



Design, Development and Verification of the Detector Control System for the TOTEM experiment at the CERN LHC

Universidad de Sevilla

Departamento de Arquitectura y Tecnología de Computadores



Fernando Lucas Rodríguez
June 2009



(This page is intentionally left blank)

Universidad de Sevilla Departamento de
Arquitectura y Tecnología de Computadores

Tesis Doctoral

Design, Development and Verification of the Detector Control System for the TOTEM experiment at the CERN LHC

Memoria presentada para aspirar al grado de doctor
según la normativa de Doctorado Europeo
por

Fernando Lucas Rodríguez
Ingeniero en Informática

Los profesores ponentes y directores:

Dr. Anton Civit
Profesor Titular
Departamento de
Arquitectura y Tecnología de Computadores

Dr. Gabriel Jiménez Moreno
Profesor Titular
Departamento de
Arquitectura y Tecnología de Computadores

June 2009

Revisions record

Version	Date	Changes
0.1	2007/12/22	Followup of the DEA
0.2	2008/05/15	Include chapters for RADMON, ELMB and connectivity
0.3	2008/08/01	Reorganization of contents
0.4	2008/11/01	Explain atomization scripts for pinout and motorization
0.5	2008/12/15	First draft for restricted review
0.6	2008/01/10	Complete draft for open review
1.0	2009/03/01	Final version

Revision: 2516

Date: 2009-05-06 08:43:41Z

Latest revision available online from
<http://info.fernandolucas.es/totemdcs/>

This thesis was typesetted with $\text{\LaTeX}$, using with the help of some extra packages such as the memoir class. All the $\text{\TeX}$ files and images were managed using SVN.

The fonts used are TrueType and Type1, various sizes but embedded into the PDF, and the layout is A4 vertical double-side.

Most of the images are vectorial, allowing high resolution zoom for the graphic details, when using the electronic version of this document.



Abstract

This thesis details how the Detector Control System for the TOTEM experiment at CERN LHC is designed. The TOTEM experiment is composed by three detectors called 'Roman Pots', 'T1' and 'T2'. The controlled subsystems include environmental monitors, high voltage and low voltage power supplies, cooling plants,...

The thesis also shows the state of the art at CERN control systems. Comparisons with industry technologies and standards are done wherever possible. A novel way to define and process automatically the wire connectivity and the logic is presented. This representation is automatically processed. The data produced by the sensors, and the bus usage, is studied using an information theory approach. Temporal considerations about the readout of the sensors, data transmission and processing are taken into account.

Resumen

Esta tesis detalla como esta diseñado el Sistema de Control de Detectores del experimento TOTEM en el acelerador LHC del CERN. Dicho experimento incluye los detectores llamados 'Roman Pots', 'T1' y 'T2'. Algunos de los subsistemas a controlar son fuentes de alta y baja tensión, sensores ambientales, sistemas de refrigeración,...

Igualmente, se hace un estudio del estado del arte de los sistemas de control en el CERN, comparando en la medida de lo posible con desarrollos industriales y posibles estándares. Se presenta una formalización para el conexionado del experimento y la descripción de la lógica operativa. Dicha formalización es procesada automáticamente. También se detalla desde un punto de vista de teoría de la información generada en los diferentes sensores, la utilización de los buses de comunicación, el análisis temporal de dichos buses y las máquinas de estados que gestionan la lógica operativa.

(This page is intentionally left blank)

Part I

About the thesis

(This page is intentionally left blank)

Preface and Objectives

This manuscript gives a general overview of the TOTEM Detector Control System at the CERN LHC. It focuses on the design (and alternatives) and the development process, but also some aspects of verification are covered. I started this project from scratch in September 2006 and now a stable version of the systems is operative and installed in the TOTEM experiment. The work described in this thesis is the result of my work, the help of some collaborators is mentioned explicitly.

CMS and TOTEM have a common physics program; this is the reason why TOTEM must be able to operate independently, but also take data together with CMS. It is necessary to ensure interoperability at the level of the electronics, DAQ, DCS, run control, offline data processing, safety and so on. From this point of view, TOTEM seems a detector of CMS; but from other points of view it is fully independent experiment, or even a piece of the LHC.

However that interoperability applies only to the interface level with CMS or the LHC; not to the underlying subsystems. This thesis makes an in-depth study to analyze the controllability requirements and to evaluate the possible solutions for the novel machines and detectors developed in the context of the TOTEM experiment. It is not possible to make any assumption or simplifications in the subsystems basing on a formal model or previous experience. All the experiment operation, measurement parameters, sensors and actuators has to be defined almost from the scratch.

Two path were possible:

- Use similar technologies in software and hardware to the ones of CMS and other LHC experiments and improve if necessary.
- Use completely different technologies and provide a compatible communication interface.

The first idea was to use actual industrial solutions, see Chapter 8, *Comparison with commercial solutions*. But the lack of common practices in industry moved the effort into reusing the already tested CERN developments, hoping that after the LHC development a CERN continues providing proper support for all the selected technologies.

Most of the existing CERN developments, described in Chapter 5, *State of the art of the LHC control systems*, are *ad-hoc* solutions without any strong standardization. This has severe implications, such as the difficulty of finding people with proper skills, delivery times of new equipment (measured in years), long term maintenance and comprehensive testing.

Both alternatives provided almost the same benefits and drawbacks, but using common CERN technologies guaranteed the help of some support groups and a stronger feedback to manufactures. Also how most of the developments were done by CERN itself, when adapting or reusing existing developments the Intellectual Properties issues would be easier.

However, the choice and usage of those technologies will be different in TOTEM than if it were just another CMS detector. For any control function or problem domain a survey was done about how

that issue was addressed the other LHC experiments or industry; trying to take the best source of inspiration.

Problem domain	AB-CO	ATLAS	ALICE	CMS	ECSS	IT-CO	LHCb	TOTEM
Development process					X			X
Operational procedure								X
Requirements formalization			X					X
Code and documentation management								X
Development guidelines				X		X		
Cables pinout and connectivity								X
High Voltage						X		
Low Voltage						X		
DCU				X				
CAN-PC connectivity							X	
CAN-ELMB distribution								X
ELMB		X						
Radiation sensors system		X					X	X
Pressure sensors								X
Humidity sensors				X				
DIP protocol						X	X	
DIM protocol						X	X	
LHC integration						X		
User Interface			X			X		X
Supervisor technology (PVSS)						X		
Finite State Machines						X	X	
Motorization control	X							
Motorization integration						X		X
Information theory calculations								X
Project organization					X			X
Project planning								X
Configuration management					X			X

Table 0.1: Source of inspiration or technology provider for each problem domain

Table 0.1 resumes those sources:

- AB-CO is the LHC group responsible for the collimator control.
- ALICE experiment provides the best ‘engineering’ guidelines, that had been of great inspiration.
- ATLAS experiment provides technology but without any kind of support.
- CMS, in general, only provides generic integration requirements but without establishing any kind of procedure.
- ECSS is the European consortium for space standardization; provides organization and management templates.
- IT-CO is the support group at CERN for Control Systems developments.
- LHCb is the originator of some technologies.

The main contributions (marked as TOTEM in Table 0.1) of this thesis lies in the design of the Control System for movable parts called ‘Roman Pots’ (RP). Most of the experiments at CERN and other particle accelerators are fixed detectors, remaining in the same position during all the accelerator runs. But this is not the case for the RP, they have to stay at a fixed distance from the trajectory of particles moving almost at the speed of the light. A failure in the control system can lead to a serious damage for the Roman Pots but also for the accelerator itself as happened at Fermilab in 2002 [MCD⁺06].

Just this objective by itself is a real challenge. In general, the LHC and its experiments, use a set of completely different technologies without the need of interoperability. Usually they just exchange a small set of mixed hardware and software flags, as explained in Chapter 6, *Beam operation*.

The connectivity (hardware and software) must be represented through diagrams to clarify the dependencies and risks. In Appendix B, *Hardware overview diagrams*, Figure B.1 and Figure B.2 are

focused on hardware, while the operational logic of Chapter 12, *Integration with the motorization*, is represented with UML.

CERN selected PVSS as the main tool for the experiments control systems. This development in principle is highly manual, not suited to automatic generation of code, or any formal analysis. This situation lead to a series of novel developments to provide a way to automatically generate code and configurations for the DCS and related tools; they conform a substantial part of the research work of this thesis.

Summarizing, this thesis is not focused in developing any new technology, but the innovation comes on the following topics:

- Formalize the hardware and the interconnections, as well as the cardinality and location.
- Define uniform ways of expressing the requirements.
Some of those representations are automatically processed. The advantage of this approach is that the same document is used for reviewing the design from the engineering point of view, verifying the requirements with the physicist responsible of the detector and is automatically processed to generate the final configuration of the equipment.
- Develop specific tools (pinout tables, FSM hierarchy tables, automatic generation scripts and the Information Exchange Calculator tool) and configure the available ones (PVSS and JCOP framework).
- Specify the deadlines for temporal response. Study the reaction time and reach appropriate compromise between elements and to monitor and response time.
- Study the impact of all previous decisions in an estimated experiment operational life of 15 years.

The thesis is structured into five parts:

- Part I, *About the thesis*, provides information about how the thesis is structured and its purpose.
- Part II, *Environment and Precedents*, gives an overview how the Control Systems are built at CERN. It is a compilation of the main features from available technologies and their applications, within in CERN and other environments. It is important to make clear that this part is not the main subject of the thesis but is needed because there is no other document discussing the different CERN approaches for the development of a Detector Control System. Consequently this comparison and grouping of the technologies should be considered research too.
- Part III, *Requirements and Solutions: Big.Brother*, study the alternatives for building a Detector Control System for the TOTEM experiment.
This is the main point of the thesis; where the choices and comparison for the development process and technological alternatives and decisions for each DCS level are presented. The main topics include among others:
 1. Represent the layout of the experiment with a high level syntax.
 2. Adapt a development process matching the technical and human constrains.
 3. Define engineering documents being at the same time human readable and machine processable. They are used for expressing all the connectivity of the experiment and its operational logic.
- Part IV, *Appendices*, provides additional information about the hardware, auxiliary tools and some other topics.
- Part V, *References*, includes the publications, the bibliography and a glossary.

And the recommendations about how to read them are:

- Any of the colleagues involved in the TOTEM experiment can proceed directly to the third part, Part III, *Requirements and Solutions: Big.Brother*.

-
- Any other DCS developer interested in the TOTEM DCS can have an overview of the Appendices and proceed to the third part, Part III, *Requirements and Solutions: Big Brother*.
 - The rest of interested persons it is recommended to read the first and second part, then continue the appendices and finally the third part.

As detailed in [Org03a] for identification of research areas, this thesis could be classified into categories from paragraph [135] up to [142]. The main purpose of designing a DCS is not Research and Development by itself; however several novel activities of R&D must be carried out for its completion and it is a vital part of a Research and Development project.

Finally, I would like to give thanks to my official supervisors Ernst Radermacher, Anton Civit Balcells, Gabriel Jiménez Moreno. Additionally Paolo Palazzi, as the TOTEM DCS coordinator, has provided innumerable suggestions. All of them allowed me to fulfil my PhD in this subject, and they always have been always available to questions.

I would also like to mention Gueorgui Antchev, Ivan Atanasov, Evangelia Dimovasili, Mario Deile, Mathias Philippe Dutour, Rauno Lauhakangas, Hubert Niewiadomski, Risto Orava, Marco Oriunno, Federico Ravotti, Gennaro Ruggiero, Walter Snoeys and Nicola Turini. They are the responsible for many of the TOTEM subsystems that the DCS needs to interact with. They have provided the requirements for the Detector Control System.

Outside of the TOTEM collaboration I would like to give thanks to Robert Gomez-Reino Garrido and Frank Gledge from the CMS-DCS central group and André Augustinus as the ALICE DCS ‘chief’ engineer. They have done ‘peer’ reviews to many of my proposals providing valuable comments.

Nothing lasts... but nothing is lost

List of contents

I	About the thesis	1
	Preface and Objectives	3
	List of contents	7
II	Environment and Precedents	15
1	CERN	17
1.1	An historical introduction	17
1.1.1	Countries	17
1.1.2	Accelerators	17
1.2	Current status: LHC era	18
1.3	Future: An upgrade for the LHC	19
2	The Large Hadron Collider	21
2.1	The LHC machine	21
2.1.1	Parameters	21
2.1.2	Bunches	22
2.1.3	Sectors and octants	22
2.1.4	Accelerator components	22
2.1.5	BLM (Beam Loss Monitors)	24
2.1.6	BPM (Beam Position Monitors)	25
2.1.7	Collimators	25
2.2	The LHC experiments	26
2.2.1	ALICE	26
2.2.2	ATLAS	26
2.2.3	CMS	26
2.2.4	LHCb	27
2.2.5	LHCf	27
2.2.6	TOTEM	28
3	The TOTEM experiment	29
3.1	Introduction	29
3.2	The physics programme	31
3.2.1	Total cross-section	31
3.2.2	Elastic scattering	31

3.2.3	Diffraction and the pomeron	32
3.3	RpMe (Roman Pot Mechanics)	33
3.3.1	System strategy and overview	33
3.3.2	The movements	34
3.4	RpSi (Roman Pot Silicon detector)	35
3.4.1	System strategy and overview	35
3.4.2	Current terminating structures	35
3.4.3	On-detector electronics	36
3.5	T1 (Telescope 1)	37
3.5.1	Detector geometry	37
3.5.2	Description of the T1 CSC detectors	38
3.6	T2 (Telescope 2)	39
3.6.1	Detector geometry	39
3.6.2	Description of the T2 GEM detectors	40
3.7	The TOTEM radiation environment	41
3.7.1	Summary of simulation results	41
3.7.2	Roman Pots radiation environment	41
3.7.3	T1 radiation environment	41
3.7.4	T2 radiation environment	41
4	Data flow	43
4.1	Readout	43
4.2	On line	44
4.2.1	Architecture	44
4.2.2	DAQ (Data Acquisition Components)	44
4.3	Databases	45
4.3.1	TOTEM data architecture	45
4.3.2	Oracle 11g	45
4.4	Off line	47
4.4.1	The grid	47
4.4.2	Reconstruction	48
5	State of the art of the LHC control systems	51
5.1	Introduction	51
5.1.1	Objectives	51
5.1.2	Requirements	52
5.2	Control functions level	53
5.2.1	High voltage control	53
5.2.2	Low voltage control	53
5.2.3	Environment monitoring	53
5.2.4	Cooling monitoring	53
5.2.5	Gas control	53
5.2.6	LHC status monitoring	54
5.2.7	DSS monitoring	54
5.2.8	Rack control	54
5.2.9	VME crates control	54
5.2.10	Access control	54
5.3	Communications level	55
5.3.1	DIM (Distributed Information System)	55
5.3.2	DIP (Distributed Interchange Protocol)	55

