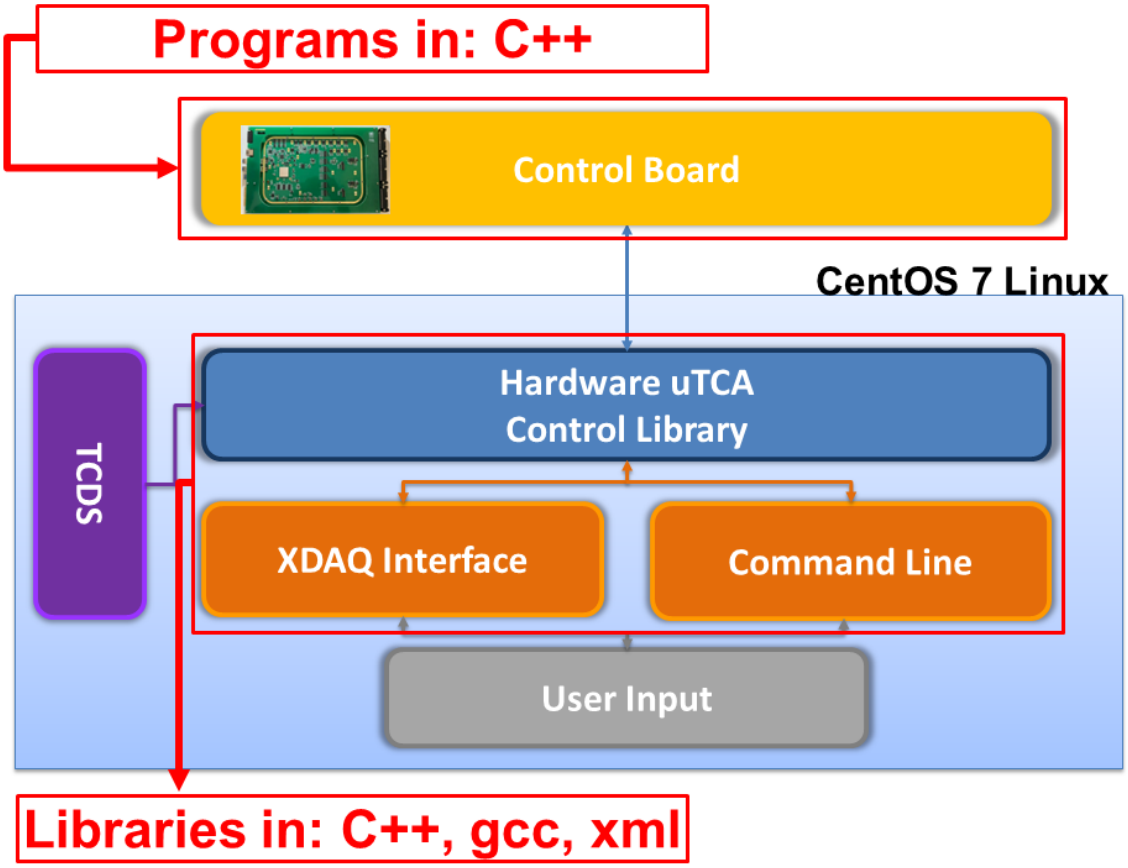


Figure 39 – The block diagram for the CentOS 7 Linux control internal process.

2.5.3.1 XDAQ Toolbox

The CERN CMS XDAQ Toolbox (BERRIAUD, 2022), an advanced software framework, plays a crucial role in XDAQ for the CMS Experiment at the LHC. This significant contribution extends to the Slow Controller project.

The architecture of the XDAQ Toolbox is distinguished by its distributed, component-based nature, promoting flexibility, scalability, and maintainability. It is organized into three layers: hardware, software, and network, each playing unique roles in efficient and reliable data acquisition.

Within the XDAQ Toolbox, several key software components ensure proficient functioning, including the XDAQ executive, XDAQ Applications, XDAQ Services, and the HyperDAQ.

Each component plays a crucial role in supervising other components, implementing specific functionalities, providing generic services, or offering a user-friendly graphical

interface for system control and monitoring.

The HyperDAQ in Figure 40, a web-based interface, stands out, providing flexibility through support for various communication protocols.

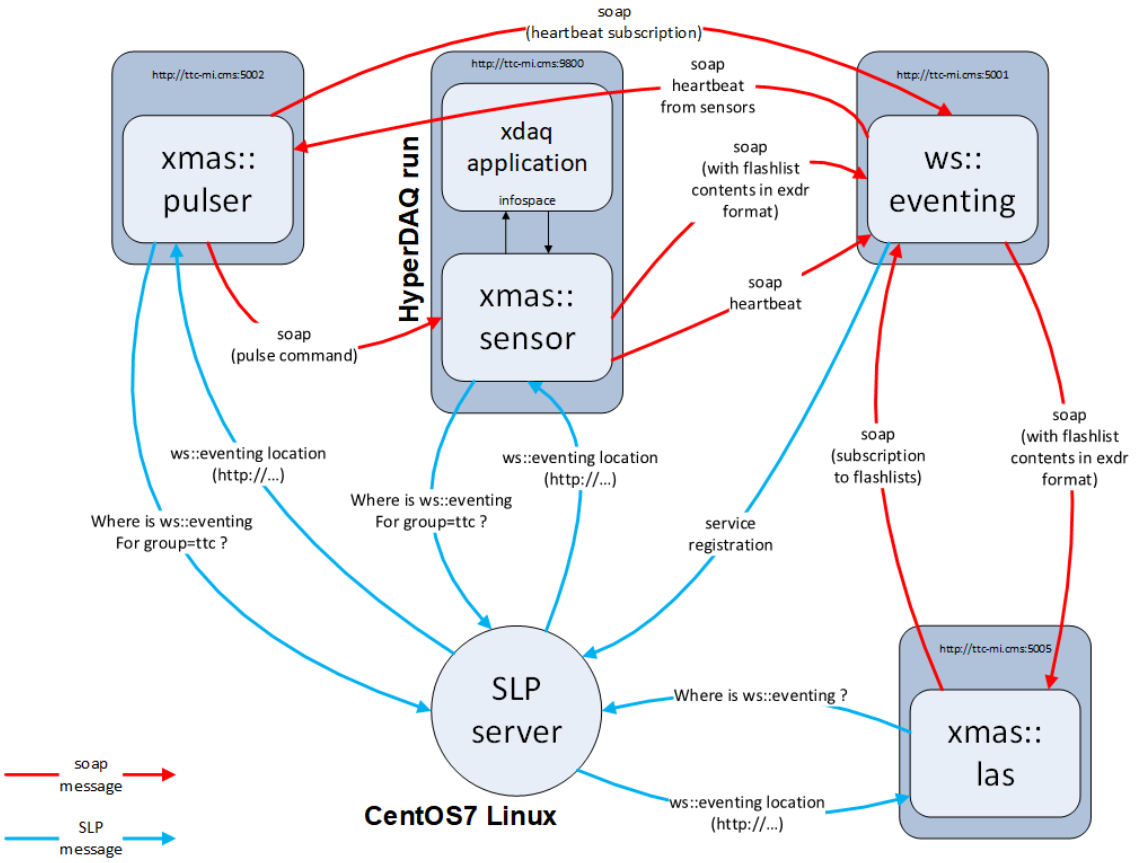


Figure 40 – A HyperDAQ server for monitoring simple quantity-specific of both events inside the application. Adapted from (HOLZNER, 2009).

The XDAQ Toolbox is instrumental in managing data acquisition and processing within the CMS experiment. It handles the vast volume of data generated by detectors and ensures efficient data transfer.

Additionally, it determines whether an event warrants attention in the triggering process. The toolbox also gathers data fragments from different detector components during event building, storing them in a suitable format for long-term storage and further analysis. This toolbox's flow of functionality is employed by the Slow Controller system, emphasizing the versatile utility of this framework.

2.5.3.2 Setup Libraries

CentOS 7 is a widely used Linux distribution that provides a stable and secure operating system environment. It includes a large number of libraries that are essential for running various applications and services. These libraries are designed to provide a common set of functions that can be used by different software programs.

Expanding on external libraries, the IPBus is a protocol made in C++, that is used in this scientific research and starts from installed external repositories and runs together with dependencies of other base libraries from CentOS 7.

It provides a standardized interface for accessing and controlling hardware devices like FPGAs, discussed in chapter 2.5.2.1.1. IPBus uses a message-based protocol that allows devices to communicate with each other using a set of commands and responses. A lot of resource libraries and packages are used in real-time during this process to control and monitor hardware devices in this experimental setup on Slow Controller.

Another component used is XML, a popular file format for storing and exchanging data. They are used for various purposes, such as configuration files, data exchange between software systems, and web services. XML files contain tags and attributes defining the data structure. Figure 41 is shown a configuration file XML example for accessing AMC13XG in the uTCA crate.

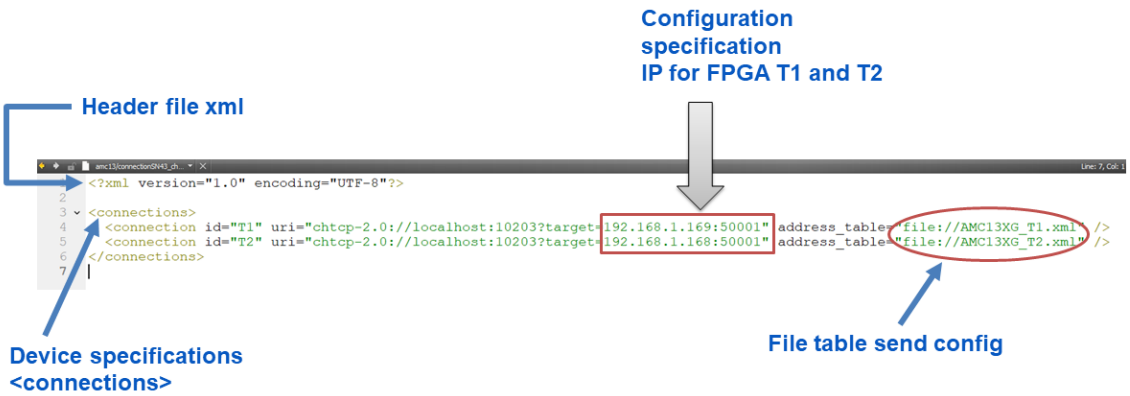


Figure 41 – Configuration file XML for connecting AMC13XG in PC host by TCP-IP.

In the given XML file, the configuration is primarily focused on establishing connections defined within the root element, `<connections>`, i.e., keeping a connection between the PC station and AMC13XG. Two connections are specified, each with unique identifiers,

denoted by the ”id” attribute: ”T1” and ”T2”. This structure allows for easy identification and manipulation of individual connections.

The first connection, with the identifier ”T1”, is characterized by a ”uri” attribute, which provides the necessary information for establishing the connection using the Hypertext Transfer Protocol Version 2 (HTTP/2). The ”uri” specifies a local host as the point of access, with the port number 10203. Further details are provided as a query string, which indicates the target IP address as 192.168.1.169 and the port number as 50001. In addition, an address table is supplied as a reference for the connection, utilizing the file ”AMC13XG_T1.xml”.

Similarly, the second connection, ”T2”, follows an identical pattern’s ”uri” and establishes the connection using HTTP/2, specifying the local host and port number 10203. The target IP address for this connection is 192.168.1.168, with a port number of 50001. This connection’s corresponding address table file is ”AMC13XG_T2.xml”.

This XML file’s well-organized and clearly defined structure enables efficient parsing and further processing by applications, in that case, the XDAQ toolbox requiring connection management. Moreover, using unique identifiers for each connection simplifies modifying or extracting specific connection details.

When compiling C++ files, the process involves converting human-readable source code into machine-executable code. The compilation process typically involves several steps, such as preprocessing, compilation, assembly, and linking. Compiling C++ files can be done using various tools, such as GNU Compiler Collection (GCC), Clang, and Visual Studio.

2.5.3.3 RPC Online Software: A Technical Overview

The RPC Online Software System (THYSSEN, 2011), integral to the CMS detector at the Muon System, carries out data acquisition, monitoring, and control. The system’s application within the Slow Controller of the RPC New Link System for the Muon subsystem is the core of this project.