5.3.3	OPC (OLE for Process Control)	55
5.3.4	XDAQ (Cross-platform DAQ framework)	56
5.3.5	CMW (Controls MiddleWare)	56
5.4	Supervisory level	56
5.4.1	Introduction	56
5.4.2	PVSS (Process Visualization and Control System)	57
5.4.3	JCOP (Joint COntrols Project) FrameWork	57
5.4.4	SMI++ (State Management Interface)	58
5.4.5	Command hierarchy	58
5.4.6	Partitioning	60
5.5	Databases	60
5.5.1	Conditions database	60
5.5.2	Configuration database	61
5.5.3	Geometry database	61
5.6	CERN Control Centre (CCC)	61
6	Beam operation	63
6.1	Beam optics	63
6.2	The LHC Machine Modes	64
6.2.1	ADJUST mode	64
6.2.2	STABLE_BEAMS mode	65
6.2.3	UNSTABLE_BEAMS mode	65
6.2.4	BEAM_DUMP	65
6.2.5	Mode Transitions	66
6.3	Beam Instrumentation Signals	66
6.4	Beam Interlocks	67
6.4.1	LHC Beam Interlock System (BIS)	67
6.4.2	TOTEM Beam Interlocks	68
III	Requirements and Solutions: Big.Brother	69
7	Thesis planning	71
7.1	Evolution of the thesis in relation with the TOTEM experiment	71
7.2	Feedback systems	72
7.2.1	Generic system	72
7.2.2	System identification	74
7.3	Hardware overview	75
7.4	Requirements formalization	76
7.4.1	Templates	76
7.4.2	Detector Product Breakdown Structure (PBS)	76
7.4.3	Naming Scheme	77
7.4.4	Ordering	79
7.4.5	Mapping	79
7.5	Development process	80
7.6	Packaging	83
8	Comparison with commercial solutions	85
8.1	Levels	85
8.2	Front end level	86

List of contents

8.2.1	Physical Level	86
8.2.2	Application Level	87
8.3	Middleware level	88
8.3.1	OPC Unified Architecture (OPC-UA)	88
8.3.2	Simple Network Management Protocol Version 3 (SNMPv3)	89
8.4	Supervisory level	89
8.4.1	LabVIEW	89
8.4.2	CS-Framework (Control System Framework)	89
8.5	Management aspects	90
8.5.1	Introduction	90
8.5.2	Capability Maturity Model Integrated (CMMI)	91
8.5.3	European Cooperation for Space Standardization (ECSS)	91
8.5.4	Development	92
8.5.5	Software Configuration Management Plan (SCMP)	93
8.5.6	Conclusions	94
9	Frontend sensors	95
9.1	Sensors types	95
9.1.1	Temperature	95
9.1.2	Pressure	96
9.1.3	Humidity	97
9.1.4	RADiation MONitors sensor	97
9.2	Measurement locations	100
9.2.1	Roman Pots	100
9.2.2	Telescope T1	101
9.2.3	Telescope T2	101
9.3	Distribution	103
9.4	RADMON readout implementation	104
9.4.1	Readout of the radfets and p-i-n diodes	104
9.4.2	Readout of the temperature	104
9.4.3	Readout of the return line	105
9.4.4	Constraints of the readout	105
9.4.5	Use case	105
9.4.6	Readout rate	106
9.5	Instrumentation problems	107
10	Connectivity	109
10.1	Introduction	109
10.2	Cabling	110
10.2.1	Available cables	110
10.3	Pinout tables and hardware generation scripts	111
10.4	RADMON Connectivity into the DCS readout electronics	115
10.4.1	The ELMB system connectivity	115
10.4.2	RADMON Patch Panel board design	116
10.5	Rack	117
10.5.1	ELMB and ELMB-DAC powering	117
10.5.2	ELMB power relays	118
10.5.3	CAN bus chain	119
10.5.4	DCS rack interconnections	120
10.6	Equipment used	124

10.7	Networks interconnection	124
11	Operational logic	127
11.1	Introduction to the FSM trees	127
11.1.1	FSM detector tree	127
11.1.2	FSM hardware tree	128
11.1.3	CMS integration	129
11.2	Behaviour formalization into FSM hierarchy tables	130
11.2.1	Using tables	130
11.2.2	Using UML	133
11.3	FSM hierarchy tables and operation logic generation script	134
11.4	LHC status and handshake	136
11.5	New datapoint types	136
11.6	Critical actions	136
11.6.1	High rates versus Beam Loss Monitors	136
11.6.2	Cooling on and vacuum loss versus shutdown all electricity and sensors powering	137
11.6.3	Data taking scenarios with several RP not working	137
12	Integration with the motorization	139
12.1	Description of the collimators control architecture	139
12.1.1	Low Level control	139
12.1.2	Collimator Supervisory System (CSS) level	140
12.1.3	Central Collimation Application (CCA) level	140
12.2	Use cases	141
12.2.1	Adjust Roman Pot position for data taking	141
12.2.2	Adapt Roman Pots positions to failures and harmful context	141
12.2.3	Extract Roman Pot(s) in emergency	143
12.3	Interactions between the motorization control and the DCS	144
13	Information theory analysis	145
13.1	Introduction	145
13.2	Tool for the calculations	146
13.3	Orders of magnitude of the information exchanged among HW levels, nodes and buses	147
13.4	Data generation for the DCS	148
13.5	Archiving the data	149
13.6	Memory for execution	150
13.7	Analysis of the response time	150
13.7.1	Model of the CAN bus	150
13.7.2	Values for the RP ELMB box for Temperature and Vacuum	151
13.7.3	Model of the OPC	152
13.7.4	Model of the FSM	152
13.7.5	Global chain	152
13.7.6	Experimental results	153
14	Verificaton of the DCS	155
14.1	Commisioning of the RADMON-DAC	155
14.2	Replicability of the system	157
14.3	Response of the critical actions	158
15	User interface	159

List of contents

15.1	ALICE interface	159
15.2	JCOP integration	159
15.3	TOTEM specific panels	160
15.4	CMS 3D view	162
16	Conclusions	165
16.1	New results	165
16.2	Future researches	166
16.3	Development recommendations	167
IV	Appendices	169
A	Locations	171
A.1	The laboratory in building 555	171
A.2	Test area H8	172
A.3	USC55: counting room	172
A.4	UXC55: experimental cavern	172
A.5	Sectors 45 and 56	177
A.6	Alcoves UC53 and UC57	177
B	Hardware overview diagrams	179
B.1	Roman Pots Motorization	180
B.2	Roman Pots Silicon Detectors	181
B.3	T1	182
B.4	T2	183
B.5	ELMB global layout	184
B.6	Counting room	185
B.7	Legenda	186
C	Hardware components specifications	187
C.1	High voltage Caen power supplies	187
C.1.1	Functional description of SY1527	187
C.1.2	Functional description of A1520P	188
C.2	Low voltage Wiener power supplies	188
C.2.1	Primary Rectifier	188
C.2.2	Maraton Power Box	189
C.2.3	Maraton Remote Controller Module	189
C.3	Controller Area Network	190
C.3.1	Relation with OSI levels	190
C.3.2	CANopen	191
C.3.3	Sysworxx CAN to USB adapter	192
C.4	ELMB (Embedded Local Monitoring Boards)	192
C.4.1	Origins and objectives	192
C.4.2	Software	193
C.4.3	Hardware	193
C.4.4	Motherboard	195
C.4.5	Adapters	196
C.5	PXI (PCI eXtensions for Instrumentation)	196

D	Front end electronics	197
D.1	VFAT	197
D.2	DCU	199
E	External systems	201
E.1	Detector safety system	201
E.1.1	Types of alarms	201
E.2	Relationship with other systems	201
E.2.1	Architecture	203
E.3	CRoss-platform DAQ Framework (XDAQ)	203
E.3.1	Introduction	203
E.3.2	Executive framework	204
E.3.3	Application interfaces and state machines	204
E.3.4	Protocols and data formats	205
E.4	Run Control and Monitoring System	205
E.4.1	Overview	205
E.5	Architecture	206
E.5.1	RCMS and DAQ operation	207
F	Additional resources	209
F.1	bigbrother.cern.ch	209
F.2	SubVersion	209
F.3	iDoc	210
F.4	TWiki	211
F.5	HyperNews	212
F.6	JIRA	212
F.7	Photo Gallery	213
F.8	eLog	213
F.9	CMS remote access	213
G	Radiation monitoring library	215
V	References	I
	Publications	III
	Bibliography	VII
	Glossary	XIV

(This page is intentionally left blank)

Part

Environment and Precedents

(This page is intentionally left blank)

Chapter 1

CERN

1.1 An historical introduction

1.1.1 Countries

CERN is run by 20 European Member States [CER08a], but many non-European countries are also involved in different ways. The current Member States are: Austria, Belgium, Bulgaria, the Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Italy, the Netherlands, Norway, Poland, Portugal, the Slovak Republic, Spain, Sweden, Switzerland and the United Kingdom.

Scientists and their funding agencies from both Member and non-Member States are responsible for the financing, construction and operation of the experiments on which they collaborate. CERN spends much of its budget on building new machines (such as the Large Hadron Collider), and it contributes only partially to the cost of the experiments.

1.1.2 Accelerators

When high energy particles from an accelerator slam into a stationary target, most of the valuable projectile energy is taken up by the target recoil, and only a small fraction actually feeds the collision.

In the 1950s, CERN worked on a new scheme [CER08b]. If two particle beams could be fired at each other, no recoil energy would be wasted, making for much more violent collisions. The idea was to use the new proton synchrotron (PS) to feed two interconnected rings (ISR), where two intense proton beams could be built up and then made to collide.

The ISR, commissioned in 1971, produced the world's first proton-proton collisions. The collider, regarded as a masterpiece, gained CERN knowledge and expertise for its subsequent colliding beam projects.

In 1979 CERN profited of its ISR investment by deciding to convert its new Super Proton Synchrotron (SPS) into the world's first proton-antiproton collider. The first proton-antiproton collisions were



Figure 1.1: CERN photo (site of Meyrin)

achieved just two years after the project was approved, and in 1983 the W and Z particles were discovered (they are the carriers of the weak interaction).

After that the Large Electron-Positron Collider (LEP) was planned, and ISR was switched off to release resources for its construction. LEP was a circular collider with a circumference of 27 kilometers built in the tunnel straddling the border of Switzerland and France. It was used from 1989 until the end of 2000, when it was shut down and then dismantled in order to make room in the tunnel for the construction of the Large Hadron Collider (LHC).

1.2 Current status: LHC era

Protons are accelerated and formed in beams in four increasingly large machines before being injected with an energy of 450 GeV into the LHC's 27 km ring. The beams will then be accelerated in the ring until their energy is increased by a factor of 15, to 7000 GeV. When that energy is reached, the beams will collide in the centres of the experiments.

The Large Hadron Collider (LHC) will ultimately collide beams of protons at an energy of 14 TeV. Beams of lead nuclei will be also accelerated, smashing together with a collision energy of 1150 TeV.

The LHC is needed because the current understanding of the Universe is incomplete. The theory most widely used, the Standard Model, leaves many unsolved questions. Among them, the reason why elementary particles have mass, and why are their masses different is one of them. The answer may lie within the Standard Model, in an idea called the Higgs mechanism. According to this, the whole of space is filled with a 'Higgs field', and by interacting with this field, particles acquire their masses. Particles which interact strongly with the Higgs field are heavy, whilst those which interact weakly are light. The Higgs field has at least one new particle associated with it, the Higgs boson. If such particle exists, the LHC will be able to make it detectable.

The LHC can also help to solve the riddle of antimatter. It was once thought that antimatter was a perfect 'reflection' of matter; that if the matter is replaced with antimatter it is not possible to observe any difference. Now it is known that this reflection is imperfect, and this could have led to the matter-antimatter imbalance.

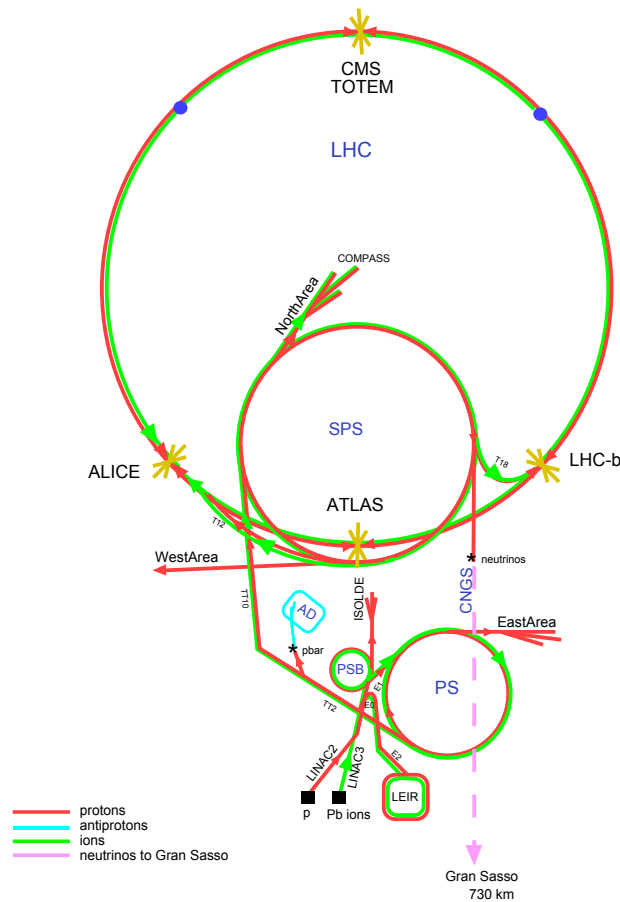


Figure 1.2: Operational CERN accelerators (not to scale); based on [LM01]

1.3 Future: An upgrade for the LHC

There are reasons for considering an upgrade of the LHC after about 6 years of operation [RZ06]. There are two possible evolutions of peak luminosity as a function of the year. Both scenarios assume an LHC start up in 2010, and that 10% of the design luminosity is reached in 2011, and 100% in 2013, but after that:

- The luminosity is raised to the nominal value by 2014 and then stays constant.
- It continues to increase linearly, reaching the ultimate value by 2019.

In one case, the luminosity is taken to be constant from then on, in the other it continues to increase linearly until the so-called ultimate luminosity (2.3 times the nominal) would be reached by 2019. The radiation damage limit of the LHC low- β quadrupoles is estimated at an integrated luminosity of 600–700 fb/1. As Figure 1.3 shows, this value would be exceeded in 2017 or 2019 depending on the scenario. The additional run-time required to halve the statistical error rises more steeply. It would exceed 7 years by 2014 or 2016, respectively.

Since the life expectancy of the interaction-region (IR) magnets is estimated to be less than 10 years due to the high radiation doses and since the time needed to halve the statistical error will exceed 5 years by 2014–2015, it is reasonable to plan a machine luminosity upgrade based on new low- β IR magnets before 2016.