The emphasis is primarily on adapting and incorporating the methodology into the Slow Controller environment, assessing its impact on Muon’s System’s functionality,

stability, and future enhancements.

From the slow control's standpoint, modifying the XDAQ system to monitor and regulate the New Link System, including the Link Board and Control Board, becomes vital. Figure 42 illustrates that this change involves creating new algorithms and metrics to examine the slow-changing parameters and trends related to the parameters of Front End Boards, status, and LBB's statistics.

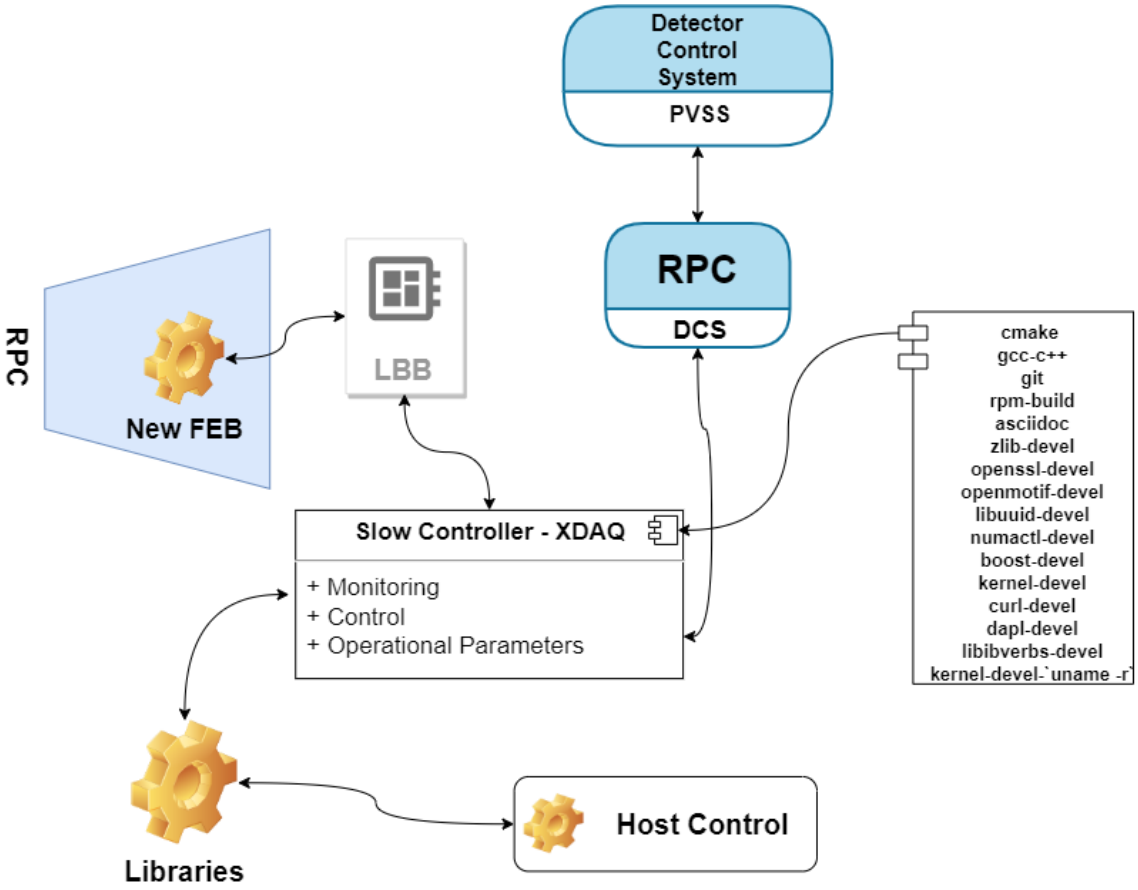


Figure 42 – Structure of the hierarchy tree of the slow control environment algorithm from geographical and hardware points of view.

It includes the active formulation of advanced algorithms and control loops to control long-term environmental factors such as temperature and pressure. Additionally, the Slow Controller adjusts operational parameters in the FEE crossing the LBB System to ensure the subsystems' stability and safety.

Using the RPC Online Software System for the Slow Controller in the CMS Muon detector presents considerable potential for enhancing the experiment's long-term performance and stability. For example, adjusting and incorporating the DCS (PAOLUCCI et al., 2010) into the Slow Controller environment might provide more accurate monitoring and

control of the detector subsystems over extended durations.

The RPC Online Software System for the Slow Controller centers on developing advanced algorithms and metrics for long-term data quality monitoring and control, optimizing environmental and hardware component management control loops for the New RPC Link System.

3- Development of the Slow Controller System

3.1- Initial Design

In the preliminary design utilizing CERN CentOS 7 x86_64, the installations may proceed in the default mode, except for the **Software Selection** stage. Therefore, it is imperative to determine the specific Base Environment packages to install at the Installation Summary screen. For a successful installation, it is necessary to choose a single software environment during installation.

For every environment, supplementary packages can be found as Add-ons. These Add-ons are displayed on the right-hand side of the screen, and the compilation of available options updates whenever a new environment is selected. Following the initial design for the Slow Controller, all Add-ons about the installation environment should be selected for CentOS 7.

Figure 43 shows Software Selection for a Server Installation on the variant of the installation International Organization for Standardization (ISO) image CERN CentOS 7 x86_64.

Another basic configuration after finishing the installation is the default network configuration for internet access. Internet access is essential for successfully configuring the necessary resources for the Slow Controller design in this thesis.

To access the NAT-MCH, connect it to the Host PC's Ethernet port. For optimal productivity, ensure the Host PC has simultaneous access to the uTCA crate and the internet. This action enables real-time implementation of settings and testing. To achieve this, install an additional network interface board on the Host PC.

After installing a second Ethernet card in the Host PC, it is necessary to configure it to receive network settings automatically. The Host PC with CentOS 7 installed from this base configuration is now ready to accept the configurations outlined in chapter 2. We will provide a detailed explanation of these configurations this chapter.

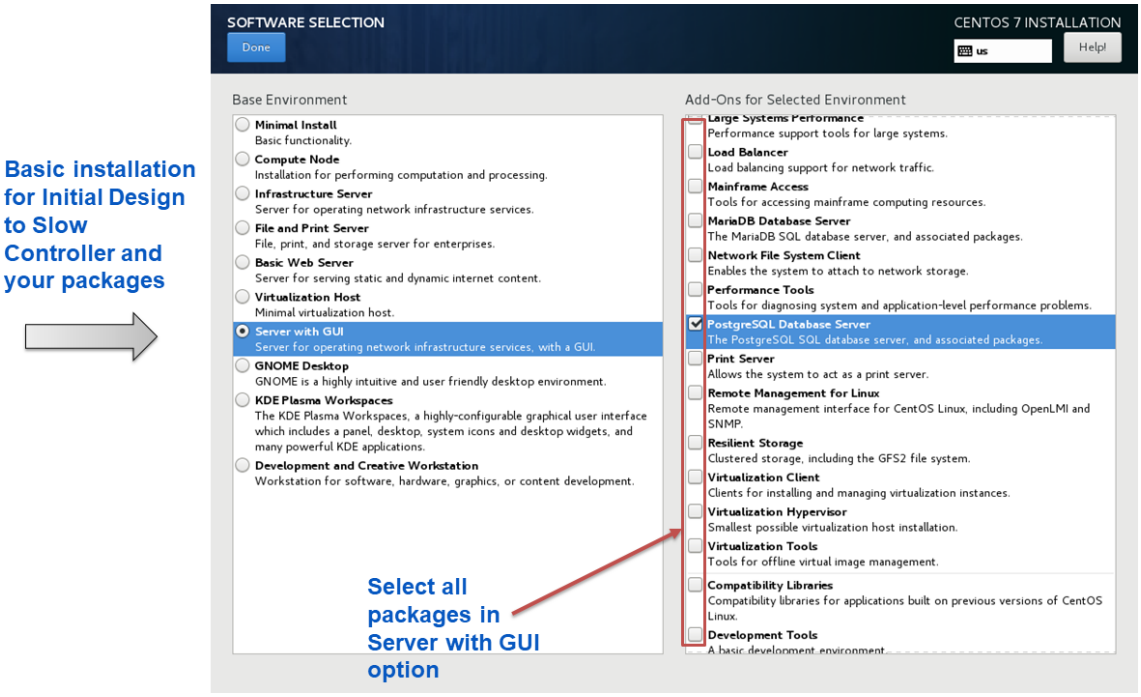


Figure 43 – Software Selection for a Server Installation in CERN CentOS 7.

3.2- uTCA Infrastructure

One of the initial Design principles of uTCA use in CMS has come from a comparison with VME point-to-point links.

In this Slow controller project, the Control Hub PCs, also called Host PC, talk to the uTCA hardware over dedicated switches on a private network (it is therefore restricted), performed by NAT-MCH work on slots MCH1 and MCH2 according to Figure 44.

This model of the network is accepted by default being logically more easily compatible for integration at the CMS control network.

The default hardware assembler in the uTCA crate catalyzes utilizing the Hardware Access Layer for uTCA (uHAL) with IPBus as the designated protocol to communicate with uTCA hardware.

Due to that, a single access point for the hardware - the uTCA crate - is set to default data communication concentrator from multiple clients engaging with that single hardware to support other cards with limited buffer capacity.

After this hardware assembly, the uTCA crate is ready. To receive and send commands to the uTCA crate, configure the NAT-MCH for the initial software configuration.

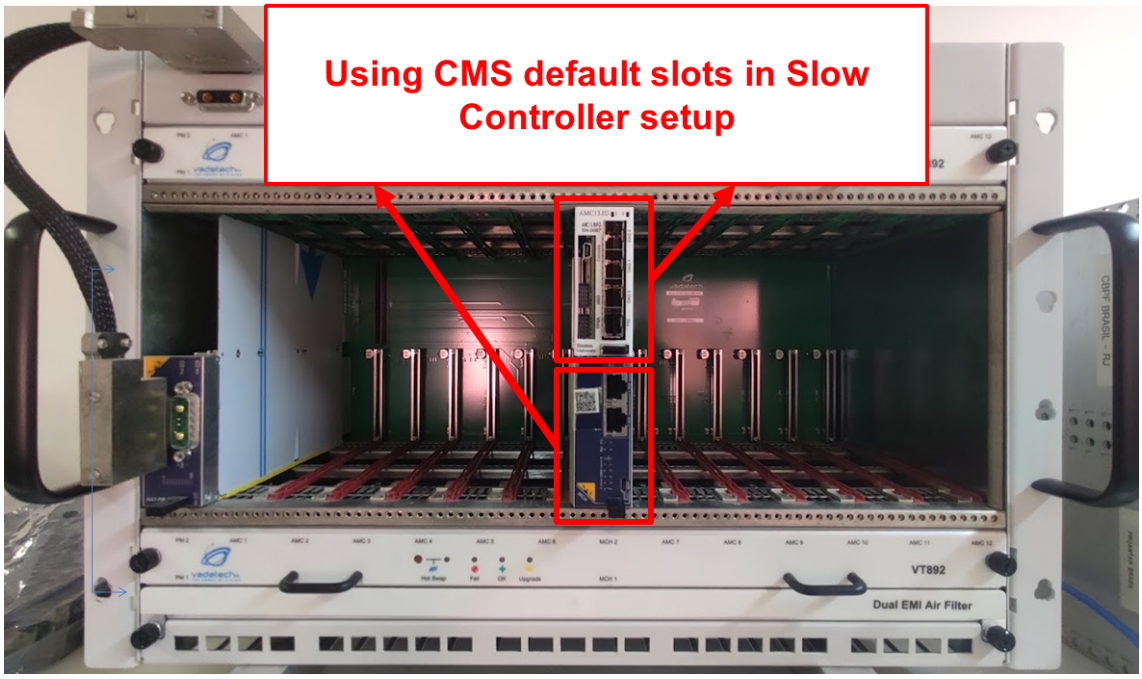


Figure 44 – Default slot configuration using MCH1 and MCH2 for starting network layout.

3.2.1 NAT-MCH

The initial action to set or discover the IP address of the NAT-MCH device is essential for managing, configuring, and enabling/disabling settings on the backplane of uTCA. According to the manual available on the manufacturer's website by NAT-MCH (EUROPE, 2023), this device has many manufacturers. Each configures a different way to set static IP for routering and access setup.

Each NAT-MCH module must have a unique IP address assigned per bridge machine, which links to the local carrier identification number, i.e., informed in the visible serial number.

To configure NAT-MCH modules with a static IP address or to gain knowledge about them, access the procedure through a USB or RS232 port using a terminal program and using the Putty program to access the USB port mapped to `/dev/ttyACM0`, with a data transfer velocity of 115200, as shown in Figure 45. Then, verify this setup using the command in Listing 1.

Listing 1 – Command to show Board configuration from NAT-MCH.

```
1 nat>bi
```

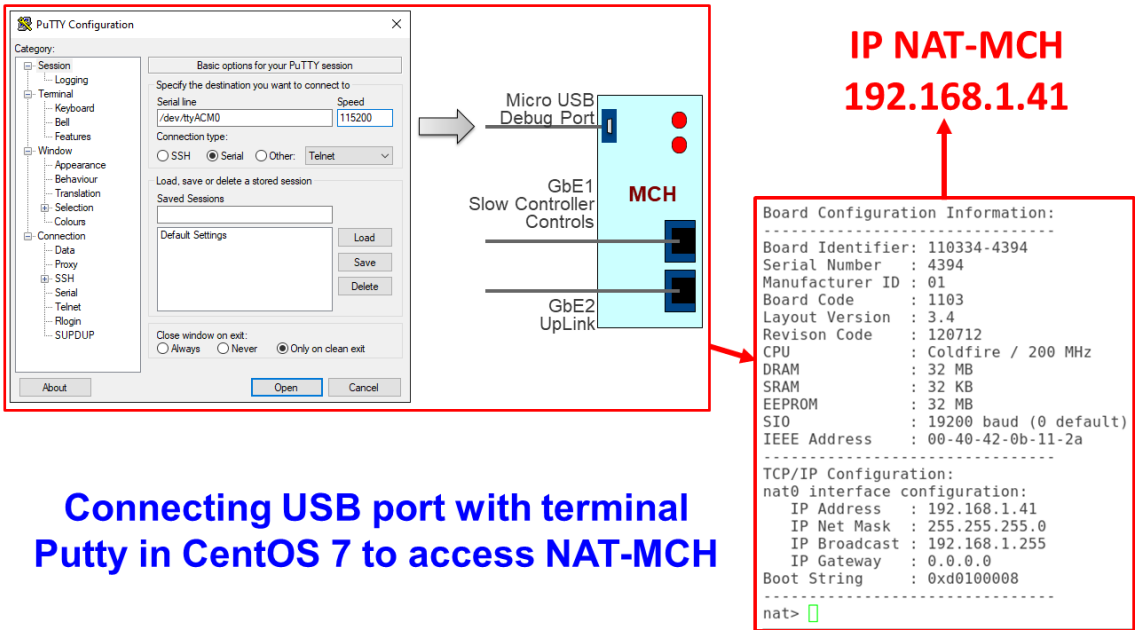


Figure 45 – Access by USB port to the discovery of IP default NAT-MCH.

The NAT-MCH has default IP 192.168.1.41, and through the Ethernet port, GbE1 will receive the network connection by PC Host computer connected to the same network class ”C” as the NAT-MCH device.

In addition, the connection router must supply a Dynamic Host Configuration Protocol (DHCP) service for the ”Default Gateway” or ”Gateway” entry in the output. This IP address of the NAT-MCH device keeps a bridge network between the PC Host and NAT-MCH.

Access to the device’s web interface is available with the IP 192.168.1.41 by the browser interface. In Figure 46, web access is provided with a username and password default by the device’s documentation default credentials.

After completing these settings, the uTCA crate can be accessed through the GbE1 port using the TCP-IP protocol for the Slow Controller of AMC modules.

Upon finalizing the hardware configuration, the CentOS 7 Linux system must utilize specific protocol-level functions to accurately transmit and receive data from the AMC modules connected to the uTCA crate.

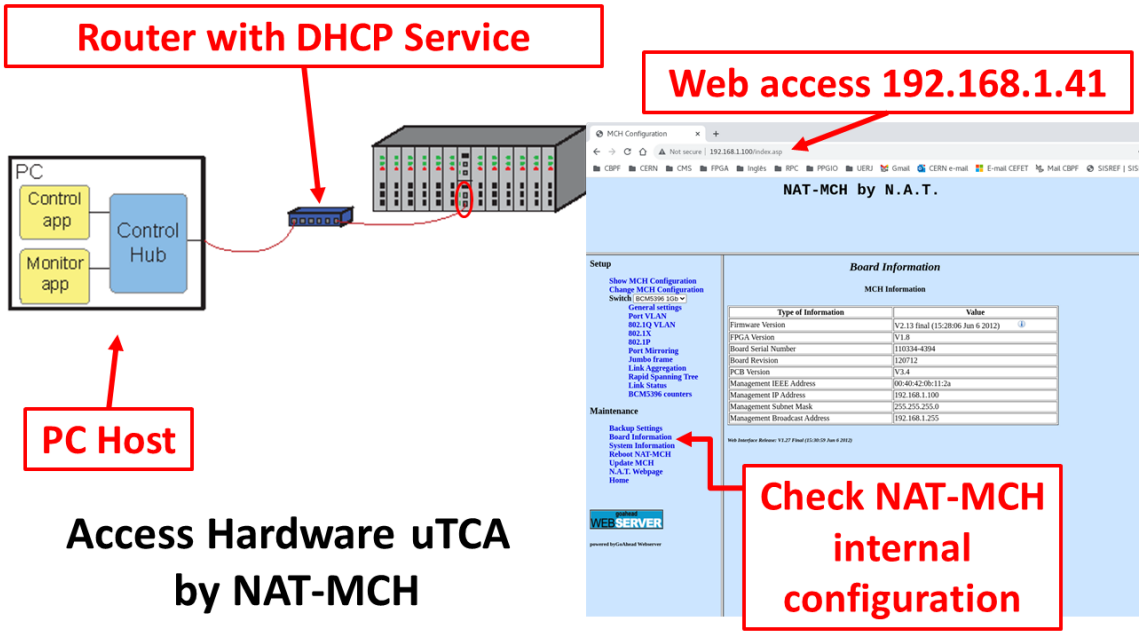


Figure 46 – Access uTCA by IP default NAT-MCH.

3.2.1.1 Installation of IPBus and ControlHub

The IPBus software suite provides a robust and efficient control link for the Slow Controller project, leveraging the IPBus protocol and ControlHub. Although uHAL resources are available, they are not utilized in the current work. Therefore, to install IPBus and ControlHub on CentOS 7, cleaning and aligning with repository versions is the first step, as indicated in Listing 2.

Listing 2 – Command to remove old versions.

```
1 $ sudo yum groupremove uhal
```

Download the appropriate repositories file and copy it to `/etc/yum.repos.d/`, or edit the individual file renaming to IPBus-sw.repo with the commands by Listing 3 from the address[1]. In this project, the file `IPBus-sw.repo` in folder `/etc/yum.repos.d/` is configured according to the new links available in CentOS 7 repositories.

Listing 3 – Installing new repositories to IPBus package.

```
1 [IPBus-sw-base]
```

[1] `https://IPBus.web.cern.ch/doc/user/html/_downloads/IPBus-sw.centos7_gcc8.x86_64.repo-o/etc/yum.repos.d/IPBus-sw.repo`

```
2  name=IPBus software repository
3  baseurl=http://www.cern.ch/IPBus/sw/release/2.8/repos/centos7_gcc8/x86_64/
       base/RPMS
4  enabled=1
5  gpgcheck=0
6
7  [IPBus-sw-updates]
8  name=IPBus software repository updates
9  baseurl=http://www.cern.ch/IPBus/sw/release/2.8/repos/centos7_gcc8/x86_64/
       updates/RPMS
10 enabled=1
11 gpgcheck=0
```

This version runs on Compiler GCC8 with dependencies boost 1.66 and pugiXML 1.9 provided in the IPBus repositories. To install the group of applications IPBus and ControlHub run commands in the terminal using Listing 4.

Listing 4 – Command to instal ControlHub.

```
1  $ sudo yum clean all
2  $ sudo yum groupinstall uhal controlhub
```

The IPBus software installs almost all its files under `/opt/cactus`. So when running applications that use resources of IPBus, needed to update the environment variables accordingly the Listing 5.

Listing 5 – Command to update the environment variables.

```
1  $ export LD_LIBRARY_PATH=/opt/cactus/lib:\$LD_LIBRARY_PATH
```

3.3- Setup AMC13XG for Slow Controller

3.3.1 Installation Settings to AMC13XG

The configuration of AMC13XG requires specific drivers and software to be installed and configured on a CentOS 7 system. In the uTCA and NAT-MCH configuration, the hardware drives and software are installed by repositories default in CentOS 7.

The AMC13XG configuration demands a particular Installation of the necessary dependencies. New repositories were added at folder `/etc/yum.repos.d/` to successfully install drives and software to the initial connection setup.

First, the AMC13XG is set with an IP address according to documentation (HAZEN, 2012) to enable access for network NAT created by NAT-MCH using the second card network installed in the PC Host. The AMC13 Tool package, consisting of drives and software, is installed from repositories using specific commands to initiate the initial setup.

CACTUS is a software framework explicitly used for control and monitoring systems. To install CACTUS on CentOS 7, create or modify the file `/etc/yum.repos.d/cactus.repo` to find the respective functional link for installing the main program and dependencies with commands by Listing 6.

Listing 6 – The file /etc/yum.repos.d/cactus.repo on CentOS 7.

```
1  [cactus-base]
2  name=CACTUS Project Software Repository
3  baseurl=https://IPBus.web.cern.ch/sw/release/2.8/repos/centos7_gcc8/x86_64/
       base/RPMS/
4  enabled=1
5  gpgcheck=0
6
7  [cactus-updates]
8  name=CACTUS Project Software Repository Updates
9  baseurl=https://IPBus.web.cern.ch/sw/release/2.8/repos/centos7_gcc8/x86_64/
       updates/RPMS/
10 enabled=1
11 gpgcheck=0
```

After configuration repositories correctly and successfully, the installation to CACTUS software framework can be done with commands by Listing 7 in the terminal.

Listing 7 – Install CACTUS and AMC13 Tools on CentOS 7.

```
1 $ sudo yum groupinstall cactus
2 $ sudo yum groupinstall amc13
```

All of the respective packages[2] should be installed. The CACTUS uses all of the packages to run in CentOS 7.

Now CentOS 7 has all libraries to operate the AMC13XG hardware library developed by CERN for hardware communication. However, these libraries only run with the GCC8 version. For this reason, the GCC8 installation must be done before editing the config file of the AMC13XG with commands by Listing 8.

[2] cactusboards-amc13-tools-1.2.14-0.centos7, cactusboards-amc13-amc13-1.2.14-0.centos7, cactuscore-uhal-grammars, cactuscore-uhal-uhal, cactuscore-uhal-tests, cactuscore-uhal-pycohal, cactuscore-uhal-gui, cactuscore-uhal-tools, cactuscore-controlhub, cactusboards-amc13-amc13, cactusboards-amc13-tools and pugixml-devel

Listing 8 – Installing Prerequisits from GCC8 on Centos 7.

```
1 $ sudo yum install centos-release-scl
2 $ sudo yum install devtoolset-8
3 $ sudo yum install git cmake3 gcc-c++ gcc binutils libX11-devel libXpm-devel
      libXft-devel libXext-devel openssl-devel
4 $ sudo yum install gcc-gfortran pcre-devel mesa-libGL-devel mesa-libGLU-devel
       glew-devel ftgl-devel mysql-devel fftw-devel cfitsio-devel graphviz-
      devel avahi-compat-libdns_sd-devel openldap-devel python-devel libxml2-
      devel gsl-devel
```

Another prerequisite to run AMC13 Tools is the Developer Toolset 8 (devtoolset-8) provides current versions of the GNU Compiler Collection, GNU Debugger, and other development, debugging, and performance monitoring tools due to working together with GCC8. To install the full devtoolset-8 on CentOS 7, the following commands it run the line code by Listing 9 in the terminal.