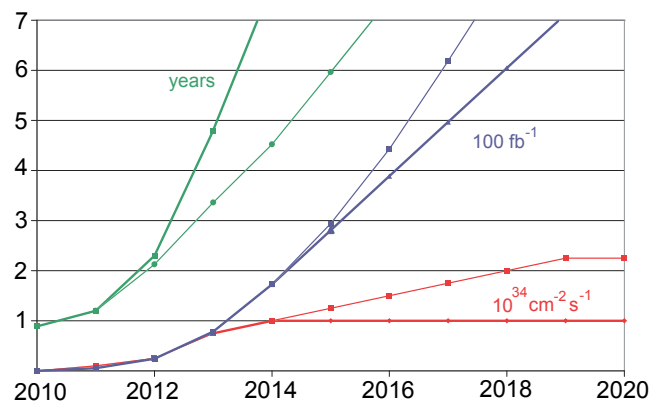


Figure 1.3: Time to halve the statistical error (green curves), integrated luminosity (blue curves), and peak luminosity (red curves); adapted from [Str03]

Chapter 2

The Large Hadron Collider

2.1 The LHC machine

2.1.1 Parameters

In an accelerator, the most important parameters are the beam energy and the number of interesting collisions. More specifically, in a collider such as the LHC the probability for a particular process varies with what is known as the luminosity. This quantity depends on the number of particles in each bunch, the frequency of complete turns around the ring, the number of bunches and the beam cross-section. A good introduction to all the parameters is given in [CER08c], while [LHC00] gives the full technical description. Table 2.1, lists the important parameters for the LHC:

Quantity	Number
Circumference	26659 m
Dipole operating temperature	1.9 K
Number of magnets	9593
Number of main dipoles	1232
Number of main quadrupoles	392
Number of RF cavities	8 per beam
Nominal energy, protons	7 TeV
Nominal energy, ions	2.76 TeV/u
Peak magnetic dipole field	8.33 T
Min. distance between bunches	~ 7 m
Design luminosity	$10^{34} \text{ cm}^{-2} \text{ s}^{-1}$
No. of bunches per proton beam	2808
No. of protons per bunch (at start)	1.1×10^{11}
Number of turns per second	11245
Number of collisions per second	600 million

Table 2.1: Main parameters of the LHC, from [CER08c]

2.1.2 Bunches

The protons of the LHC circulate around the ring in well-defined bunches. The bunch structure of a modern accelerator is a direct consequence of the radio frequency (RF) acceleration scheme. Protons can only be accelerated when the RF field has the correct orientation when particles pass through an accelerating cavity, which happens at well specified moments during an RF cycle. In the LHC, under nominal operating conditions, each proton beam has 2808 bunches, with each bunch containing about 10^{11} protons.

The bunch size is not constant around the ring. Each bunch, as it circulates around the LHC, gets squeezed and expanded; see Section 6.1, *Beam optics*. Bunches of particles measure a few centimeters long and a millimeter wide when they are far from a collision point. However, as they approach the collision points, they are squeezed to about $16\text{ }\mu\text{m}$ to allow for a greater chance of proton-proton collisions. Increasing the number of bunches is one of the ways to increase luminosity in a machine. The LHC uses a bunch spacing of 25 ns (or about 7 m), while LEP, operated with as few as 4 bunches).

The bunch spacing of 25 ns corresponds to a frequency of 40 MHz, which implies that bunches should pass each of the collision points in the LHC 40 million times a second.

2.1.3 Sectors and octants

The LHC is not a perfect circle. It is made of eight arcs and eight ‘insertions’. An insertion consists of a long straight section plus two (one at each end) transition regions (‘dispersion suppressors’). The exact layout of the straight section depends on the specific use of the insertion: physics (beam collisions within an experiment), injection, beam dumping, beam cleaning.

Figure 2.3 shows all the LHC sectors and octants. A sector is defined as the part of the machine between two insertion points. The eight sectors are the working units of the LHC: the magnet installation happens sector by sector, the hardware is commissioned sector by sector and all the dipoles of a sector are connected in series and are in the same continuous cryostat. Powering of each sector is essentially independent. An octant starts from the middle of an arc and ends in the middle of the following arc and thus spans a full insertion.

2.1.4 Accelerator components

In the accelerators, particles circulate in a vacuum tube and are manipulated using electromagnetic devices: dipole magnets keep the particles in their nearly circular orbits, quadrupole magnets focus the beam, and accelerating cavities are electromagnetic resonators that accelerate particles and then keep them at a constant energy by compensating for energy losses.

Vacuum

The LHC has the particularity of having not one, but three vacuum systems:

- insulation vacuum for cryomagnets
- insulation vacuum for the helium distribution line
- beam vacuum

The beam vacuum pressure will be 10^{-13} atm (ultrahigh vacuum), to avoid collisions with gas molecules.

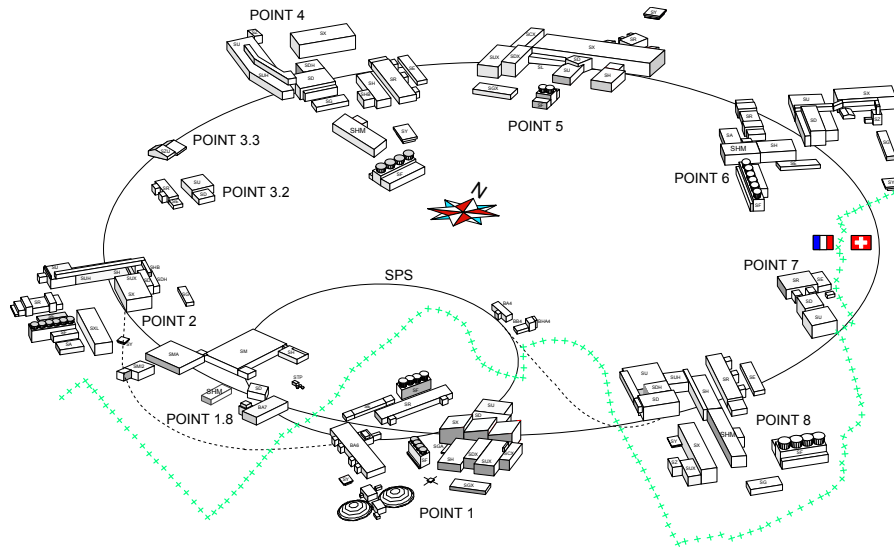


Figure 2.1: LHC Surface (not to scale); adapted from [CER06]

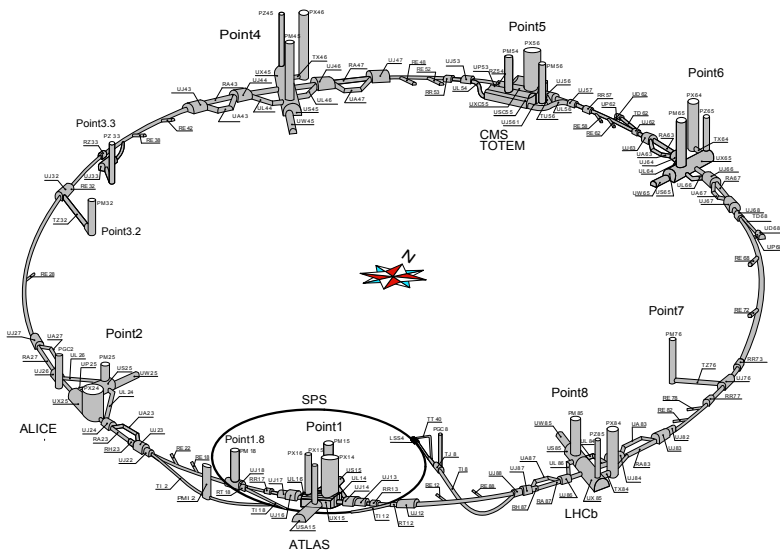


Figure 2.2: LHC Underground (not to scale); adapted from [CER06]

Magnets

There is a large variety of magnets in the LHC, including dipoles, quadrupoles, sextupoles, octupoles, decapoles, etc. giving a total of about 9600 magnets. Each type of magnet contributes to optimizing a particle's trajectory. Most of the correction magnets are embedded in the cold mass of the main dipoles and quadrupoles. The LHC magnets have either a twin aperture (for example, the main dipoles), or a single aperture (for example, some of the insertion quadrupoles). Insertion quadrupoles are special magnets used to focus the beam down to the smallest possible size at the collision points, thereby maximizing the chance of two protons smashing head-on into each other.

Cavities

The main role of the LHC cavities is to keep the 2808 proton bunches tightly bunched to ensure high luminosity at the collision points and hence, maximize the number of collisions. They also deliver

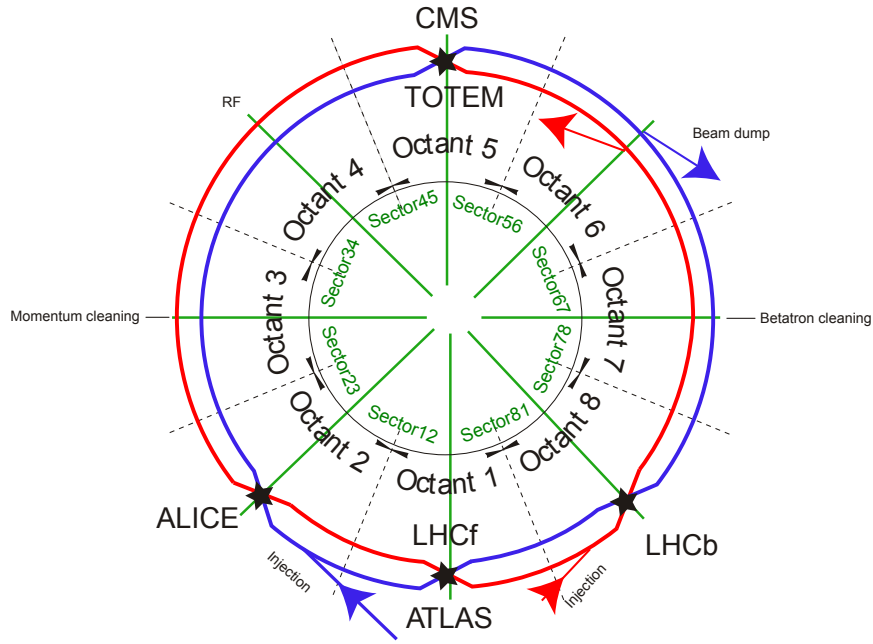


Figure 2.3: Sectors and octants

radiofrequency (RF) power to the beam during acceleration to the top energy. Superconducting cavities with small energy losses and large stored energy are the best solution. The LHC will use eight cavities per beam, each delivering 2 MV (an accelerating field of 5 MV/m) at 400 MHz. The cavities will operate at 4.5 K (the LHC magnets will use superfluid helium at 1.9 K).

Dipoles

The dipoles of the LHC represented the most important technological challenge for the LHC design. In a proton accelerator like the LHC, the maximum energy that can be achieved is directly proportional to the strength of the dipole field, given a specific acceleration circumference. At the LHC the dipole magnets are superconducting electromagnets and able to provide the very high field of 8.3 T over their length. No practical solution could have been designed using ‘warm’ magnets instead of superconducting ones.

2.1.5 BLM (Beam Loss Monitors)

Comparing transverse energy densities, LHC advances the state of the art by even three orders of magnitude, from 1 MJ/mm² to 1 GJ/mm², which makes the LHC beams highly destructive. At the same time the superconducting magnets in the LHC would quench at 7 TeV if small amounts of energy (on the level of 30 mJ/cm³, induced by a local transient loss of 4×10^7 protons) are deposited into the superconducting magnet coils [JLOT96].

The detection of the lost beam protons allows protection of the equipment against quenches and damage by generating a beam dump trigger when the losses exceed thresholds. In addition to the quench prevention and damage protection, the loss detection allows the observation of local aperture restrictions, orbit distortion, beam oscillations and particle diffusion, etc. Since a repair of a superconducting magnet would cause a down time of several weeks, the protection against damage has highest priority.

2.1.6 BPM (Beam Position Monitors)

The LHC orbit and trajectory measurement system has been developed to fulfil the functional specifications described in [Kou02; LHC00]. The system consists of 516 monitors per LHC ring, all measuring in both horizontal and vertical planes. The acquisition electronics is capable of 40 MHz bunch-by-bunch measurements and will provide closed orbit feedback at 1 MHz.

2.1.7 Collimators

The handling of the high intensity LHC beams and the associated high loss rates of protons requires a powerful collimation system with the following functionality [LHC00]:

- Efficient cleaning of the beam halo during the full LHC beam cycle, such that beam-induced quenches of the super-conducting magnets are avoided during routine operation.
- Minimisation of halo-induced backgrounds in the particle physics experiments.
- Passive protection of the machine aperture against abnormal beam loss. Beam loss monitors at the collimators detect any unusually high loss rates and generate a beam abort trigger.
- Scraping of beam tails and diagnostics of halo population.
- Abort gap cleaning in order to avoid spurious quenches after normal beam dumps.

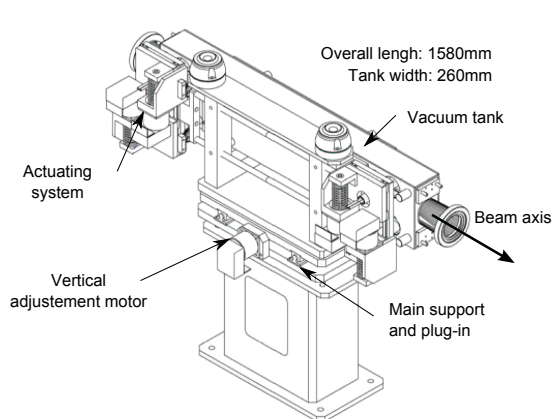


Figure 2.4: General layout and dimensions of the LHC secondary collimator (vertical configuration)

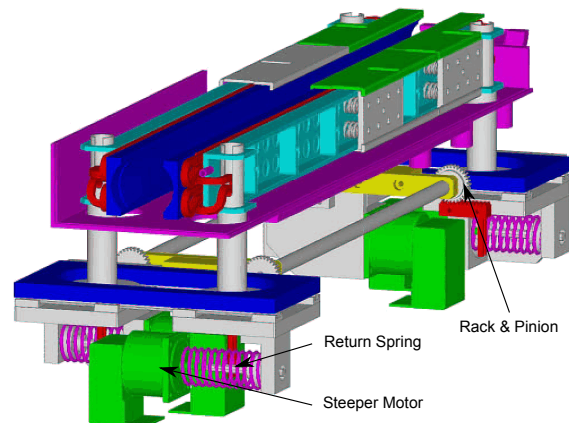


Figure 2.5: Motorisation and actuation system

Beam impact at the collimators is divided into normal and abnormal processes [AAB⁺03; AGVW02; ABB⁺02a]:

- Normal proton losses can occur due to beam dynamics (particle diffusion [ASZZ02], scattering processes, instabilities) or operational variations (orbit, tune, chromaticity changes during ramp, squeeze, collision). These losses must be minimised but cannot be avoided completely.
- Abnormal losses result from failure or irregular behaviour of accelerator components.