Listing 9 – Installation a devtoolset-8 on CentOS 7 package with repositories.

```
1 $ sudo yum install centos-release-scl
2 $ sudo yum install devtoolset-8
3 $ sudo scl enable devtoolset-8 -- bash
```

After installing all libraries according to CMS documentation, the folder `/opt/cactus/` becomes available, containing all AMC13Tools installed in the default folder. To finish the setup resources for AMC13 package Tools, insert a line in the environment variables to run `AMC13Tools2.exe` in folder `/opt/cactus/etc/amc13/` as line command Listing 10 in the terminal.

Listing 10 – Command to update the environment variables.

```
1 $ export AMC13_ADDRESS_TABLE_PATH=~/amc13/amc13/etc/amc13
```

The AMC13 is accessible via an Ethernet connection to the commercial NAT-MCH module and therefore is accessible via NAT network. Each AMC13 has two IP addresses, one for the Spartan FPGA and another for the Kintex FPGA, and their network assignments are based on the AMC13 Serial Number (SN) (HILL, 2012):

- Spartan IP Address: 192.168.1.(254-2*SN);
- Kintex IP Address: 192.168.1.(255-2*SN);

At the default configuration in AMC13XG documentation, the file XML is created or edited in the default folder[3].

To know the IP addresses, one for T1 and one for T2, by default, in AMC13XG using in Slow Controller project, certified the SN of the AMC13XG at Slow Controller is 087. Appling at the math expression to complete the numbers of the IP addresses, the result it is:

- For the Spartan-6 FPGA: (254-2*SN) = 254 – 2*SN = 254 - 2*(87) = 080 = T2;
- For the Kintex-7 FPGA: (255-2*SN) = 255 – 2*SN = 255 - 2*(87) = 081 = T1;

Then, we have the IP access in NAT network by NAT-MCH. The IP for the Spartan-6 FPGA IP = 192.168.1.80, and IP for the Virtex-6 FPGA IP = 192.168.1.81. The XML file in `/opt/cactus/amc13/amc13/connectionSN87_ch.xml` it is changed as shown in Listing 11.

Listing 11 – Configuration IP in xml file for AMC13XG.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <connections>
3 <connection id="T1" uri="chtcp-2.0://localhost:10203?target
      =192.168.1.81:50001" address_table="file://AMC13XG_T1.xml" />
4 <connection id="T2" uri="chtcp-2.0://localhost:10203?target
      =192.168.1.80:50001" address_table="file://AMC13XG_T2.xml" />
5 </connections>
```

As the documentation follows, the XML file configuration requests a connection simultaneous with T1 and T2. This connection method avoids connection errors. The Controlhub service starts before the connection in AMC13XG because it provides controlling support across devices using XML tables in the default folder from AMC13Tools.

[3] /opt/cactus/amc13/amc13/name-xml-file.xml

4- Implementing Instrumentation and Control

Prioritizing successful transmission and reception of all commands is crucial for assembling instrumentation with an AMC13XG card system. This chapter details the development of a basic control system with AMC13XG based on the research conducted and the specific hardware assembly and configuration used in the project.

While several approaches exist to test this functionality, the most effective methods often involve using a Linux terminal. The chapter illustrates the execution of commands directly on the terminal connected to the AMC13XG. Later, we examine the behavior of each command and command group.

We employ a file containing a sequence of commands to optimize command insertion in the terminal further. The commands are inserted into this file to automate the initialization of critical functions and the management and test data's transmission and reception.

Upon conducting the necessary data transmission and reception management tests, we utilized numerous Python scripts to automate and streamline the process, enhancing instrumentation implementation.

4.1- Test Connections to AMC13XG

To initiate the procedure of accessing the hardware from the AMC13XG connection, the first step is to export the links from XML tables from the environment variables path folder `/opt/cactus/lib`. To achieve this task successfully, the command in Listing 12 runs within the terminal before the initial access.

Listing 12 – Set the environment folder path for the links from XML tables.

```
1 export LD_LIBRARY_PATH=/opt/cactus/lib
```

So, as indicated in the hardware control documentation executed by the IPBus software, before any connection to the AMC13XG, it is necessary to start the ControlHub

service shown in Listing 13 to guarantee the correct routing of the packets received and sent between the Host PC and the hardware enveloped in the datagram by the IPBus.

Listing 13 – Start ControlHub service in CentOS 7.

```
1          $ sudo service controlhub start
```

After these specific settings, the connection on AMC13XG runs with the command in Listing 14.

Listing 14 – Set the environment variables the links from xml tables.

```
1 AMCTool2.exe -c /opt/cactus/etc/amc13/connectionSN87_ch.xml
```

The command has successfully established the connection to the AMC13XG device. First, the basic command ”list” determines how many AMC13XG devices are on the uTCA crate. Next, certify the SN 87 of the device connected to the uTCA Crate.

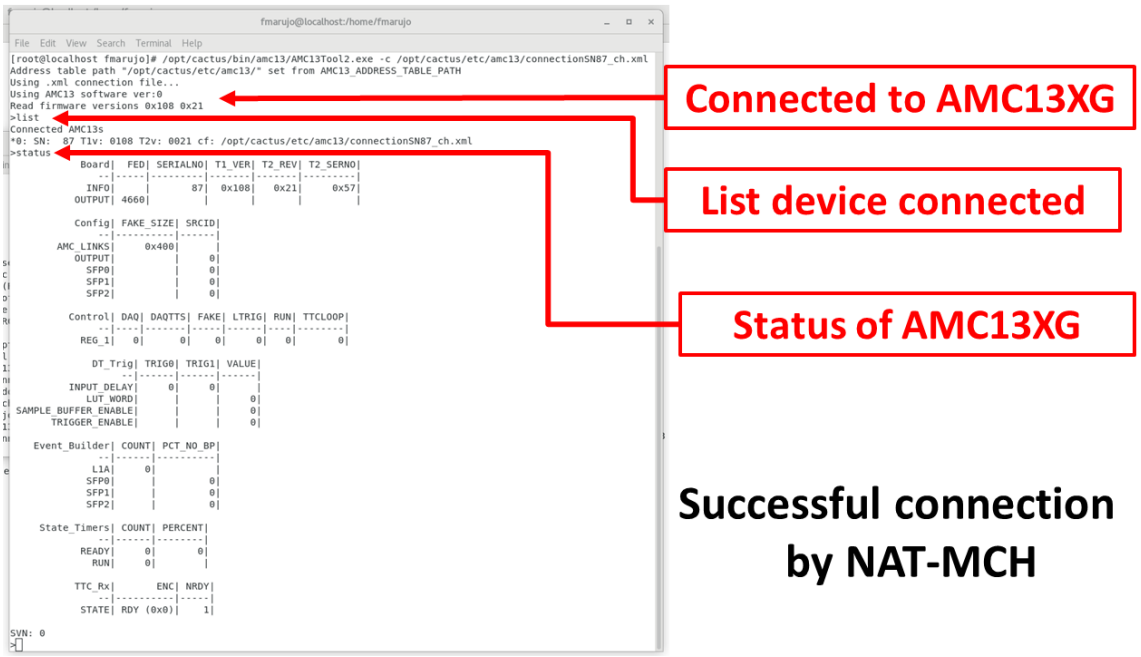


Figure 47 – Connection successful on AMC13XG runs in terminal Linux.

In the testing context, execute the command ”status” to examine all the operational functions on the device. While remaining connected to the device, Figure 47 demonstrates that upon running the ”status” command, it verified the expected functionality of the AMC13XG and provided an overview of all functions, indicating the absence of data reception or transmission.

4.2- Local Event Builder Test

The LHC dispatches synchronization signals via the TCDS system, and the AMC13XG can internally simulate this signal. This signal must be distributed across all slots in the uTCA crate, enabling other AMC modules to receive this synchronization generated by the AMC13XG.

The CMS standard configurations with the AMC13XG module in the upper MCH2 site for the SFP+3 transceivers, situated in the TTC (bottom) side of the AMC13XG, is activated in Loopback mode with an optical fiber bridging the ”out” and ”in” of the TTC at the respective SFP+3.

Following the hardware configuration with the fiber optic on the SFP+3 TTC (bottom), the AMC13XG dispatches a command to initiate a general reset and counter reset.

Listing 15 – General reset and counter reset commands in AMC13XG.

```
1 [root@cms ~]# AMC13Tool2.exe -c /opt/cactus/etc/amc13/connectionSN87_ch.xml
2 Address table path "/opt/cactus/etc/amc13/" set from AMC13_ADDRESS_TABLE_PATH
3 Using .xml connection file...
4 Using AMC13 software ver:50470
5 Read firmware versions 0x108 0x21
6 >rg #General reset
7 >rc #Counter reset
```

The general reset function clears the prior logic to avoid any issues during data transfer. In contrast, the counter reset function resets all the counters within the AMC13XG logic. These actions occur at lines 6 and 7 of Listing 15. It is important to note that for firmware versions preceding 0x20f/0x4006, this reset also affects the DAQ links. However, from this version onwards, the ’rd’ command should be utilized to reset the DAQ separately.

The generation of false events in the AMC13XG and the reading of false data by the IPBus are essential components of slow controller functionality. This requirement necessitates enabling the backplane link with false event options and distributing the TTC signals in loopback mode from the AMC13XG to all AMC cards through the uTCA crate bus.

Listing 16 – The AMC13 generates fake data signals and enable Backplane Links.

```
1 >en 1-12 f t # enable all backplane links
2 parsed list "1-12" as mask 0xfff
3 Enabling fake data
4 Enabling TTS as TTC for loop-back
5 AMC13 out of run mode
6 AMC13 is back in run mode and ready
```

Upon executing these commands as specified in line 1 of Listing 16, the slow controller activates the AMC13XG's internal trigger and disseminates the emulated uTCA signal of this backplane to all slots.

Algorithm 1 – Basic experimental to start control system.

Input: $en1-12$ f t;
$en1-12$ = enables all inputs which show AMC link ready;
F = to generate fake events in AMC13;
T = to enable loop-back TTC on TTS output fiber;
Output: on $AMC13$ Event Builder waiting $L1A$ signal

1 **begin**
2 $uTCA \leftarrow$ enable$(1-12)$;
3 $T1-Kintex \leftarrow$ Event Builder on;
4 **foreach** set state on for **do**
5 $L1A \leftarrow$ Level 1 Trigger;
6 $Bc0 \leftarrow$ Bunch-count zero with bunch-count reset;
7 $OcR \leftarrow$ Orbit-count reset;
8 $EcR \leftarrow$ Event-count reset;
9 $TTC \leftarrow$ waiting;
10 $L1Abitstream \leftarrow$ waiting

At the hardware level, the commands executed at the terminal prompt activate many specific bits to bypass manual insertion in their respective registers. Algorithm 1 demonstrates the logical sequence of the primary control initialization set at the hardware level.

From this point forward, the AMC13XG is either in a state of receiving or sending data. This operation runs only when the AMC13XG receives an actual signal from the intricate TCDS system is configured in loopback mode.

4.3- DAQ Outputs

Once the AMC13XG has assembled an event and prepared it for readout, the packet is dispatched to the Synchronous Dynamic Random Access Memory (SDRAM) for storage, to the DAQ Link for readout over optical fiber, or both. When transmitted over the DAQ Link, the fully-formatted receiver eventually receives the DAQ event.

This receiver is responsible for delivering the event to the CMS high-level trigger. However, a part of this packet is subsequently discarded and solely utilized for the delivery confirmation of these bytes.

When the SFP+3 port is in loopback mode, the Listing 17 command enables the SFP+1 port for local transmission upon receiving a simulated trigger signal from its internal event generator feature.

Listing 17 – From Event Builder enables DAQLSC to output Fiber.

```
1 >daq 2 L #enable SFP1 DAQ fiber output SFP1
2 SFP0 and SFP1 enabled
3 Local mode enabled
4 AMC13 set to ignore DAQ data
5 Best to do a DAQ reset (rd) after changing link settings
```

The firmware controlling the AMC13XG links incorporates the ability to buffer DAQ Link for readout over an optical fiber (DAQLSC). This buffering process occurs between T1 and the enabled SFP+1 optical link, facilitating data transfer.

A script is implemented from this point to automate the command insertion process through the terminal, streamlining and optimizing the overall procedure.

The script presented in Algorithm 2 initializes the AMC13XG and enables the SFP+1 port as a DAQ output to produce data output. The internal logic produces an artificial trigger signal, fed back through the loopback on the SFP+3 port and directed to exit based on the internal hardware construction logic.

This script conceptualizes the transmission of 64-bit wide words originating from the registers within the FPGA T1 (Kintex), as implemented in the firmware[1] of this particular version.

[1] http://physics.bu.edu/~wusx/download/AMC13/AMC13XGspecK0x108S0x21.txt

Algorithm 2 – The command insertion process optimize.

Input: $en1-12$ f t;
$daq\ 2\ L$ = enables all inputs which show AMC link ready;
$fed\ 0\ 100$ = Set an AMC13's DAQ-link's fed IDusage;
$lt\ c$ = enable continuous mode (configure with localL1A);
$localL1A$ o 1 1= Configure LocalL1A enabled mode=0 burst=1 rate=1 rules=0;
$lt\ 5$ = generate five trigger;
Output: on $SFP1$ output five package data to the DAQ Link buffer

```
1  begin
2      AMC13XG ← enable − SFP1;
3      T1 − Kintex ← 5 trigger;
4      foreach set state on for do
5          SFP1 ← send byte register 5 (64-bit word);
6          SFP1 ← send byte register 4 (64-bit word);
7          SFP1 ← send byte register 3 (64-bit word);
8          SFP1 ← send byte register 2 (64-bit word);
9          SFP1 ← send byte register 1 (64-bit word);
10     Trigger ← stop;
11     file ← AMC13XGDump_[year]-[month]-[day]_[hour][minute][second].txt
```

4.4- DAQ Inputs

Similar to the DAQ output mode seen in chapter 4.3, the AMC13XG hardware operates as a data transmission input with the same SFP+ port selection procedure chosen by the driver.

After that correct selection, the fiber optic DAQ link is activated and, with that, is set up to receive events with 64-bit words. Consequently, the top-level format is always determined in 64-bit terms and is ready for readout by the AMC13XG from this stage.

The data transfer occurs as a stream consisting of 16-bit units. Data is retrieved from the AMC13XG memory buffer as 32-bit words via IPBus and can be directed either to the SDRAM for storage, the DAQ Link for transmission through optical fiber, or both concurrently.

The 64-bit packages (in little-endian format) are ultimately recovered and conveyed to the T1 firmware. The script detailed in Algorithm 3 initiates the AMC13XG and activates the SFP+1 port to input data. An internal trigger signal is synthetically generated following the hardware's construction to provoke data input.

The optical link is enabled differently during AMC13XG driver initialization for data

Algorithm 3 – Algorithm by data input on SFP1.

Input: $en1 - 12$ f t;
$daq\ 2\ L$ = enable all inputs indicating AMC link readiness;
$lt\ c$ = activate continuous mode (setup with localL1A);
$localL1A$ o 1 1= Set LocalL1A enabled mode=0 burst=1 rate=1 rules=0;
$lt\ 5$ = generate five triggers;
Output: on $SFP1$ input to the DAQ Link buffer

```
1  begin
2      AMC13XG ← activate − SFP1;
3      T1 − Kintex ← 5 triggers;
4      foreach activate state for do
5          buffer SDRAM/T1 ← SFP1 data raw 1;
6          buffer SDRAM/T1 ← SFP1 data raw 2;
7          buffer SDRAM/T1 ← SFP1 data raw 3;
8          buffer SDRAM/T1 ← SFP1 data raw 4;
9          buffer SDRAM/T1 ← SFP1 data raw 5;
10     Trigger ← halt;
11     file ← AMC13XGDump_[year]-[month]-[day]_[hour][minute][second].txt
```

input and output. In other firmware versions, it is feasible to allow SFP+s for both input and output concurrently.

The addresses of registers can vary based on the firmware version utilized. Chapter 5 will detail the addresses used in this specific firmware version, 0x108.

4.5- Python in Slow Controller

Python, a versatile programming language acclaimed for its readability and adaptability, is extensively utilized in high-energy physics thanks to its robust scientific library ecosystem.

The high-level software community at CERN frequently transitions to Python. This shift is advantageous as it aids in managing the instrumentation required for high-energy physics experiments, such as detectors and data acquisition systems.

The migration of resources for Python enables researchers in high-energy physics to concentrate more on analyzing data from particle collisions, leading to quicker discoveries rather than dealing with intricate hardware communication.

A central program, followed by other subprograms in Python, makes and processes

events utilizing tools such as `AMC13Tools2.exe`, following CMS hardware and software development standards.

It performs specific actions like turning DAQ Links on/off and inspecting/modifying DAQ Path status. These operations are particularly significant in the context of a high-energy physics experiment, especially in a fundamental Slow Controller project.

Algorithm 4 displays the functionality flow of the Python-written program. This integration into the Slow Controller validates its compatibility with other functionalities.

Algorithm 4 – Python code to implement basic control system.

Input: Set directories and tools;
Disable Links in AMC13;
Enable DAQ Links and Start Run;
Output: on $SFP1$ output five package data to the DAQ Link buffer
begin
    Create appropriate AMC13 script for initialization;
    foreach set state on for **do**
        $L1A \leftarrow$ Level 1 Trigger;
        $Bc0 \leftarrow$ Bunch-count zero with bunch-count reset;
        $OcR \leftarrow$ Orbit-count reset;
        $EcR \leftarrow$ Event-count reset;
    $Trigger \leftarrow$ halt;
    $file \leftarrow$ AMC13XGDump_[year]-[month]-[day]_[hour][minute][second].txt

The Slow Controller's Python code defines a dictionary of functions, bundling operations such as initialization, writing, and reading from registers, and error management.

Additionally, it structures an interactive program that employs user input to invoke and execute the functions defined in the dictionary (config files), granting users direct control over these operations in the AMC13XG hardware.

The Python language-based interactive program for managing an AMC13XG device validates interaction functionality. It incorporates a range of commands for initialization, status checking, register reading and writing, and error detection, utilizing methods crucial in the instrumentation of a slow controller.

This program lays a foundation for constructing an interactive, command-driven system to control hardware devices using Python, explicitly targeting AMC card devices and uTCA technology.

5- Results and Discussions

Within the critical context of high-energy physics and the CMS experiment at CERN, this chapter unveils the results of the basic implementation of our electronic instrumentation project for the Slow Controller.

Each result is expounded upon, elucidating its implications for the intertwined elements of the project, encompassing the hardware, software, firmware, and their collective impact on the overall upgrade in the Muon Group and subsystems.

As we decode the outcomes, we place them against high-energy physics and impact on research and functional system possible applicability, demonstrating their potential to enhance future next steps at development in the Slow Controller.

All of the results and conclusions of this work justify their true significance within the CMS experiment's extensive and diversified scope, from the findings in the Slow Controller have profound implications for the scientific community involved and the group collaboration, providing new solutions and paving the way for innovations within the dynamic field of electronic instrumentation.

5.1- Initial Considerations

Engaging in comprehensive discussions on the results attained from this project of the Slow Controller, we maintain a balanced focus on quantitative and qualitative analyses. These analyses improve it with a dual perspective: punctual, attending to the minutiae of the Link System, and global, encompassing the broader implications for the experimental development group.

The quantitative results depict measurable progress and concrete milestones about using the AMC13XG card. They provide precise data points and indicators that quantify the advancements in electronic instrumentation, enhancing our understanding of the Link System, specifically about the Control Board and its routine under various experimental conditions.

Conversely, the qualitative results provide a nuanced exploration of less tangible aspects of this project, but achievable in the following stages of development.

They paint a rich picture of our work's experimental hardware and software and relational dimensions, capturing the complexities of the hardware-software interplay within the Link System and the broader practical context for the upgrade LS3 period and Back-End.

These results offer valuable insights into the intricate dynamics and unforeseen challenges during the project, paving the way for future explorations and innovations in electronic systems for high-energy physics.

5.2- Advances in Firmware

The project establishes a functional primary system controller based on firmware, using FPGA T1 with version 0108 and FPGA T2 with version 0021 as the standard.

Due to project constraints, with only one AMC13XG unit available in the laboratory, we decided against altering the firmware FPGA T1 with version 0108 and FPGA T2 with version 0021.

AMC card firmware updates require a reliable source, considering the array of AMC13XG manufacturers and the multitude of firmware versions available, totaling 366. Table 3 contains all versions that have available.

Table 3 – Firmware available to AM13XG: 366 versions.

Firmware type	Number of versions	Generation
AMC13XG CMS with New Event Builder (5G DAQ)	78	Old/ before April 2016
AMC13XG CMS with New Event Builder (10G DAQ)	6	
AMC13XG HCAL with Local Trigger (5G DAQ)	74	
AMC13XG HCAL with Local Trigger (10G DAQ)	8	
AMC13XG Fake AMC Data Generator	10	
AMC13XG Test data receiver	5	
AMC13XG G-2	68	
AMC13XG old event builder (obsolete)	23	
AMC13 v1	48	
AMC13XG CMS with New Event Builder (10G DAQ)	15	New/ After April 2016
AMC13XG HCAL with Local Trigger (10G DAQ)	20	
AMC13XG G-2	11	

As the research advanced, the team comprehended the necessity to evaluate firmware alterations preceding April 2016[1], scrutinize the operation of the existing version, and delve into modifications incorporated in versions postdating April 2016[2].

Precautions are necessary to avoid any negative impact on research results due to unsuccessful in firmware or non-compliant changes that may cause permanent damage to the hardware.

The firmware version currently deployed in project development demonstrated adequacy for data analysis. The investigation revealed that this firmware version could transmit and receive the fundamental 80-bit hardware Slow Controller data, accompanied by four identification bits, as facilitated by the uTCA crate in software.

The AMC13XG hardware's support for sending and receiving data with 8b/10b encoding is crucial for this implementation. Without it, it would be necessary to rewrite a specific firmware for T1 and T2.

This initial work served as a point of discussion for the research in collaboration with the IPM (BOGHRATI et al., 2023). This analysis of the Slow Controller's implementation achieved satisfactory hardware and software results, positioning it as a benchmark for a

[1] `http://ohm.bu.edu/~dzou/preApr2016-firmware.cgi`
[2] `http://ohm.bu.edu/~amc13/firmware.cgi`

large-scale system.

The characteristics in the AMC13XG and Control Board firmware suggest further research towards developing innovative firmware. This development endeavor would encompass the formulation of an optimized protocol aimed at enhancing monitoring and control in high-level software while guaranteeing compatibility with the CMS standard.

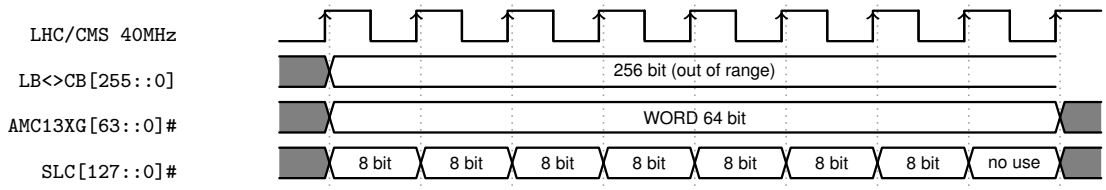


Figure 48 – Differences between Data-frames assembly by firmware.

Still addressing firmware differences, Figure 48 illustrates a communication data frame between the Link Board and the Control Board and a comparison with the AMC13XG communication datagram.

The datagram, dispatched to the AMC13XG's optical link via TCP-IP, can undergo packaging following the four standard CMS headers. This provision is contingent on the Control Board firmware's design to recognize the 84-bit hardware in conjunction with the Link Board selection in the forthcoming firmware version from the Control Board.

In the basic specification of the Control Board, as outlined in Chapter 2, 80 bits of primary internal control, 4 bits of external control, and additional bits for data confirmation are required, totaling 120 bits. The AMC13XG can handle this data exchange even with its older firmware version.

5.3- Integration in LHC signals

The LHC's TCDS System provides signals from the CMS TTC system and all subsystems in the muon detector.

The all-encompassing TTC system architecture derives its foundation from CMS standard hardware. It employs the AMC13XG to distribute synchronous timing, level-1 trigger, and broadcast and individually-addressed control signals to electronic controllers. The timing of these signals is meticulously synchronized relative to the LHC bunch struc-

ture, duly considering delays attributed to particle time-of-flight and signal propagation.

The AMC13XG card is radiation-resistant and used in every trigger distribution zone. The Slow Controller System generates the same signals as the TCDS broadcasts locally.

The Slow Controller is also exploring this functionality experimentally, specifically the sending and receiving data packages when a trigger signal activates the hardware logic. This trigger signal could be local or physically connected to the CMS TTC system to share with LBB.

The paramount finding from this research is the Slow Controller's ability to independently broadcast control and monitoring signals, irrespective of its connectivity to the signals emanating from the LHC. This autonomous functioning notably amplifies the subsystem's versatility and adaptability.

5.4- Electronics Design

The program developed as the foundational groundwork for the Slow Controller instrumentation, depicted in Algorithm 2, implements a universal reset of commands, states, and internal counters in the AMC13XG. It acquires all the registers utilized by FPGA T1 and T2, culminating in a comprehensive status report at the highest level (status reports have nine levels).

The level 9 status produces an all-encompassing hardware report. However, this part of the research describes collecting information about the transmission or reception of words in bytes and the addresses used in data storage.

The Buffer Monitor, the SFP+ memory, and the Slink Express transmission system are all halted in the given initial status, as demonstrated in Figure 49. The algorithm forces the initial reset state to ensure that AMC13XG places its digital systems without information or communication data.

As the algorithm routine progresses, following the generation of a simulated LHC signal to trigger an event, the hardware becomes prepared for data transmission or reception. The SFP+ ports are initialized and reset to ensure the absence of any unwanted data stored in the internal buffer.