The design of the collimation system relies on the specified normal and abnormal operational conditions and if these conditions are met it is expected that the collimation system will work correctly and that components will not be damaged. It is assumed that the beams are dumped when the proton loss rates exceed the specified maximum rates.

2.2 The LHC experiments

2.2.1 ALICE

ALICE is an experiment specialized in analysing lead-ion collisions. It will study the properties of quark-gluon plasma, a state of matter where quarks and gluons, under conditions of very high temperatures and densities, are no longer confined inside hadrons. Such a state of matter probably existed just after the Big Bang, before particles such as protons and neutrons were formed, [CER08c; ALI08].

Parameter	Value
Size	26 m long, 16 m high, 16 m wide
Weight	10000 tonnes
Design	central barrel plus single arm forward muon spectrometer
Material cost	115 MCHF
Collaboration	~ 1500 members from 104 institutes in 31 countries

Figure 2.6: ALICE parameters

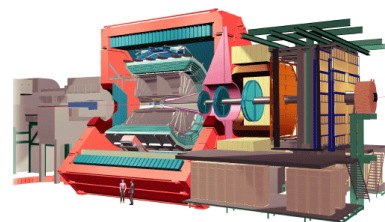


Figure 2.7: ALICE diagram

2.2.2 ATLAS

ATLAS is a general purpose experiment designed to cover the widest possible range of physics at the LHC, from the search for the Higgs boson to supersymmetry and extra dimensions. The main feature of the ATLAS experiment is its enormous doughnut-shaped magnet system. This consists of eight 25 m long superconducting magnet coils, arranged to form a cylinder around the beam pipe through the centre of the experiment, [CER08c; ATL08].

Parameter	Value
Size	46 m long, 25 m high and 25 m wide
Weight	7000 tonnes
Design	barrel plus end caps
Material cost	540 MCHF
Collaboration	~ 1900 members from 164 institutes in 35 countries

Figure 2.8: ATLAS parameters

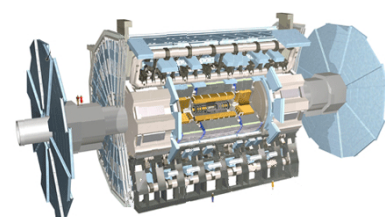


Figure 2.9: ATLAS diagram

2.2.3 CMS

CMS is a general purpose experiment with the same physics goals as ATLAS, but different technical solutions and design. It is built around a huge superconducting solenoid. This takes the form of a cylindrical coil of superconducting cable that will generate a magnetic field of 4 T, about 100 000 times that of the Earth, [CER08c; CMS08].

Parameter	Value
Size	21 m long, 15 m high, 15 m wide
Weight	12500 tonnes
Design	barrel plus end caps
Material cost	500 MCHF
Collaboration	~ 2000 members from 181 institutes in 38 countries

Figure 2.10: CMS parameters

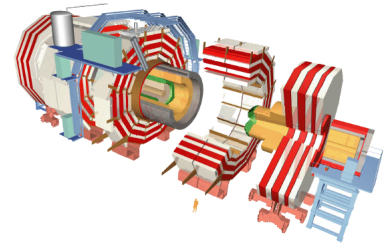


Figure 2.11: CMS diagram

2.2.4 LHCb

LHCb specializes in the study of the slight asymmetry between matter and antimatter present in interactions of B-particles (particles containing the b quark). Instead of surrounding the entire collision point with an enclosed detector, the LHCb experiment uses a series of detectors to detect mainly forward particles. The first detector is built around the collision point, the next ones stand one behind the other, over a length of 20 m, [CER08c; LHC08].

Parameter	Value
Size	21 m long, 10 m high, 13 m wide
Weight	5600 tonnes
Design	forward spectrometer with planar detectors
Material cost	75 MCHF
Collaboration	~ 650 members from 47 institutes in 14 countries

Figure 2.12: LHCb parameters

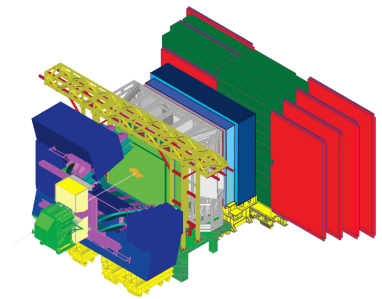


Figure 2.13: LHCb diagram

2.2.5 LHCf

LHCf is a small experiment that will measure particles produced very close to the direction of the beams in the proton-proton collisions at the LHC. The motivation is to test models used to estimate the primary energy of the ultra high-energy cosmic rays. It will have detectors 140 m far from the ATLAS collision point, [CER08c].

Parameter	Value
Size	two detectors of 30 cm long, 10 cm high and 10 cm wide
Weight	40 kg each
Design	longitudinal detectors
Collaboration	~ 21 members from 10 institutes in 6 countries

Table 2.2: LHCf parameters

2.2.6 TOTEM

TOTEM will measure the effective size or ‘cross-section’ of the proton at LHC. To do this TOTEM must be able to detect particles produced very close to the LHC beams. It will include detectors housed in specially designed vacuum chambers called ‘Roman pots’, which are connected to the beam pipes in the LHC. Eight Roman pots will be placed in pairs at four locations near the collision point of the CMS experiment, [CER08c; TOT08].

Parameter	Value
Size	440 m long, 5 m high, 5 m wide
Weight	20 tonnes
Design	longitudinal detectors
Material cost	6.5 MCHF
Collaboration	~ 70 members from 10 institutes in 7 countries

Table 2.3: TOTEM parameters

Chapter 3

The TOTEM experiment

3.1 Introduction

TOTEM was proposed in 1997 [KBB⁺97]. Having received favourable consideration from the LHCC and the Research Board, the Collaboration prepared a Technical Proposal [KOP⁺99] in 1999 in which they identified CMS as the optimal host experiment for TOTEM. The full Technical Proposal was released in 2004 [BCR⁺04], and the physics program was detailed in 2006 [AAA⁺07]. A detailed report was published during 2008 [AAA⁺08].

TOTEM will pursue a physics program spanning a wide range from total cross-section and elastic scattering measurements to the study of diffractive and forward phenomena. The TOTEM program will lead to a better understanding of the fundamental aspects of strong interactions. For the first time at hadron colliders, the very forward rapidity range, containing 90% of the energy flow and explored in high-energy cosmic ray experiments, is covered, allowing the search for unusual phenomena hinted by cosmic ray experiments.

A scheme to tag events from Double-Pomeron-Exchange by diffractive protons on both sides, transforms the LHC into an almost clean ‘gluon’ collider, where the centre-of-mass energy is determined by the momentum losses of the forward protons, thus offering an interesting way to search for new particles. In a later stage, the combination of CMS and TOTEM provides an unprecedented almost complete rapidity coverage, allowing a variety of new studies, including hard diffraction.

Specifically TOTEM measures:

- The total pp (proton-proton) cross-section with an absolute error of 1 mb by using the luminosity independent method. This requires the simultaneous measurement of the elastic pp scattering down to the four-momentum transfer of $-t \approx 10^{-3} \text{ GeV}^2$ and of the inelastic pp interaction rate with an adequate acceptance in the forward region.
- Elastic proton scattering over a wide range in momentum transfer up to $-t \approx 10 \text{ GeV}^2$.
- Diffractive dissociation, including single, double and central diffraction topologies using the forward inelastic detectors in combination with one of the big LHC experiments.

Two tracking telescopes, T1 and T2, installed on each side of the IP (Figure 3.2) provides measurements of the inelastic pp interaction in the forward region. The T1 telescope will be placed in the CMS

The TOTEM experiment

endcaps, while T2 will be in the shielding behind the CMS Hadronic Forward (HF) calorimeter. T1 and T2 add charged particle tracking and trigger capabilities to the CMS experiment over a rapidity interval $3 \leq |\eta| \leq 6.8$.



Figure 3.1: Insertion of T1 and T2 into CMS

The precise determination of the total cross-section requires that TOTEM must also measure $d\sigma_{el}/dt$ down to $-t \approx 10^{-3} \text{ GeV}^2$. This is accomplished with two sets of silicon detectors in Roman Pots located symmetrically on each side of the IP at 147 m and 220 m (Figure 3.2).

It is important to note that the 147 m Roman Pots are located before the D2 magnet, while the 220 m tracking station is well behind it. This geometry naturally implements a magnetic spectrometer in the standard insertion, permitting TOTEM to measure particle momenta, with an accuracy of a few parts per thousand. This will allow the accurate determination of the momentum loss of quasi-elastically scattered protons in diffractive processes.

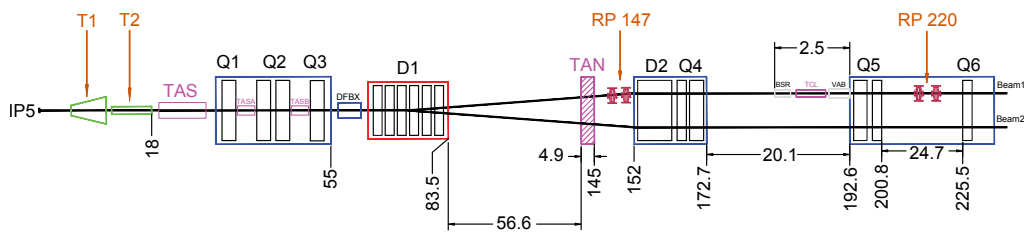


Figure 3.2: The LHC beam line, T1, T2 and the Roman pots at 147 m and 220 m

The details of the machine optics are crucial to this measurement. A optics with a β^* of 1540 m has been developed, with parallel-to-point focusing conditions in the vertical plane at the 147 m stations and in the horizontal plane at the 220 m stations.

TOTEM will only need a few days of running with special running conditions ($\beta^* \approx 1540 \text{ m}$, 43 bunches, and zero crossing angles) at a luminosity of $10^{28} \text{ cm}^{-2}\text{s}^{-1}$ in order to measure the total cross-section. Increasing the proton density per bunch and the number of bunches and lowering the β^* to a few hundred meters will allow TOTEM to run at luminosities of a few times $10^{30} \text{ cm}^{-2}\text{s}^{-1}$ while still detecting most of the diffractive protons.

3.2 The physics programme

3.2.1 Total cross-section

The COMPETE collaboration [CEG⁺02] has made an overall fit of the energy dependence of the total cross-section and the ratio of the real to the imaginary part of the elastic scattering (ρ parameter), taking into account all available data. Their prediction for the energy dependence of the total pp cross-section is shown in Figure 3.3 from [CEG⁺02].

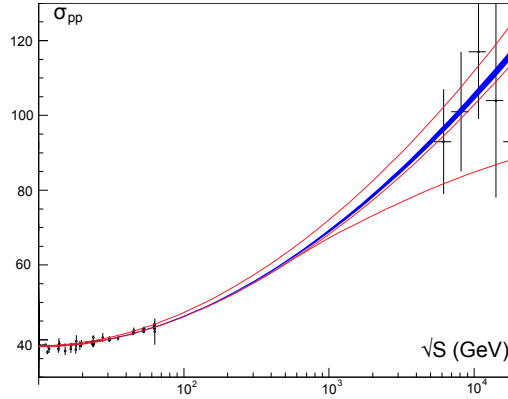


Figure 3.3: Predictions for total pp cross-sections , including ISR and cosmic ray data

The black error band shows the statistical errors to the best fit, the closest curves near it give the sum of statistical and systematic errors to the best fit due to the ambiguity in the TEVATRON data and the highest and lowest curves show the total error bands from all models considered.

For the LHC energy they obtain the following values for σ_{tot} and ρ :

$$\sigma_{\text{tot}} = 111.5 \pm 1.2^{+4.1}_{-2.1} \text{ mb} \quad (3.1)$$

$$\rho = 0.1361 \pm 0.0015^{+0.0058}_{-0.0025} \quad (3.2)$$

The first error is the statistical error to the best fit and the second one arises from the ambiguity in the TEVATRON data. In order to have an input on the selection of different models the experimental error of σ_{tot} should be in the order of 1 mb. The total cross-section will be determined in a luminosity independent way using the optical theorem (see equation 3.3).

$$\sigma_{\text{tot}} = \frac{16\pi}{(1 + \rho^2)} \frac{(dN_{\text{el}}/dt)_{t=0}}{N_{\text{el}} + N_{\text{inel}}} \quad (3.3)$$

Thus the optical point at $t = 0$ has to be extrapolated from the measurement of the elastic scattering at low momentum transfers.

3.2.2 Elastic scattering

The field model, underlying the phenomenological analysis in [ILP03], describes the nucleon as having an outer cloud of quark-antiquark condensed ground states, an inner core of topological baryonic charge of radius 0.44 fm and a still smaller valence quark-bag of radius < 0.1 fm. These different shells of the nucleon participate in different t -regions of the elastic scattering.

To discriminate between different models it is thus important to precisely measure the elastic scattering over the whole kinematical region as the model predictions may be rather different for higher values of t .

The t -distribution of the elastic scattering, assuming the BSW model [BSW02], is given in Figure 3.4. The number of events at the right scale corresponds to an integrated luminosity of 10^{33} cm^{-2} and 10^{37} cm^{-2} . The dotted line indicates the highest observable t -value due to aperture limitation in the high- β^* optics.

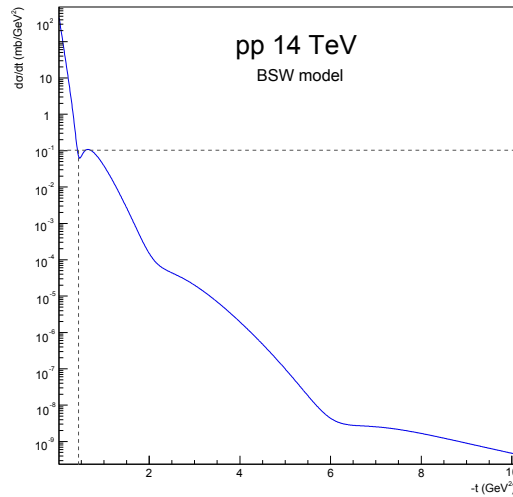


Figure 3.4: Elastic scattering cross-section using the model from BSW

It extends over 11 orders-of-magnitude and has therefore to be measured with different optics settings. The exponential fall at low t is followed by a diffractive structure at $\sim 1 \text{ GeV}^2$ and continues to large t -values where perturbative calculations suggest a power-law behaviour (t^{-8}). The number of events at the right side of the plot refers to a few days running at two conditions. The maximal detectable t -value due to aperture limitations in the LHC is $0.5 (10 \text{ GeV}^2)$.