```
 1    Monitor_Buffer| VALUE|
 2                  --|------|
 3          AVAILABLE|     1|
 4              EMPTY|     1|
 5           EOI_TYPE|     0|
 6      EVT_AFTER_EOI|     0|
 7
 8             SFP_ROM|     TTC|
 9                  --|--------|
10           ROM_DATA| 0x70403|
11
12      Slink_Express| L|
13                  --|--|
14           ANY_DOWN| 1|
```

Figure 49 – Hardware status in level 9 in a halted state.

One of the key findings thus far is the successful manipulation of the hardware to enable data transmission through the SFP+ port with DAQLSC mode activated. Figure 50a illustrates the criticality of enabling the SFP+ ports for seamless data communication.

The hardware block diagram of the AMC13XG, as depicted in Figure 50b, demonstrates the data flow when an SFP+ port operates in DAQLSC mode. In this mode, data can be output from an AMC module connected to the uTCA or stored in the FPGA T1 registers for sending serialization GBT.

As mentioned earlier, the block diagram routine is actually incorporated in this firmware version on both T1 and T2, as their intricate electronic designs on the 12-layer boards require physical connections to their respective devices.

With certain limitations, it is possible to modify this state machine, implemented through firmware, to enhance the integration of the upcoming version, particularly in the Slow Controller, aligning it more closely with the requirements of the Control Board.

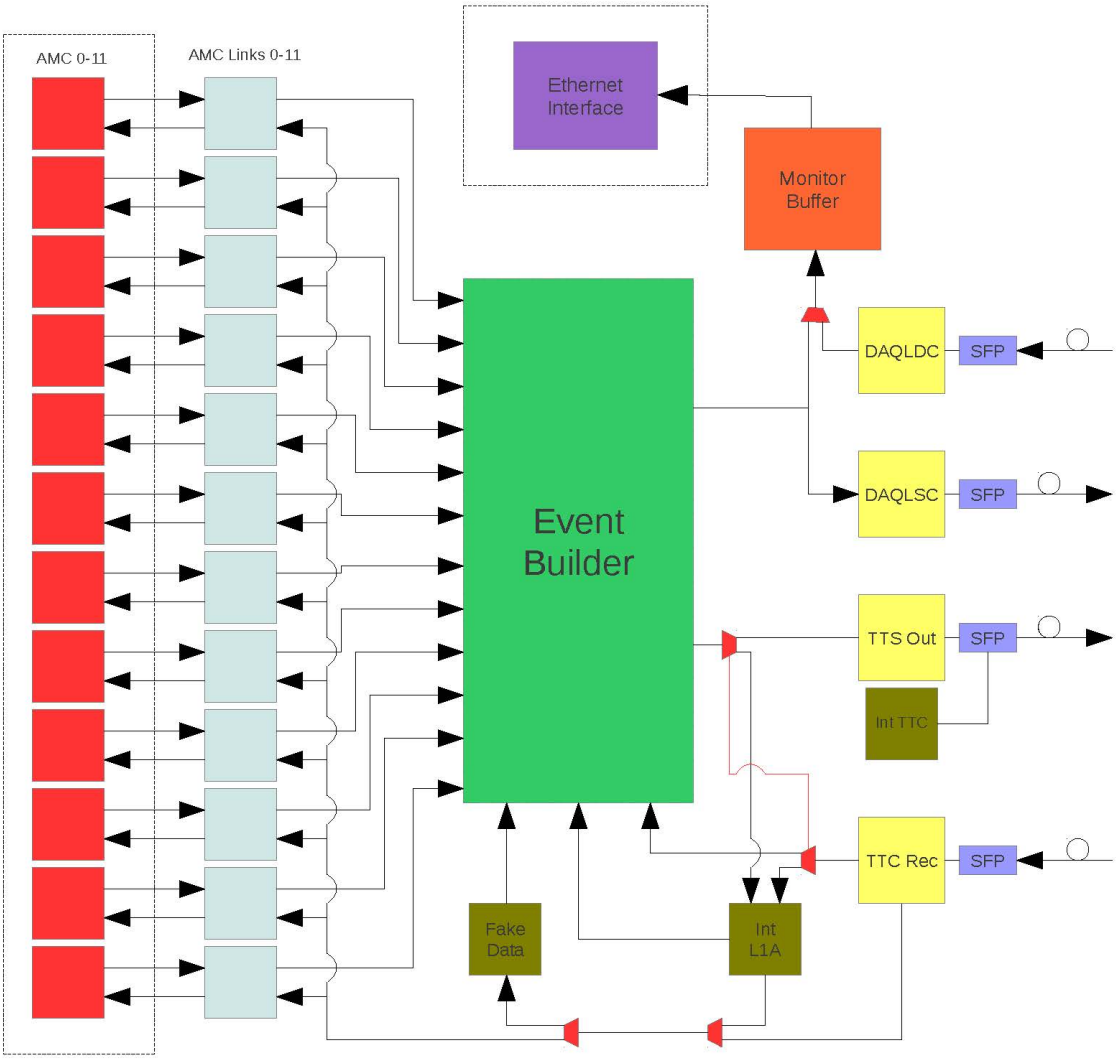
The DAQLSC Link of the AMC13XG operates at a speed of 5 Gbp/s using 8b/10b encoding (MAADI, 2015) is a line encoding technique that transforms an 8-bit byte into a 10-bit symbol to maintain a balance of transmitted zeros and ones and thus allows direct current control.

It is effectively providing an actual data capacity of 4 Gbp/s or 500 Megabyte per second (MB/s). Hence, the maximum sustained write clock to the link is 62.5 MHz (16 ns per 64-bit word).

The DAQ Link input buffer features four 4k bytes pages, with one page allocated per event. The objective is to encapsulate and embed the Control Board's Data Frame into

```
1 Monitor_Buffer| COUNT|   SFP0| SFP2| VALUE|
2               --|------|------|-----|------|
3      AVAILABLE|      |      |     |     1|
4          EMPTY|      |      |     |     0|
5       EOI_TYPE|      |      |     |     0|
6  EVT_AFTER_EOI|      |      |     |     0|
7     UNREAD_BLK| 0x3E8|      |     |      |
8          WORDS|      | 0x212|  0xE|      |
9
```

(a) Enable ports SFP in AMC13XG.



(b) Illustration AMC13 Block Diagram to out data by DAQLSC mode.

the DAQ Link's Data Frame. This process adheres to the specifications of Event creation and the data transmission capacity of the AMC13XG.

In the ongoing analysis of the results, Figure 51 illustrates the outcome of enabled SFP+ ports. The highlighted SFP+2 port signifies the transmission of a trigger signal,

```
          SFP_ROM|       SFP0|       SFP1|    SFP2|     TTC|
              --|-----------|-----------|--------|--------|
         ROM_DATA| 0x10070403| 0x10070403| 0x70403| 0x70403|

    Slink_Express| L|          SFP0|          SFP1|      SFP2|
                 --|--|--------------|--------------|----------|
          ANY_DOWN| 1|              |              |          |
       BLOCKS_FREE|  |             0|             0|         0|
       BLOCKS_SENT|  |              |              | 0x   1B58|
  BLOCK_READY_MASK|  |             1|             1|         2|
   BLOCK_USED_MASK|  | 0x          D| 0x          D|         8|
       CMS_CRC_ERR|  | 0x    3E803E8| 0x     700007|          |
       CORE_STATUS|  | 0x     12001D| 0x     12001D| 0x  40B28|
    EVENT_LGTH_SUM|  | 0x3E800000000| 0x3E8000003E8| 0x    3E8|
          EXPERT_1|  | 0x         14|             0|         0|
          EXPERT_2|  |             0|             0|         0|
       INITIALIZED|  |             0|             0|         0|
       LINKNOTFULL|  |             0|             0|         0|
           LINK_UP|  |             0|             0|         1|
   NO_BACKPRESSURE|  |             0|             0|         0|
      PACKETS_RCVD|  | 0x   5D9481BF| 0x   5D9482DF| 0x  40B28|
          REVISION|  |             0|             0|         0|
       RX_SIG_LOST|  |             1|             1|         1|
      SFP_LSC_DOWN|  |             1|             1|         1|
          STATUS_D|  |              |              | 0x    3E8|
         SYNC_LOSS|  |              | 0x        3E8| 0x3E803E8|
         TEST_MODE|  |             0|             0|         0|
        WORDS_SENT|  |              |              | 0x    3E8|
```

Figure 51 – Status generated by Slow Controller code at level 9 using the AMC13XG driver.

whereby the digital logic of the AMC13XG facilitated the sending of a data packet stored in the internal registers of T1 to out.

AMC13XG data is internally stored as 64-bit words, albeit with the need for a revision on the specifics of this process. The data is conveyed to the DAQLSC link firmware either through the SFP+2 port or another AMC card (via the backplane, such as the uHTR card), utilizing a 16-bit stream. Subsequently, the data is extracted from the AMC13XG memory buffer as 32-bit words via IPBus.

Upon commissioning the fiber optic DAQLSC link, it will transmit data in 64-bit words, thus defining the top-level format based on this word length. The packing arrangement is outlined in Figure 52, which illustrates the analysis of registers in T1 after a trigger of 3E8Hex (1000Dec) samples. Each line encompasses 128 bytes of data transmitted, divided into two blocks of 64 kilobytes when sent via the DAQLSC optical link.

The data with addresses of the 0x140$_{Hex}$ - 0x15F$_{Hex}$ SFP+2 ROM data (first 128