3.2.3 Diffraction and the pomeron

Two types of processes, the Single-Pomeron-Exchange and the Double-Pomeron-Exchange are of particular interest.

$$p + p \rightarrow p + X \quad (3.4)$$

$$p + p \rightarrow p + X + p \quad (3.5)$$

It has been noted [IS85] that hard Pomeron-proton collisions create final states such as high- p_t jets, $t\bar{t}$ or $b\bar{b}$ in the X system (reaction 3.4). These events can be quite spectacular, consisting of a co-planar dijet accompanied by the two unfragmented protons and nothing else. In some cases the Pomeron will give most of its momentum to one gluon (the hard structure, observed by UA8) which then interacts with a parton of the other proton creating high mass states, like e.g. the Higgs, in the central rapidity region.

The study of the systems X in reactions (3.4) and (3.5) will answer many fundamental questions about the Pomeron structure and the Pomeron effective flux factor in the proton. In particular, detailed studies of the kinematics show that the longitudinal momentum of the mass X reflects well the

Pomeron structure function. An estimate of the cross-section of reaction (3.4) at the LHC with jets $E_t > 50$ GeV gives 10 nb with a factor 2-3 uncertainty. This implies that one can expect to acquire about 10^3 di-jet events for an integrated luminosity of 10^{35} cm^{-2} , corresponding to a few runs at large β^* with luminosities between $10^{29} - 10^{30} \text{ cm}^{-2} \text{ s}^{-1}$.

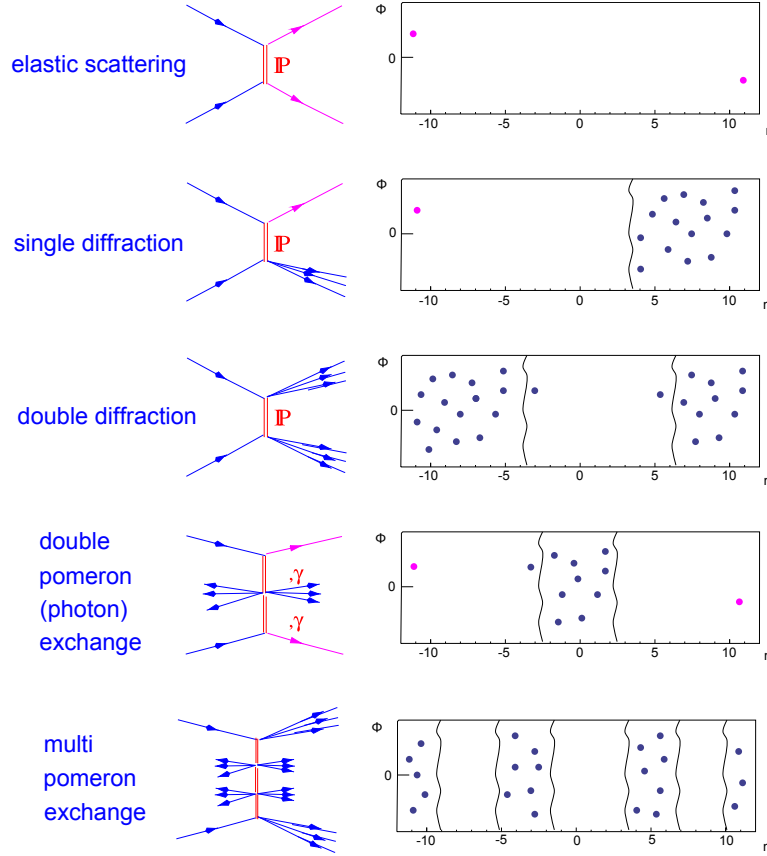


Figure 3.5: Visualization of diffractive processes in the rapidity-azimuth-plot

3.3 RpMe (Roman Pot Mechanics)

3.3.1 System strategy and overview

The detection of very forward protons in movable beam insertions –called Roman Pots (RP) –is an experimental technique introduced at the ISR by a CERN/Rome group in the early 1970s, this is the reason why they are called ‘roman’. It has been successfully employed in other colliders like the SPS, TEVATRON, RHIC and HERA. Detectors are placed inside a secondary vacuum vessel, called a pot, and moved into the primary vacuum of the machine through vacuum bellows. In this way, the detectors are physically separated from the primary vacuum which is thus preserved against an uncontrolled out-gassing of the detector's materials.

On each side of IP5, two stations of Roman Pots will be mounted on the beam pipe of the outgoing beam [AAA⁺08]. The centre of the first station (‘RP147’) is placed at 149.6 m from IP5, and the second (‘RP220’) at 217.3 m. Each RP station is composed of two units separated by a distance limited by integration constraints with the other beam elements (Figure 3.6). In summary, a total of 8 identical Roman Pot units or 24 individual pots are installed in the LHC.

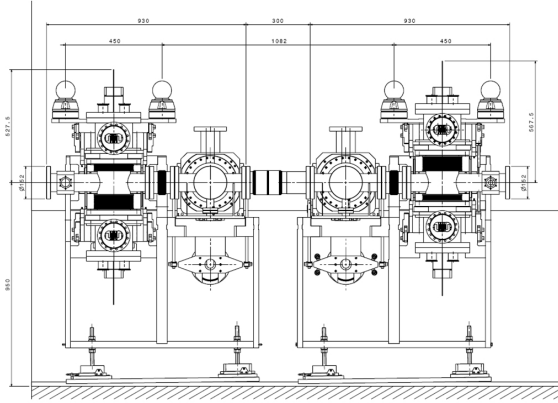


Figure 3.6: Design drawing of the station RP147; an assembly of two RP units

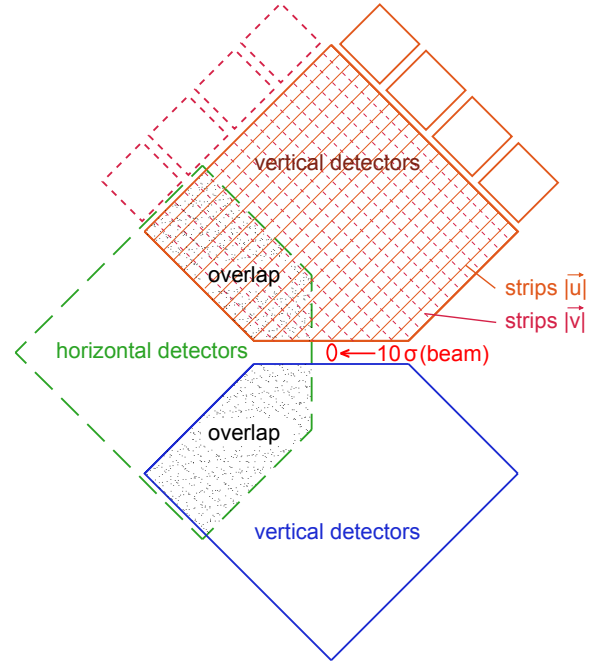


Figure 3.7: The overlap between the horizontal and vertical detectors

The single horizontal pot in each unit, placed on the outer side of the LHC ring, serves two purposes. Firstly, it completes the acceptance for diffractively scattered protons. On the inner side of the LHC ring no detector is needed since only background protons arrive in that position. Secondly, the detectors in the horizontal pots overlap with the ones in the vertical pots, which correlates their positions via common particle tracks (see Figure 3.7). This feature is used for the relative alignment of the three pots in a unit. For the absolute alignment with respect to the beam, a Beam Position Monitor (BPM), based on a button feed-through technology, is integrated in the vacuum chamber of the vertical RP.

3.3.2 The movements

Each pot is independently moved by micro-stepping motors with angular steps of $(0.90 \pm 0.03)^\circ$ per step, corresponding to 400 steps per turn. The transformation from the motor's rotational movement to the pot's translation movement is done by roller screws which provide high precision and zero backlashes.

A mechanical compensation system (Figure 3.8) balances the atmospheric pressure load on the pot. The system relieves the stress on the driving mechanism, improving the movement accuracy and the safety of the operations. It is based on a separate vacuum system connected to the primary vacuum of the machine through a by-pass. The atmospheric pressure load on the pot-bellow system is ~ 3000 N. With such a compensation system the stepper motor works only against the weight of the pot assembly (~ 100 N), leaving the possibility to achieve a better accuracy of the motor drive mechanism. With bellows on the compensation system larger than the pot bellows, a constant pulling load on the pots is guaranteed, and since the roller screws are a reversible mechanism, this feature is exploited to provide auto-retraction of the pots in case of a motor power cut.

The nominal mechanical pot-positioning resolution of the driving mechanism is $5 \mu\text{m}$, but the final precision depends on the assembly of the motors and the roller screws. The stepper motors are

equipped with angular resolvers which give the absolute position of each pot with respect to the nominal beam axis. Additional displacement inductive sensors (LVDT) provide the absolute position of each pot.

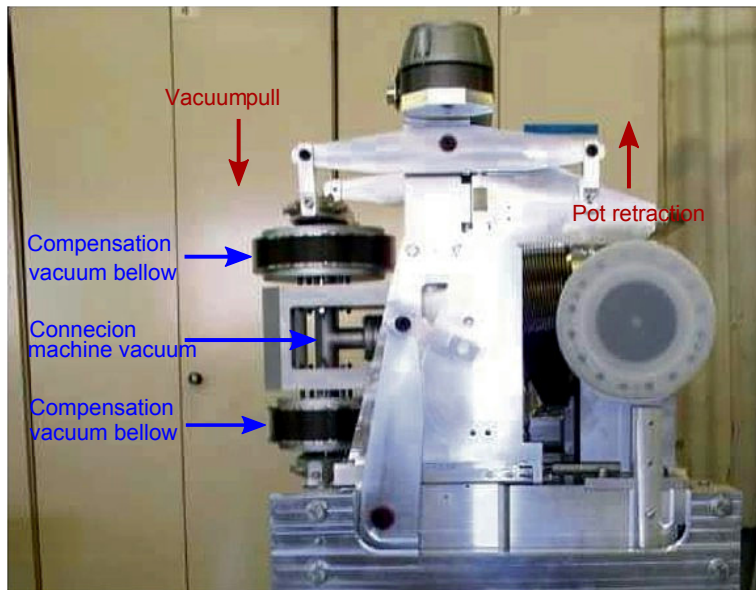


Figure 3.8: The vacuum compensation system

3.4 RpSi (Roman Pot Silicon detector)

3.4.1 System strategy and overview

Each pot is equipped with a stack of 10 planes of ‘edgeless’ silicon strip detectors (Section 3.4.2, *Current terminating structures*). Half of them will have their strips oriented at an angle of $+45^\circ$ with respect to the edge facing the beam, and the other half at an angle of -45° , measuring the coordinates $\vec{u}$ and $\vec{v}$ respectively. This configuration has the advantage that the hit profiles in the two projections are equivalent. The measurement of each track projection in five planes is advantageous for the reduction of uncorrelated background via programmable coincidences, requiring e.g. collinear hits in a majority of the planes.

The detector or sensor is usually called the ‘plane’ and this is the simplest part that can be controlled individually. In each plane, the strips are grouped in 4 sets, each one with 128 strips corresponding to the inputs of one single readout chip (VFAT2). In each set every 10 strips are numbered with digits, starting with 0 on the first strip of the set.

3.4.2 Current terminating structures

For segmented devices with this new so-called ‘Current Terminating Structure’ (CTS), the potential applied to bias the device has to be applied also across the cut edges via a guardring running along the die cut and surrounding the whole sample. This external guardring, also called ‘Current Terminating Ring’ (CTR) collects the current generated in the highly damaged region at the cut edge, avoiding its diffusion into the sensitive volume, and is separated from the biasing electrode. In this manner the

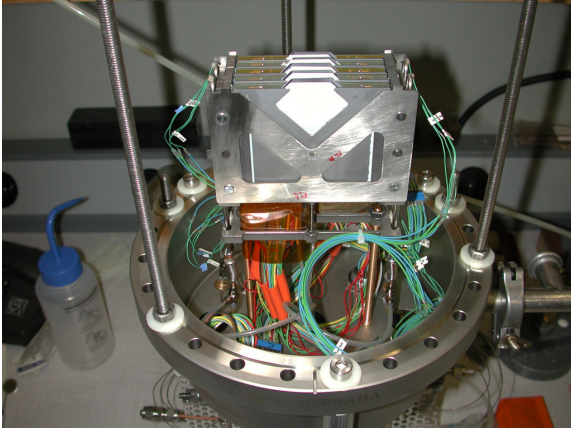


Figure 3.9: Planes insertion into the bottom Roman Pot

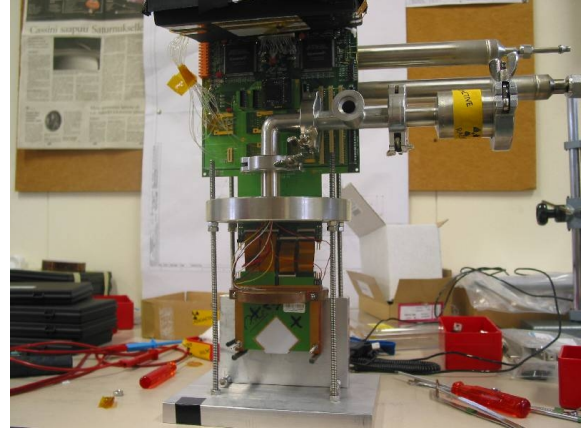


Figure 3.10: Motherboard and hybrid

sensitive volume can start at less than $50\text{ }\mu\text{m}$ from the cut edge. To prevent any further diffusion of this edge current into the sensitive volume, another implanted ring –the Clean-up Ring –can be placed between the CTR and the sensitive volume.

For devices with this type of CTS, the leakage current in the sensitive volume (I_{BE}) which contributes to noise is not affected by the edge current ($I_{CTR} + I_{CR}$). The leakage current and the edge current have been shown to be completely decoupled. Moreover, for such devices, the charge collection efficiency has been shown to rise steeply from the edge of the sensitive volume reaching full efficiency within a few tens of micrometers.