```
1 0x140:00070403 20000000 01150c40 0000002b 0000070f 47415641 2020204f 20202020
2 0x148:20202020 6a170000 52424641 5237352d 5a454135 20202020 20202020 90005203
3 0x150:00001a00 31313941 4b443933 20204236 20202020 30313131 20203430 e901f068
4 0x158:00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                                                                Last address 0x15F
```

Figure 52 – Data taken from the acquisition by the Slow Controller containing the addresses and values of the registers stored in the FPGA T1.

bytes, little endian) registers were intentionally extracted, which deals with the exchange of data with the 8b/10b encoding.

This address range (0x140$_{Hex}$-0x15F$_{Hex}$) is used in this firmware version for data exchange with little endian encoding. The Slow Controller using this specific address in T1 has the advantage of the security certification of sending 120-bit commands to be received by the Control Board.

A standard frame AMC13XG Output Data Format spends a minimum of 8 bytes up to 1760 bytes in one event trigger. Considering the worse case, in one event trigger (in data receiver mode), the start receives 4096 bytes per page, less 1760 bytes of standard signals from TCDS (heads and time stamps), the Slow Controller project can operate in the spare space of frame with 2336 bytes remaining the data frame.

The Slow Controller can be framed in just one T1 address, in the width of 32 bytes (there is still 8-bit left), so for a trigger with 4096 bytes, the Slow Controller uses only 32 bytes. Equation 3 shows the total amount of 4064 bytes left per page.

$$\text{Bytes to Slow Controller} = 4096_{bytes/page} - 1760_{bytes\ standart} = 2336_{bytes\ avaliable} \quad (3)$$

The Slow Controller solution concatenates the 120-bit + 8-bit (not used) in the word transmitted per address (2 words of 64-bit), which by choice, uses the address range 0x140$_{Hex}$-0x15F$_{Hex}$. However, there is the possibility to use other addresses to send commands to the Control Board without the security of little-endian encoding.

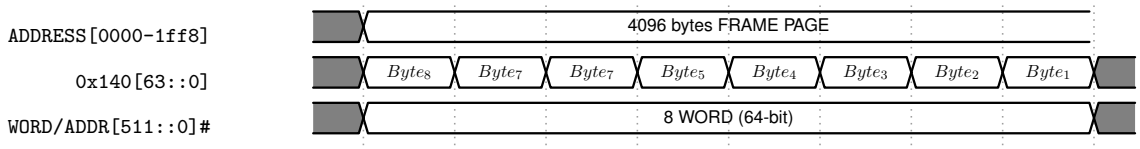


Figure 53 – Register by Dump 1 from FPGA T1 Registers.

In Figure 53, the Slow Controller utilizes the address 0x140$_{Hex}$ within a frame that contains 4096 bytes as an instance of storing the bits required for monitoring and controlling the Control Board. The conclusion highlights the use of 8 bytes in two 64-bit words.

In the context of a laboratory prototype, the initial condition posits a transmission time of 6 ns per meter of optical fiber. Regarding data transmission, be it sending or receiving, we expect a round-trip latency of 12 ns per meter due to fiber travel.

It is reasonable to deem the time taken by the acknowledgment send-and-receive logic negligible, using 2-meter optical fiber in the experimental setup of the Slow Controller.

Therefore, we can write three words to the DAQLSC Link buffer, equating to 24 Bytes, which is only half the size of an event header for every 4 meters of fiber.

In the final phase of the Slow Controller development, the objective is to employ 90 meters of fiber in a realistic scenario. Consequently, during the waiting period for an event to be transmitted over the fiber, writing 540 bytes of data to the DAQLSC Link buffer becomes feasible. However, it is worth noting that this amount remains relatively small compared to the 2336 bytes typically available in a standard support event frame transmission, regardless of whether it is initiated by a local trigger or provided by the TCDS System.

5.5- Application Development

In the CentOS 7 Linux version, several XDAQ packages share functionality with the Slow Controller project, which leads to a complete XDAQ installation without any previous XDAQ software version.

Figure 54 illustrates a successful test of the GUI utilizing the web mode.

This achievement is notably significant for the subsequent phases of the Slow Controller, as this web interface serves as an effective method for exchanging monitoring and control information for the LBB, which is currently under development in this project.

The software relies on a submodule called hyperdaq, which utilizes XML to store configuration data in a table format.

This storage capability can be integrated into the Slow Controller by incorporating

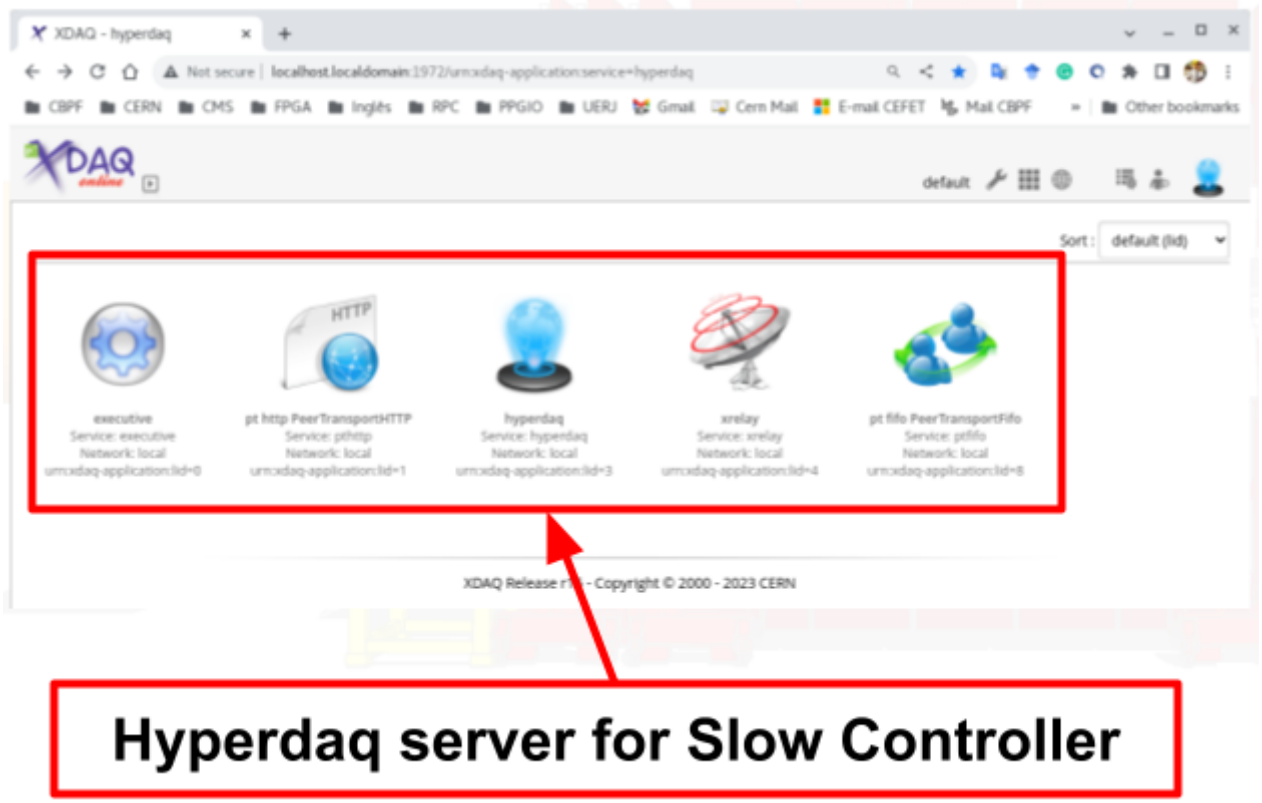


Figure 54 – Hyperdaq runs in PC Host for Slow Controller.

commands from the hardware access driver of the T1 and T2 FPGAs into XML tables, operating as a ”look-up table” via the web interface.

Figure 55 show one of many configuration files of the AMC13XG that accesses FPGA T1. It enables numerous internal bits to change modes (reading or writing), thus permitting the software driver to facilitate reading and writing more optimized and quicker.

Lines 5 and 6 of the XML file utilized by the AMC13XG primarily encode several bits to activate the DAQLSC optical link for data transmission or reception.

Implementing XDAQ as a Service software (XaaS) within the Slow Controller allows the system to be remotely accessed and configured via a cloud-based infrastructure.

These three non-operational from XDAQ packages: Core, Worksuite, and XaaS, will offer enhanced flexibility and scalability to replicate the Slow Controller solution 216 times for all data acquisition systems locally hosted on the New RPC Link System.

Moreover, XaaS enables the deployment of new services and upgrades and simplified maintenance and support, providing an efficient approach to managing the Slow Controller solution.

```
1 <node id="SFP0">
2         <node id="ROM" address="0x00000100" permission="r" description="SFP0
     ROM data(first 128 bytes, little endian)" size="0x1f" parameters="Table=
    SFP_ROM;Row=ROM_Data;Column=SFP0;Status=9"/>
3         <node id="RX_SIG_LOST_MASK" address="0x00000004" permission="r"
    description="'1' indicates SFP0..2 Receiver signal lost" parameters="
    Table=Slink_Express;Row=RX_SIG_LOST;Column=SFP0;Status=1;Show=nz" mask="0
    x00000010"/>
4         <node id="SFP_ABSENT_MASK" address="0x00000004" permission="r"
    description="'1' indicates SFP0..2 absent" parameters="Table=
    Slink_Express;Row=SFP_ABSENT;Column=SFP0;Status=1;Show=nz" mask="0
    x00000001"/>
5         <node id="SFP_LSC_DOWN_MASK" address="0x00000003" permission="r"
    description="'1' when DAQLSC of SFP0..2 is down" parameters="Table=
    Slink_Express;Row=SFP_LSC_DOWN;Column=SFP0;Status=1" mask="0x10000000"/>
6         <node id="TX_FAULT_MASK" address="0x00000004" permission="r"
    description="'1' indicates SFP0..2 TxFault" parameters="Table=
    Slink_Express;Row=TX_FAULT;Column=SFP0;Status=1;Show=nz" mask="0x00000100
    "/>
7       </node>
```

Figure 55 – File XML from AMC13XG configuration for FPGA T1.

Conclusions

Initial Considerations

The initial plan consolidated the project's development within the confines of the CERN laboratories. Regrettably, executing the project solely at CERN proved unfeasible. Nevertheless, the project team facilitated two short-stay research visits to investigate the Slow Controller's integration with the New Link Board and the New Back-End System in the CMS Muon System.

In response to the global pandemic emergencies that necessitated national and international travel restrictions, certain aspects of this research underwent modification for facilitated the efficient consolidation of research elements in software, hardware, and firmware.

Despite numerous challenges, the research job adhered to the planned trajectory and maintained the integrity of the work. However, completion still awaits certain sections, emphasizing specific aspects of the hardware in the following phases.

The hardware documentation in practice is highly sparse and lacks a standard. There is only a non-official website with little accurate information and no detailed software and hardware version change records.

The entire development of this work results from reverse engineering in advanced digital systems where they were studied and tested associated with TCP-IP protocol standards together with the minimum documentation[3] of the AMC13XG observing the comments of previous projects.

This research concluded that groups involved in hardware and software documentation on the AMC13XG discontinued this activity due to new possibilities to adopt a new standard, ATCA, while the current hardware operates in other experiments.

At the project's onset, the Electronic Coordination of CERN did not anticipate a connection between the Slow Controller and the intricate TCDS system. Nevertheless, as the research progressed, the Electronic Coordination of CERN recognized the necessity for

[3] `http://www.amc13.info`

minimal integration and compatibility to facilitate subsequent stages of the Slow Controller's development.

While the software facilitated the Slow Controller's integration with the TCDS System, it warrants further refinement. Implementing additional modules, like the uHTR card that supplies support for a standard uTCA CMS system, becomes essential to bolster this integration further.

The Research Successful

The development detailed in this thesis aligns with the main objectives of the LS2 upgrade for the Muon Detector System, addressing the aspect of data transmission and reception from the project subsystem's small-scale Slow Controller design.

Our current research exemplifies the potential of these implementations to construct a reliable system for real-time monitoring equipped with feedback mechanisms to maintain function or troubleshoot any issues as they arise.

The employed technologies offer a data transmission rate ranging from 1.6 Gbp/s to 10 Gbp/s using optical fibers, aligning with the data transmission rate upgrades for the HL-LHC.

The newly developed Slow Controller satisfies the requirements of the RPC Link System for high, uninterrupted, and secure data transmission, preventing data loss. In addition to adhering to the CERN specification pattern, the Slow Controller is a development to function effectively in a high-radiation environment.

The first version of the Slow Controller for small-scale system monitoring/control works like the functional prototype in the lab. This assembly possesses all necessary properties for phase 2 integration into the medium-scale system, aiming for a final large-scale system connection with 216 Control Board units.

Innovations in the Theme

Another notable outcome of this work is that the Slow Controller, apart from being developed for compatibility, utilizes all hardware already consolidated and certified by CERN. This hardware is resistant to radiation and aligns entirely with the standard systems currently employed in the CMS Experiment. Furthermore, the Slow Controller is replicable for the 216 optical links of the New Link System control boards in the final version.

Based on the tests conducted with the AMC13XG modules, the results suggest the feasibility of transferring TCDS signals (TTC/TTS) to the uTCA's backplane and distributing them to all the connected modules. The Slow Controller can also share the TCDS signal in real-time via SFP+ optical connections.

This functionality is crucial for a control system synchronizing the LHC via the General Slow Controller system, enabling real-time monitoring and instantaneous control of the New Link System for 216 independent systems. Each system possesses its specific configuration accessible via the web server if necessary.

The Slow Controller has a relevant contribution to science because, with the refinement of data communication control, it contributes to generating more accurate results in monitoring and thus offers results not achieved before in this area.

Future Works

During the first phase of the prototype, the decision was made not to utilize the remote firmware write features. However, the final prototype system facilitates the remote reconfiguration of FPGAs without service interruption, given that they need to insert into the uTCA crate through TCP-IP communication links.

By employing CERN standard systems, the Slow Controller project achieves comprehensive integration of the Back-End system and TCDS with the final version of the Slow Controller.

The Slow Controller is adaptable for introducing new features to rigorously adjust AMC13XG's optical data transmission settings. It is capable of implementing additional

software configurations within the system. This adaptability is crucial for efficient data transmission to and from the Control Board.

An additional AMC card, the uHTR from HCAL, is presently under review for possible integration with the Slow Controller. The selection of this card is attributable to its compatibility, coupled with its capacity to extend SFP+ optical links, distribute signals via uTCA, and control firmware through TCP-IP.

The Slow Controller offers a service of regular firmware updates that are essential to align with the predefined requirements. This alignment, extending from the prototype stage to the final product, ensures optimal performance throughout the system's development.

References

AAD, Georges et al. Combined Measurement of the Higgs Boson Mass in pp Collisions at $\sqrt{s} = 7$ and 8 TeV with the ATLAS and CMS Experiments. **Phys. Rev. Lett.**, American Physical Society, v. 114, p. 191803, 19 2015. DOI: 10.1103/PhysRevLett.114.191803.

AAD, Georges et al. The ATLAS experiment at the CERN large hadron collider. **Jinst**, v. 3, s08003, 2008.

ABBRESCIA, M et al. Local and global performance of double-gap resistive plate chambers operated in avalanche mode. **Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment**, v. 434, n. 2, p. 244–253, 1999. ISSN 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(99)00534-3.

AGUILAR-SAAVEDRA, Juan Antonio et al. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. **Physics Letters B**, v. 716, n. 1, p. 1–29, 2012. ISSN 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2012.08.020.

ALIBI, Mohamed; ROY, Bhaskarjyoti. **Mastering CentOS 7 Linux Server**. 35 Livery Street, Birmingham B3 2PB, UK: Packt Publishing Ltd, 2016.

ANDERSON, Jake; FREEMAN, Jim; WHITMORE, Juliana. Progress on the Upgrade of the CMS Hadron Calorimeter Front-End Electronics. **Physics Procedia**, v. 37, p. 190–197, 2012. Proceedings of the 2nd International Conference on Technology and Instrumentation in Particle Physics (TIPP 2011). ISSN 1875-3892. DOI: https://doi.org/10.1016/j.phpro.2012.03.708.

ASILAR, Ece. **Performance of improved RPCs demonstrator for the CMS Phase 2**. Geneva, 2022.

BARNEY, David. **CMS Slice**: A few variations of the CMS slice image including larger transverse/longitudinal and 3-D views. Dutch, French, German. Ed. by CMS-OUTREACH-2018-017. Version 1. A few variations of the CMS slice image including larger transverse/-

longitudinal and 3-D views. CMS Communications Team. 16 Feb. 2015. Available from: ¡https://cds.cern.ch/record/2628641¿. Visited on: 30 Nov. 2020.

BERRIAUD, Damien. Study of HTTP technologies in XDAQ 2nd generation environment. CERN Document Server, 1211 Geneva 23, Switzerland, 2022.

BERTI, L. et al. Using XDAQ in Application Scenarios of the CMS Experiment. arXiv, 2003. DOI: 10.48550/ARXIV.HEP-EX/0305076.

BOGHRATI, B et al. CMS phase-II upgrade of the RPC Link System. **Journal of Instrumentation**, IOP Publishing, v. 16, n. 05, p. c05003, 2021.

BOGHRATI, B. et al. New slow control emulator for the CMS phase-2 upgrade RPC link system. **Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment**, v. 1052, p. 168268, 2023. ISSN 0168-9002. DOI: https://doi.org/10.1016/j.nima.2023.168268.

CAO, P et al. Research and development of the back-end electronics for the two-dimensional improved resistive plate chambers in CMS upgrade. **Radiation Detection Technology and Methods**, Springer, v. 5, n. 2, p. 181–191, 2021.

CHATRCHYAN, Serguei et al. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. **Physics Letters B**, Elsevier, v. 716, n. 1, p. 30–61, 2012. ISSN 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2012.08.021.

CMS. **The CMS detector uses a huge solenoid magnet to bend the paths of particles from collisions in the LHC**. CERN. 2020. Available from: ¡https://home.cern/science/experiments/cms¿. Visited on: 24 Nov. 2020.

COLLABORATION, CMS. CMS MUON Project Technical Design Report. **Preprint**, p. 97–32, 1997.

COLLABORATION, CMS et al. A new boson with a mass of 125 GeV observed with the CMS experiment at the Large Hadron Collider. **Science**, American Association for the Advancement of Science, v. 338, n. 6114, p. 1569–1575, 2012.

COLLABORATION, The CMS. **Technical proposal for the upgrade of the CMS detector through 2020**. Geneva, 2011.

COLLABORATION, The CMS. The performance of the CMS muon detector in proton-proton collisions at $\sqrt{s} = 7$ TeV at the LHC. **Journal of Instrumentation**, IOP Publishing, v. 8, n. 11, p11002–p11002, 2013. DOI: 10.1088/1748-0221/8/11/p11002.

COMMITTEE, The LHC experiments. **Letter of Intent for the Phase-I Upgrade of the ATLAS Experiment**. Geneva, 2011.

COSTANZA, F. et al. The next generation Front-End Controller for the Phase-I Upgrade of the CMS Hadron Calorimeters. **Journal of Instrumentation**, IOP Publishing, v. 12, n. 03, p. c03003–c03003, 2017. DOI: 10.1088/1748-0221/12/03/c03003.

DENG, Binwei et al. Design and hardware evaluation of the optical-link system for the ATLAS Liquid Argon Calorimeter Phase-II Upgrade. **Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment**, v. 981, p. 164495, 2020. ISSN 0168-9002. DOI: https://doi.org/10.1016/j.nima.2020.164495.

DOOLING, Samantha Katherine. **Differential Cross Section Measurement of Drell-Yan Production and associated Jets with the CMS Experiment at the LHC**. 2015. presented 17 Nov 2014.

EUROPE, N.A.T. **NAT-MCH – User's Manual**. Ed. by N.A.T. Europe. Konrad-Zuse-Platz 9, 53227 Bonn-Oberkassel, 2023.

EVANS, Lyndon; BRYANT, Philip. LHC Machine. journal of instrumentation, 3. **S08001**, 2008.

FRAZIER, R et al. The IPbus Protocol, An IP based control protocol for ATCA/TCA. **CERN, Switzerland**, CERN, Switzerland, Meyrin, Esplanade des Particules 1, 2013.

FRAZIER, Robert et al. Software and firmware for controlling CMS trigger and readout hardware via gigabit Ethernet. **Physics Procedia**, v. 37, p. 1892–1899, 2012. Proceedings of the 2nd International Conference on Technology and Instrumentation in Particle Physics (TIPP 2011). ISSN 1875-3892. DOI: https://doi.org/10.1016/j.phpro.2012.02.516.

GMBH, N.A.T. **NAT MicroTCA Carrier Hub (NAT-MCH)**. Ed. by N.A.T. GmbH. N.A.T. GmbH. 2023. Available from: ¡https://www.picmg.org/spec-product/nat-microtca-carrier-hub-nat-mch-2/¿. Visited on: 27 Feb. 2023.

GOLA, Mohit. CMS phase-II upgrade with GEM detector. **Journal of Physics: Conference Series**, IOP Publishing, v. 1498, n. 1, p. 012054, 2020. DOI: 10.1088/1742-6596/1498/1/012054.

GREGORY, Barrie Archie. **An introduction to electrical instrumentation and measurement systems: a guide to the use, selection, and limitations of electrical instruments and measurement systems**. Second Edition. Houndmills, Basingstoke, Hampshire RR21 2XS: Macmillan International Higher Education, 1981. 445 pp. ISBN 978-0-333-29384-3. DOI: 10.1007/978-1-349-16482-0.

HAZEN, E et al. The AMC13XG: a new generation clock/timing/DAQ module for CMS MicroTCA. **Journal of Instrumentation**, v. 8, n. 12, p. c12036, 2013. DOI: 10.1088/1748-0221/8/12/C12036.

HAZEN, Eric. **CMS AMC13 Hub Card**. Chris Lawlor 590, Commonwealth Ave Physics Rm 255 Boston, MA 02215, 2012.

HEGEMAN, Jeroen et al. The CMS Timing and Control Distribution System. In: 2015 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC). San Diego, CA, USA: IEEE, 2015. P. 1–3. DOI: 10.1109/NSSMIC.2015.7581984.

HILL, Charlie. **AMC13 XG Operational Specification**. Chris Lawlor 590, Commonwealth Ave Physics Rm 255 Boston, MA 02215, 2012.

HOLZNER, Andre. **My notes on XDAQ/XMAS monitoring**. Accessed: 2023-03-15. 2009. Available from: ¡https://twiki.cern.ch/twiki/bin/view/Main/AndreHolznerXdaqMonitoring¿.

ILES, Gregory et al. CMS MicroTCA crate concepts & AMC card requirements. **Version 0.9**, 2011.

LARREA, C. Ghabrous et al. IPbus: a flexible Ethernet-based control system for xTCA hardware. **Journal of Instrumentation**, IOP Publishing, v. 10, n. 02, p. c02019–c02019, 2015. DOI: 10.1088/1748-0221/10/02/c02019.

LEAL REYES, Nadia Mariana. Development of online software tools for Precision Proton Spectrometer (CMS-PPS/CERN), 2021.

MAADI, Mohammad. An 8b/10b Encoding Serializer/Deserializer (SerDes) Circuit for High Speed Communication Applications Using a DC Balanced, Partitioned-Block, 8b/10b T. **International Journal of Electronics and Electrical Engineering**, v. 3, p. 144, Apr. 2015. DOI: 10.12720/ijeee.3.2.144-148.

MANS, J; FRAHM, E. Status report on a MicroTCA card for HCAL trigger and readout at SLHC. **Journal of Instrumentation**, IOP Publishing, v. 5, n. 12, p. c12027–c12027, 2010. DOI: 10.1088/1748-0221/5/12/c12027.

MOREIRA, Paulo. The GBT, a proposed architecture for multi-Gbps data transmission in high energy physics, p. 71, Sept. 2007. CERN, 1211 Geneva 23, Switzerland.

PAOLUCCI, P et al. **The Detector Control Systems for the CMS Resistive Plate Chamber**. Geneva, 2010. DOI: 10.1088/1742-6596/219/2/022019.

POY, A Barriuso et al. The detector control system of the ATLAS experiment. **Journal of Instrumentation**, IOP Publishing, v. 3, n. 05, p05006, 2008.

RAUSCH, R; TAVLET, Marc. On-Line Radiation Test Facility for Industrial Equipment needed for the Large Hadron Collider at CERN, 1999.

SHCHABLO, Konstantin et al. Front-end electronics for CMS iRPC detectors. **Journal of Instrumentation**, IOP Publishing, v. 16, n. 05, p. c05002, 2021. DOI: 10.1088/1748-0221/16/05/C05002.

THE CMS COLLABORATION, et al. The CMS experiment at the CERN LHC. **Journal of Instrumentation**, IOP Publishing, v. 3, n. 08, s08004–s08004, 2008. DOI: 10.1088/1748-0221/3/08/s08004.

THE LHC EXPERIMENTS COMMITTEE. **The Phase-2 Upgrade of the CMS Muon Detectors**. Geneva, 2017. This is the final version, approved by the LHCC.

THYSSEN, F. Front-End control and monitoring system for the Resistive Plate Chambers at the CMS experiment. In: 2011 IEEE Nuclear Science Symposium Conference Record. Valencia, Spain: IEEE, 2011. P. 1119–1123. DOI: 10.1109/NSSMIC.2011.6154585.

ZHANG, Jie; LI, Hangxu; JIANG, Xiaoshan. The MicroTCA Fast Control Board for Generic Control and Data Acquisition Applications in HEP Experiments. **IEEE Transactions on Nuclear Science**, v. 66, n. 7, p. 1169–1173, 2019. DOI: 10.1109/TNS.2018.2890284.