Figure 3.11: Planar Edgeless Detector with CTS

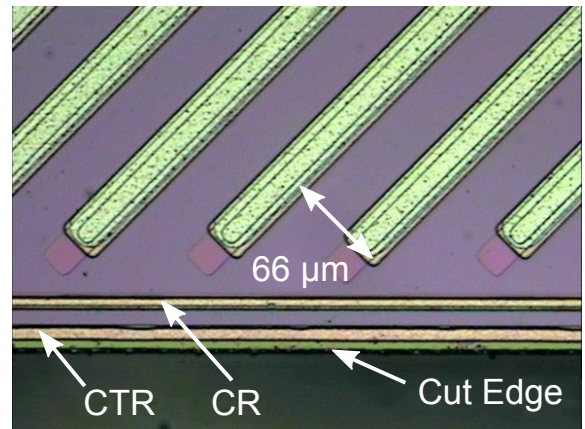


Figure 3.12: The magnification of a portion of the chip cut region shows the details of the CTS

3.4.3 On-detector electronics

The silicon detector hybrid (Figure 3.13) carries the detector with 512 strips wire-bonded to the input channels of 4 readout chips ‘VFAT’, and a Detector Control Unit (DCU) chip. Each VFAT provides tracking and trigger generation signals from 128 strips. The DCU chip is used to monitor detector

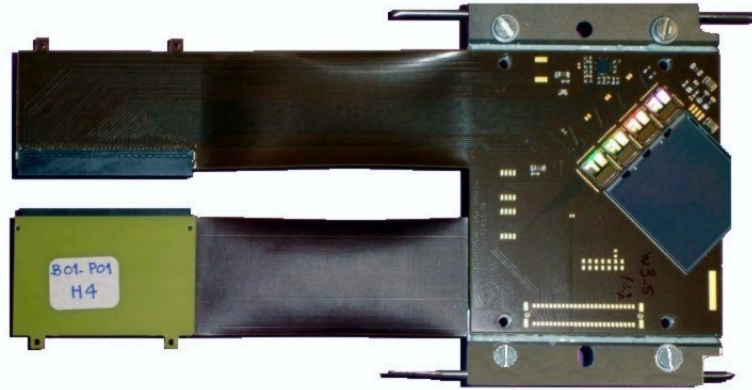


Figure 3.13: Two RP hybrids mounted back-to-back. Each hybrid carries the silicon detector and four VFAT readout chips.

leakage currents, power supply voltages and temperature. Via an 80 pin connector each VFAT will send the trigger outputs and the serialised tracking data from all strips to the motherboard, together with clock and trigger input signals, HV and LV power, and connections for a heater and a PT100 temperature sensor.

3.5 T1 (Telescope 1)

3.5.1 Detector geometry

CSCs of very large dimensions have been built for ATLAS [ATL97], CMS [CMS97] and LHCb [LHC01].

The design of the T1 consists of ten cathode strip chambers with trigger capabilities, equally spaced over 3 m along the beam. Each chamber provides a space point with a precision of the order of 0.5 mm. This permits the reconstruction of the primary collision vertex in the transverse plane within a few mm, enough to discriminate between beam –beam and beam –gas events [AAA⁺08].

The two *arms* of the T1 telescope, one on either side of the IP5, fit in the space between two conical surfaces, the beam pipe and the inner envelope of the flux return yoke of the CMS end-cap, at a distance between 7.5 m and 10.5 m from the interaction point. The vacuum chamber is in place and aligned when the installation of T1 takes place: for this reason each telescope arm is built in two vertically divided halves (*half arms*) as depicted in Figure 3.14.

The T1 telescopes will be the last piece to be inserted when closing and the first on to be removed when opening the CMS experiment.

A detector plane is composed of six CSC wire chambers covering roughly a region of 60° in ϕ and, as mentioned above, is split in two halves and mounted on different supports. Overlap is provided between adjacent detectors (also for the ones on different supports) to cover with continuous efficiency the approximately circular region of each telescope plane. In addition, the detector sextants in each plane are slightly rotated with respect to each other by angles varying from -6° to $+6^\circ$ in steps of 3° , the ‘reference’ orientation being that the layer number 5. This arrangement is useful for pattern recognition and helps to reduce the localised concentration of material in front of the CMS HF (Hadronic Forward) Calorimeter.

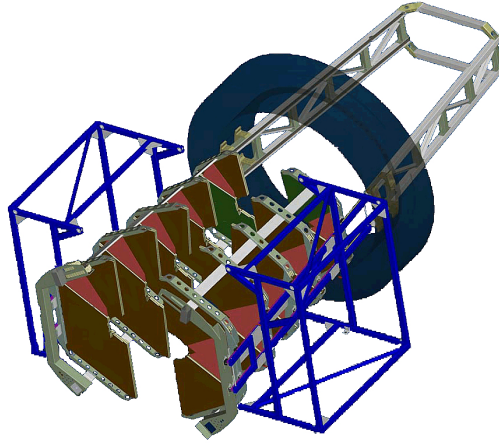


Figure 3.14: The two halves of one TOTEM T1 telescope arm before insertion in CMS

3.5.2 Description of the T1 CSC detectors

An exploded view of the different components making up a chamber assembly is shown in Figure 3.15.

Two stiff panels of trapezoidal shape determine the flat surfaces of the cathode planes. A thin continuous frame is inserted between the two panels to keep with a good precision the two cathode planes parallel with a gap of 10.0 mm.

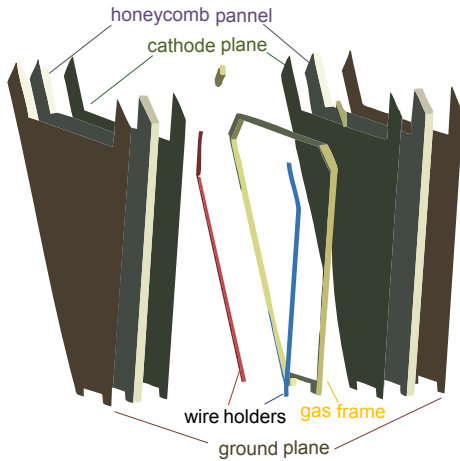


Figure 3.15: Exploded view of a TOTEM T1 CSC detector

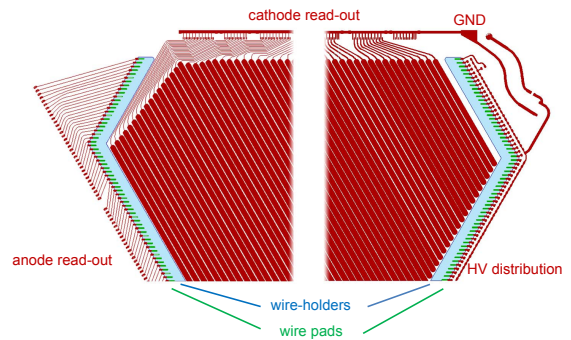


Figure 3.16: TOTEM T1 Cathode Strips and wire-holder printed circuit boards

The two panels are made with a 15 mm thick Nomex hexagonal honeycomb, enclosed between two 0.8 mm thick ‘skins’ of fiberglass/epoxy laminate. Both skins are covered by a 35 μm thick copper layer. Cathode strips are etched and gold-plated with standard PCB technology. The correct width of the gas gap is ensured by a G-10 frame glued to one of the two panels (‘gas frame’). Besides acting as spacer, the frame guarantees gas tightness to the detector and gas distribution. The gas input and output lines enter the detector in the large side of the trapezoid and continue through a narrow duct machined through the full length of the sides: uniform gas distribution to the sensitive volume of the detector is achieved on each side via six equally spaced holes of 1.0 mm diameter.

The anode of the detector is composed of gold-plated (gold content of 6-8%) tungsten wires with 30 μm diameter; the wires are strung parallel to the base of the trapezoid. The support for the wires ('wire holder' in Figure 3.16) is provided by two printed circuit bars precisely machined to a thickness of 5.0 mm glued on the wire panel along the oblique sides of the detector and inside the gas volume. The first and the last anode wire, close to the inner and the outer edge of the detector, are field-shaping electrodes and have a larger diameter (100 μm). High voltage is applied on one side; on the opposite side the front-end card is directly soldered to pads connected to each single anode wire.

The cathode electrodes are parallel strips obtained as gold-plated tracks oriented at $\pm 60^\circ$ with respect to the direction of the wires and have 5.0 mm pitch (4.5 mm width and 0.5 mm separation). Each strip is connected to high-density connectors mounted outside the gas volume as shown in Figure 3.16.

3.6 T2 (Telescope 2)

3.6.1 Detector geometry

GEM technology has provided excellent results of the COMPASS experiment, obtained during several years in high-rate environment. Consequently, the COMPASS GEM design was adopted as a guideline for the GEMs of the TOTEM T2 telescope.

The T2 telescopes are installed in the forward shielding of CMS between the vacuum chamber and the inner shielding of the HF Calorimeter [AAA⁺08].

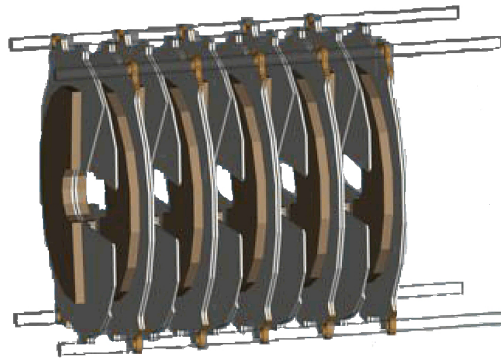


Figure 3.17: TOTEM T2 CAD drawing

In each T2 arm, 20 semi-circular Gaseous Electron Multiplier (GEM) planes –with overlapping regions –are interleaved on both sides of the beam vacuum chamber to form ten detector planes (Figure 3.17). In each detector layer, two GEM half-planes are slid together for maximal azimuthal angle coverage. With the ten double detector layers both high efficiency for detecting the primary tracks from the interaction point and efficient rejection of interactions with the LHC vacuum chamber is achieved.

The GEMs are installed as pairs with a back-to-back configuration. This arrangement of active detector planes allows both track coordinates and local trigger –based on hit multiplicities and track routes pointing to the interaction region –to be obtained. The material budget of T2 telescopes is minimised by using low-Z construction materials and honeycomb structures in manufacturing the GEM support mechanics.

3.6.2 Description of the T2 GEM detectors

The shape of the GEM detector used in T2 telescope is semi-circular with an active area covering an azimuthal angle of 192° and extending from 43 mm up to 144 mm in radius from the beam axis (Figure 3.19).

The T2 GEM foils consists of $50\ \mu\text{m}$ polyimide foil with $5\ \mu\text{m}$ copper cladding on both sides. Due to the bidirectional wet etching process used by the workshop the shapes of the holes are double conical. The diameters of the holes in the middle of the foil and on the surface are $65\ \mu\text{m}$ and $80\ \mu\text{m}$, respectively.

Three GEM foils are used as a cascade in one detector (Figure 3.18) to reduce the discharge probability below 10^{-12} [ABd⁺04]. For the same reason the voltage divider supplying the voltages for the foils is designed such that the potential difference is gradually decreasing from the uppermost foil to the lowest one (nearest to the readout board). Moreover, the high voltage side of each foil is divided into four concentric segments for limiting the energy available for sparks. The segments are biased separately through high voltage resistors, enabling switching off the innermost segment if required. The ground sides of the foils are continuous.

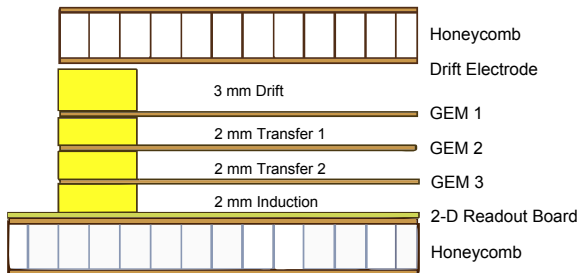


Figure 3.18: A side view of the TOTEM T2 GEM detector structure

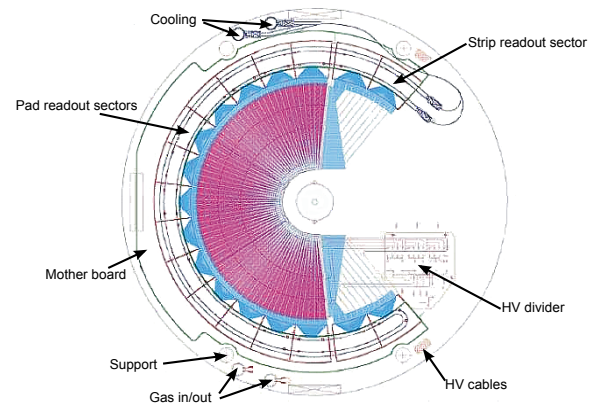


Figure 3.19: The design drawing of the TOTEM T2 GEM detector

A 3 mm drift space is followed by two 2 mm deep charge transfer regions (Transfer 1 and Transfer 2) and a 2 mm charge induction space. The large signal charges are collected, in two dimensions, by a readout board underneath of the induction layer. The lightweight construction and support materials are chosen for low-Z material budget and mechanical robustness. The thickness of the drift space is 3 mm, whereas the transfer 1 and 2 and the induction gaps are all 2 mm (see Figure 3.18). The corresponding electric fields over the gaps are approximately 2.4 kV/cm and 3.6 kV/cm, respectively.

The GEM foils are stretched and glued over supporting frames, made from fiberglass reinforced epoxy plates (Permaglas) with thicknesses of 2 mm. Two additional supporting spacers of thickness 0.5 mm are designed in the middle of the frames. Their position is slightly asymmetric to minimise dead areas. The drift frame is similar, except that the thickness is 3 mm and that no thin spacers in the middle of the frame are used.

A printed circuit board covered by polyimide foil with a pattern of strips and pads is used as a two-dimensional readout board. The readout board contains 2×256 concentric strips for the radial coordinates and a matrix of 1560 pads for azimuthal coordinates and for the T2 local trigger (Figure 3.19).

3.7 The TOTEM radiation environment

3.7.1 Summary of simulation results

In this section a summary of the radiation levels expected in the different TOTEM detectors is given. All reference values from Monte Carlo simulations are quoted at nominal LHC luminosity. However it is important to remember that the full luminosity operation will be only achieved at the end of a commissioning period. Therefore, the LHC will reach nominal performances towards a series of stages in which all machine parameters (e.g. luminosity, beam intensity, bunch spacing, etc.) will be considerably reduced. For this reason, during the first years of operation, the corresponding radiation field intensities can be reduced by several orders of magnitude with respect to the annual estimations quoted in the following points.

3.7.2 Roman Pots radiation environment

The location where the front-end electronics for the Roman Pot silicon detectors sits, the Motherboard (MB), is about 20 cm from the LHC beam-pipes. The radiation environment in this region of the LHC has been simulated in [MRKS03]. According to these results, the expected radiation dose will be about 5 kGy/year for the 147 m. RP station, while a factor 10 lower is expected for the 220 m. RP station. The annual global F of neutrons and charged hadrons is of about $7.5 \times 10^{12} \text{ cm}^{-2}$ at the 147 m RP station and $7.5 \times 10^{11} \text{ cm}^{-2}$ at the 220 m RP station. All the simulations have been done for the nominal luminosity of $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ assuming an LHC operation time of $1.5 \times 10^7 \text{ sec/year}$.

The radiation field in these locations will be dominated mostly by high energy photons (92%). The contributions of charged hadrons, neutrons and the electromagnetic component rise up to 8%.

Since the Motherboard PCB extends over 22 cm length in the radial direction, the aforementioned values present the highest expected levels, however they are a significant estimation for the selection of the RADMON sensors.

3.7.3 T1 radiation environment

The radiation field in the T1 region has been simulated in the context of the CMS experiment [Huh06]. In the CMS end-cap region, at Z ranging from 500 to 1100 cm and for $R < 110 \text{ cm}$, a maximum dose of about 15 kGy/year and an annual $F_{n > 100 \text{ keV}}$ of about $1.5 \times 10^{13} \text{ cm}^{-2}$ have been evaluated at $10^{34} \text{ cm}^{-2}\text{s}^{-1}$. Although this value does not represent the real dose to which the T1 detector is exposed during its operation, these values are a significant estimation for the selection of the RADMON sensors.

3.7.4 T2 radiation environment

The radiation field in the T2 region has been also simulated in the context of the CMS experiment [Huh06]. A maximum dose of about 10 kGy/year (at the radius $r > 15 \text{ cm}$ in the T2 reference system where the electronic equipment is located) and an annual $F_{n > 100 \text{ keV}}$ close to $1 \times 10^{15} \text{ cm}^{-2}$ have been evaluated at $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ for the T2 region. At $r < 15 \text{ cm}$ the calculated annual dose increases by about one order of magnitude.

(This page is intentionally left blank)

Chapter 4

Data flow

4.1 Readout

The architecture of the TOTEM Readout System is common to all the three detectors.

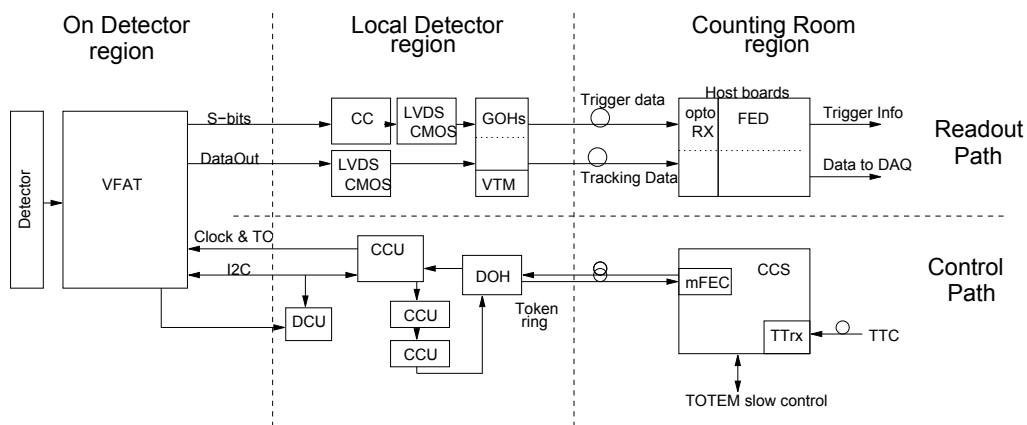


Figure 4.1: Functional block diagram of the TOTEM electronics system architecture

Figure 4.1 shows a basic block diagram of the functional components used in the system. It is subdivided into physically separated regions and data flow. The ‘On Detector regions’ have the VFAT front-end ASIC located as close to the detector as physically possible. The ‘Local Detector regions’ are for readout boards in the vicinity of the detector used for grouping and distributing control signals.

VFAT produces both Trigger and Tracking data. These two types of data have very different timing requirements hence are treated separately.

4.2 On line

4.2.1 Architecture

This section gives an overview of the online software architecture. That means all the software tools and services needed for the transportation and processing of data as well as for the configuration, control and monitoring of all devices during the detector operation.

The top level block diagram of the online software is shown in Figure 4.2. The architecture consists of five main parts [CMS02]:

- Cross-platform DAQ framework: XDAQ (Section E.3, *Cross-platform DAQ Framework (XDAQ)*)
- Data acquisition components
- Run Control and Monitor System (RCMS) (Section E.4, *Run Control and Monitoring System*)
- Detector Control System (DCS) (Chapter 5, *State of the art of the LHC control systems*)
- Detector Safety System (DSS) (Section E.1, *Detector safety system*)

The RCMS, DCS and data acquisition components interoperate through a distributed processing environment called XDAQ (cross-platform DAQ framework) to achieve the functionality required to operate the data acquisition system of the experiment. All components participate in the data acquisition process through a uniform communication backbone, where a common set of rules have been defined for the format of exchanged data.

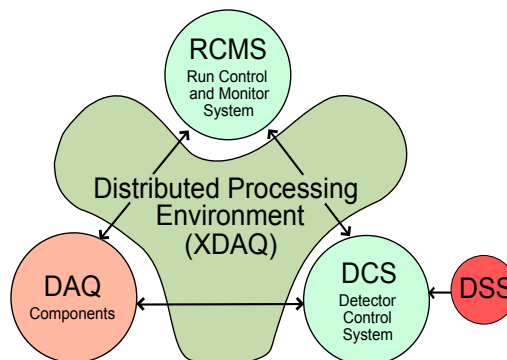


Figure 4.2: Overall online software architecture. Circles represent subsystems that are connected via XDAQ.

4.2.2 DAQ (Data Acquisition Components)

The collection of DAQ components includes applications such as the distributed Event Builder (EVB), detector electronics configuration and monitoring components (FEC and FED), and software-based trigger-throttling mechanisms (aTTS). These applications require interfaces to the Front-End Drivers, the trigger system, the high-level trigger services as well as the DCS. The relevant RCMS services connect to the data acquisition components to steer the data acquisition.

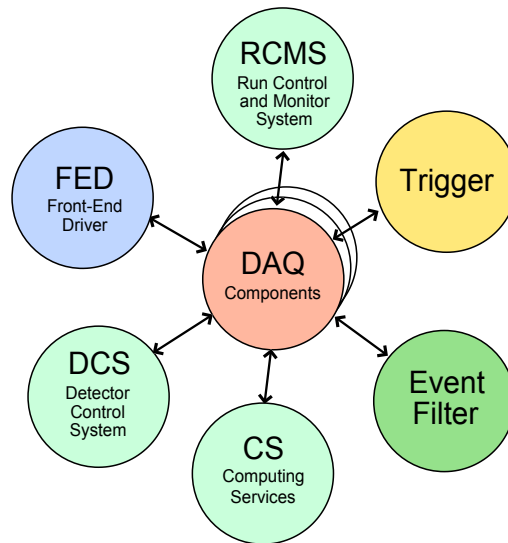


Figure 4.3: DAQ components interface. Circles represent subsystems internal to the DAQ.

4.3 Databases

4.3.1 TOTEM data architecture

The overall TOTEM data architecture is defined in [GTR⁺08]. In this set of documents all data aggregates are identified. They also reflect how the different subsystems need to exchange pieces of information during the experiment lifetime. Figure 4.4 presents the summary diagram of all this relationships; while Section 5.5, *Databases* explains in detail the DCS related databases. The direction of the arrows means that the origin block needs to access the information of the destination block.

4.3.2 Oracle 11g

CERN main storage uses an homogeneous database service based on Oracle [KR05].

In a Real Application Clusters (RAC) set-up the Oracle 11g ‘service’ concept allows one to structure larger clusters into groups of nodes that are allocated to particular database applications (e.g., online configuration DB, Grid file catalogue). By adding nodes to a cluster, the number of queries and concurrent sessions can be increased together with the total amount of database cache memory, which is shared across all cluster nodes. This preallocation of cluster resources is required to limit inter-node communication and to isolate key applications from lower-priority tasks executing on the same cluster. How far a RAC set-up will be able to scale for a given application depends on the application design. Limiting factors are typically inter-node network communication (cache coherency) and application contention on shared resources.

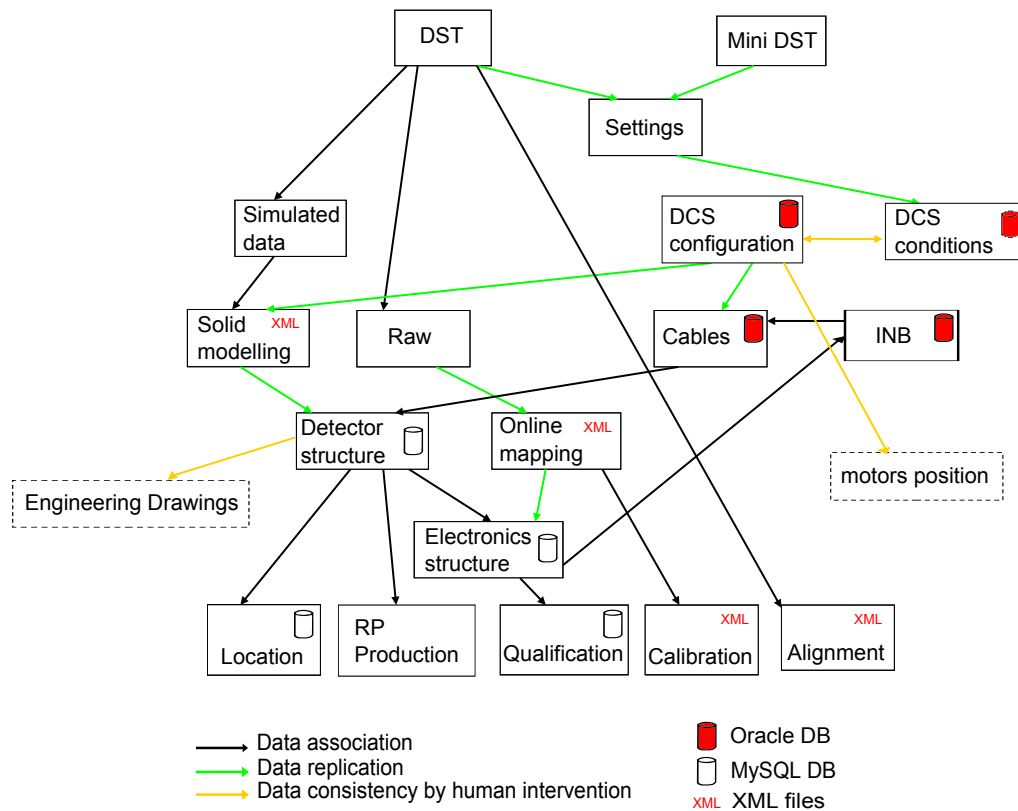


Figure 4.4: TOTEM data architecture

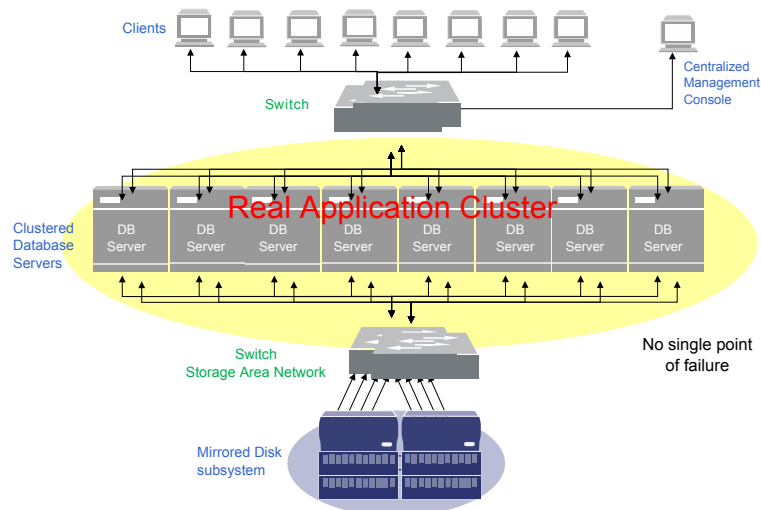


Figure 4.5: Oracle RAC diagram: the cluster database

4.4 Off line

4.4.1 The grid

Hierarchy

The LHC Computing Grid (LCG) Project will implement a Grid to support the computing models of the experiments using a distributed four-tiered model. The data from the LHC experiments will be distributed around the globe, according to this hierarchical model; see [KR05] and Figure 4.6).

- The original raw data emerging from the data acquisition systems will be recorded at the Tier-0 centre at CERN. The maximum aggregate bandwidth for raw data recording for a single experiment (ALICE) is 1.25 GB/s. The first-pass reconstruction will take place at the Tier-0, where a copy of the reconstructed data will be stored. The Tier-0 will distribute a second copy of the raw data across the Tier-1 centres associated with the experiment. Additional copies of the reconstructed data will also be distributed across the Tier-1 centres according to the policies of each experiment.
- The role of the Tier-1 centres varies according to the experiment, but in general they have the prime responsibility for managing the permanent data storage –raw, simulated and processed data –and providing computational capacity for re-processing and for analysis processes that require access to large amounts of data. At present 11 Tier-1 centres have been defined (Table 4.1), most of them serving several experiments.
- The role of the Tier-2 centres is to provide computational capacity and appropriate storage services for Monte Carlo event simulation and for end-user analysis. The Tier-2 centres will obtain data as required from Tier-1 centres, and the data generated at Tier-2 centres will be sent to Tier-1 centres for permanent storage. More than 100 Tier-2 centres have been identified.
- Other computing facilities in universities and laboratories will take part in the processing and analysis of LHC data as Tier-3 facilities.

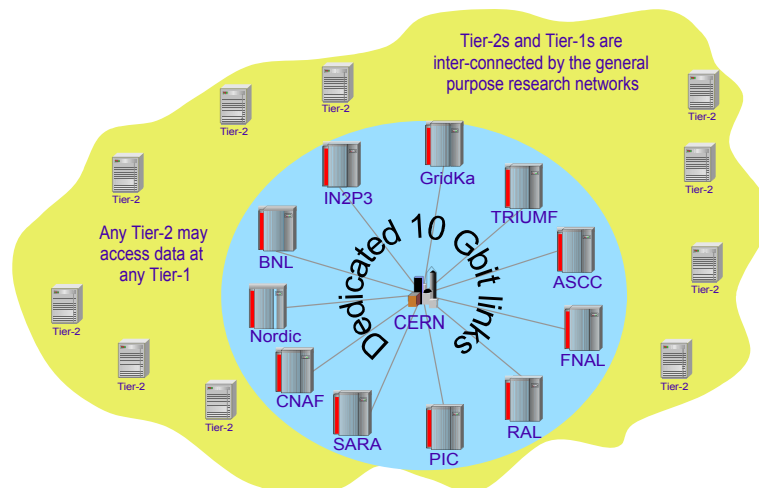


Figure 4.6: Tier-0, Tier-1 and Tier-2

Basic Tier-0 to Tier-1 Dataflow

Data coming from the experiment data acquisition systems is written to tape in the CERN Tier-0 facility, and a second copy of the raw data is simultaneously provided to the Tier-1 sites, with each site accepting an agreed share of the raw data. How this sharing will be done on a file-by-file basis will

be based on experiment policy. The File Transfer Service (FTS) will manage this data copy to the Tier-1 facilities in a reliable way, ensuring that copies are guaranteed to arrive at the remote sites. As this data arrives at the Tier-1, it must ensure that it is written to tape and archived in a timely manner. Copies arriving at the Tier-1 sites should trigger updates to the relevant file and data location catalogues.

The networking among the Tier-0 and the Tier-1 is based on a permanent Optical Private Network (OPN) for the LHC Grid, with a bandwidth of 10 Gigabit/s.

Raw data at the Tier-0 will be reconstructed according to the scheme of the experiment, and the resulting datasets also distributed to the Tier-1 sites. This replication uses the same mechanisms as above and again includes ensuring the update of relevant catalogue entries. In this case, however, it is anticipated that all reconstructed data will be copied to all of the Tier-1 sites for that experiment.

Tier-1 Centre	Experiments served with priority			
	ALICE	ATLAS	CMS	LHCb
TRIUMF, Canada		X		
GridKA, Germany	X	X	X	X
CC, IN2P3, France	X	X	X	X
CNAF, Italy	X	X	X	X
SARA NIKHEF, NL	X	X		X
Nordic Data Grid Facility (NDGF)	X	X	X	
ASCC, Taipei		X	X	
RAL, UK	X	X	X	X
BNL, USA		X		
FNAL, USA			X	
PIC, Spain		X	X	X

Table 4.1: Relationship among the Tier-1 and experiments

4.4.2 Reconstruction

After the collision in the interaction point, some of the intact protons traverse the Roman Pot devices and deposit the energy in the silicon detectors which is converted into strip hits recorded by the Data Acquisition System. The objective of the RP reconstruction software is to find, basing on the RP strip hits and the machine optics model, the proton kinematics in the interaction point after the collision.

A schematic diagram of the Roman Pot reconstruction software is presented in Figure 4.7 from [Nie08], which shows its most important modules and their interactions.

The reconstruction input data may come from the simulation software (Step 1a), from the beam tests (Step 1b) and from the real experiment (Step 1c). The RP reconstruction software can cooperate with the reconstruction modules of the T1 and T2 TOTEM telescopes (Step 8). Another important reconstruction input is the detector geometry. As in the case of the simulation software, two types of geometry are used: ideal and real geometry (Step 2a). The geometry is used in the various stages of the RP local track reconstruction.

The aim of the RP local reconstruction is the computation of the proton tracks in the Roman Pot devices on the basis of the strip hits provided at the reconstruction input. The strip hits are transformed into strip clusters (Step 3), which, afterwards, with the help of geometry information, are converted into spatial points (Step 4). The pattern recognition reconstruction stage (Step 5), is responsible for discovering the proton track candidates composed of the spatial points. The road search algorithm is applied to find the candidates approximately parallel to the beam. Finally, the RP track candidates are fitted with straight lines (Step 6), which constitute the input to the proton reconstruction modules (Step 7).

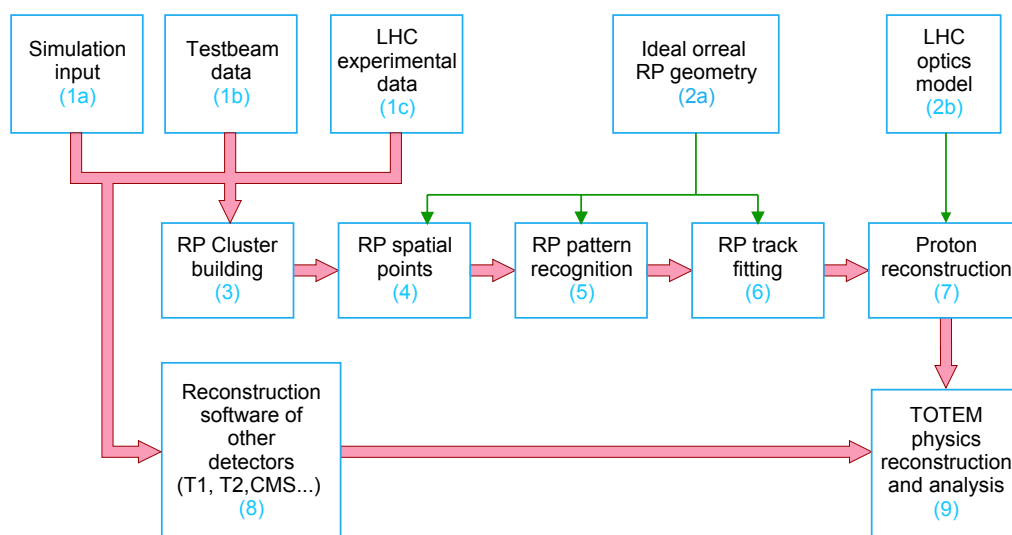


Figure 4.7: Work flow diagram of the TOTEM Roman Pot reconstruction software

(This page is intentionally left blank)

Chapter 5

State of the art of the LHC control systems

5.1 Introduction

The Detector Control Systems (DCS) for LHC experiments are the evolution of slow control systems of the LEP era. Their aim is to permit the physicist on shift to operate and control various detector subsystems such as high and low voltage power supplies, gas circulation systems, cooling and so on, and monitor their performance as well as various relevant environmental parameters, for example temperature and radiation levels. All DCS systems of LHC experiments are built using the industrial PVSS control supervisory software running on networks of PCs (Windows and/or Linux), augmented with modules developed at CERN for typical HEP control functions and equipment, the JCOP framework. Big.brother, the TOTEM DCS system, also uses the same guidelines, technology, tools and components.

5.1.1 Objectives

The main aim of the Detector Control System (DCS) is to ensure the correct operation of the experiment [Mye99], so that the data taken with the apparatus is of high quality. The scope of the Detector Control System (DCS) is therefore very wide and includes all subsystems and other individual elements involved in the control and monitor of the detector, its active elements, the electronics inside and outside the detector, the experimental hall as well as communications with the accelerator. The DCS also plays a major role in the protection of the experiment from any adverse events [ABB⁺02b]. Many of the functions provided by DCS are needed at all times, and as a result selected parts of the DCS must function continually on a 24-hour basis during the entire year.

The primary function of the DCS will be the overall control of the detector status and its environment. In addition DCS has to communicate with external entities, in particular the RCMS and the accelerator. The DCS will be a slave to the Run Control and Monitor System which will be in charge of the overall control and monitoring of the data-taking process. When the DCS will operate outside data-taking periods it will act as the master of the active detector components and other parts of the system. The major tasks in this set will be:

- central supervision of all DCS elements
- communication with external systems (RCMS, accelerator, DAQ, motorization, ...)
- user interfaces
- bookkeeping of detector parameters and used DCS hardware/software components

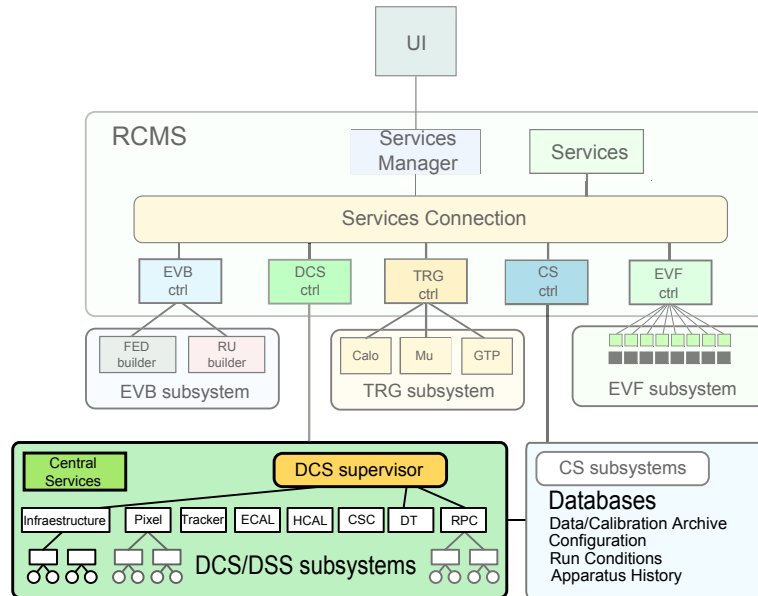


Figure 5.1: Integration of DCS in the CMS online system

Another major task of the DCS is the control and monitoring of the systems regulating the environment at and near the experiment, a set of tasks traditionally referred to as ‘Slow Controls’, and which include, among others:

- Handling the supply of electricity to the detector
- Environment monitoring (temperature, radiation, vacuum, ...)
- Other DCS-related detector electronics (e.g. calibration systems)
- Control of the cooling facilities and the overall environment near the detectors
- Supervision of all gas and fluids subsystems
- Control of all racks and crates

5.1.2 Requirements

The major system-wide requirements on a DCS are:

- Reliable: this requires safe power, redundancy and reliable hardware in numerous places.
- Partitionable in order to allow independent control of individual detectors or detectors parts. Conversely, integration of different parts into the global system must be possible with ease.
- Automation features: in order to speed up the execution of commonly executed actions and also to avoid mostly human mistakes in such repetitive actions.
- Easy to operate: a few non experts should be able to control the routine operations of the experiment.
- Generic interfaces to other systems, e.g. the accelerator, the Run Control and Monitor system.
- Easy to maintain. This requirement strongly favors the usage of commercial hardware and software components.

- Homogeneous, which will greatly facilitate its integration, maintenance and possible upgrades, and has to have a uniform ‘look and feel’ throughout all of its parts.
- Remote control via the internet to allow non CERN resident detector groups to do maintenance and control from their home station.

5.2 Control functions level

5.2.1 High voltage control

The DCS will then communicate with the CAEN systems using its built-in OPC client. The corresponding implementation based on PVSS II is delivered within the JCOP framework.

5.2.2 Low voltage control

Similarity for LV, the usual provider is Wiener, and also an OPC server is provided using Ethernet connectivity. Also is important to mention that those power crates are autosensing. They need an additional line apart of the main power to check the voltage provided at the end of the line and readjust it consequently.

5.2.3 Environment monitoring

Sensors will be standardized as much as possible, as have to work mostly in a hostile environment with radiation and a magnetic field. In this way the calibration and degradation data can be shared among experiments. The most usual ones are:

- Temperature measurement
- Humidity measurement
- Pressure measurement
- Radiation measurement

5.2.4 Cooling monitoring

To this end the JCOV (Joint COoling and Ventilation) project has been created. The sensors and actuators will be connected via I/O modules embedded in the PLC frame. The software communication will be done using the MODBUS protocol.

5.2.5 Gas control

The control of all CERN gas systems covering hardware and software will be designed, installed and maintained centrally by the CERN gas working group. A PVSS II system for the experiments gas control will be provided by them with specific user interfaces for all the detectors using gas.

5.2.6 LHC status monitoring

The DCS has to make sure that the detector is in an appropriate state (e.g. Roman Pots retracted) before LHC is allowed to inject particles.

Machine parameters like status signals for setting up, shut-down, controlled access, stable beams, beam cleaning must be transferred from the accelerator to the experiment; see Section 6.3, *Beam Instrumentation Signals*. The machine should also provide information on the beam like emittance, focusing parameters, energy, number of particles per bunch, a horizontal and vertical profile, needed for offline physics analysis. Information on the vacuum conditions in the vicinity of and in the experimental straight section and position of the collimator are also of interest to the experiment.

5.2.7 DSS monitoring

The DCS will show the values from all the sensors connected to the DSS. In the same way, alarms generated by the DSS system will be also shown in the DCS user interface. All this communication is done using by software the DIP protocol.

5.2.8 Rack control

The control of the main power distribution to the racks will be done using the equipment of the global power infrastructure of CMS [Gle08]. A small control unit housed in the turbine case will measure the air flow and temperatures in the racks and do smoke detection. It will report those values to the DCS system via CANBUS. In case of emergency (e.g. overheating or fire) the power to the rack will be cut via a hardwired safety line to the main power distribution.

5.2.9 VME crates control

VME crates control in CMS is done through the same CANBUS used for the racks that house the crates. In this way, the same interface between the crate controller and PVSS II can be used for the racks. A remote control of the fan unit and the power supply is provided.

For TOTEM due to the special relationship with CMS the rack monitoring will be done by CMS central DCS. TOTEM will communicate to CMS in software level using the PVSS capabilities for distributed projects to perform any action or monitor any value related to the racks.

5.2.10 Access control

Communication techniques based on the Internet will be offered for remote maintenance and surveillance of the experiment cavern and all its service caverns. This implies a security risk. An access control mechanism must be put in place in order to prevent any damage or other disturbance caused by such incidents. The DCS system will utilize standard encryption and authentication techniques. On the other hand access control will be needed to allow for certain actions depending on the expertise of the user. For instance an experimenter on shift might only be allowed to switch a detector on or off, while an expert might be allowed to also change various key voltage settings.

5.3 Communications level

5.3.1 DIM (Distributed Information System)

DIM, like most communication systems, is based on the client/server paradigm [GDC01]. The basic concept in the DIM approach is the concept of 'service'. Servers provide services to clients. A service is normally a set of data (of any type or size) and it is recognized by a name - 'named services'.

Servers 'publish' their services by registering them with the name server (normally once, at startup).

Clients 'subscribe' to services by asking the name server which server provides the service and then contacting the server directly, providing the type of service and the type of update as parameters.

Data can be updated by the server either at regular time intervals or whenever the conditions change (according to the type of service requested by the client). The client updating mechanism can be of two types, either by executing a callback routine or by updating a client buffer with the new set of data, or both at the same time.

5.3.2 DIP (Distributed Interchange Protocol)

DIP is a system which allows relatively small amounts of real-time data to be exchanged between very loosely coupled heterogeneous systems. These systems do not need very low latency. The data is assumed to be mostly summarized data rather than low-level parameters from the individual systems, e.g. cooling plant status rather than the opening level of a particular valve.

Currently DIP is based on DIM and they are very similar conceptually [BS04].

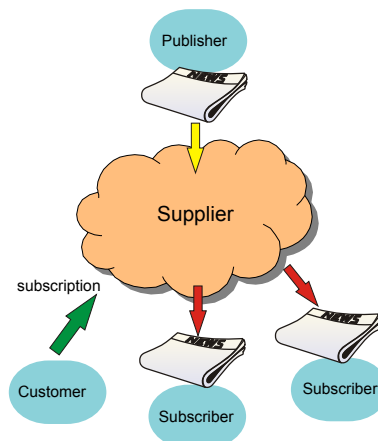


Figure 5.2: DIP communications schematic

5.3.3 OPC (OLE for Process Control)

OPC (OLE for Process Control), defines a set of interfaces, based on OLE/COM and DCOM technology, for truly open software application inter-operability between automation/control applications, fieldbus/device and business/office applications.

OPC is managed by the OPC Foundation [OPC96], which is comprised of more than 220 companies and institutes as members. The foundation has defined and released five sets of interfaces:

- OPC Data Access
- OPC Alarms and Events
- OPC Batch Interface
- OPC Historical Data
- OPC Security

The most interesting interface, for the applications in the DCS system, is the OPC Data Access. This interface prevents the need for specific drivers to connect to commercial hardware. Essentially all industrial hardware are marketed with an interface of this type. The software that provides this interface is called an ‘OPC server’. Control software, like a SCADA system, are ‘OPC clients’ and can be easily connected to an OPC server. OPC, which is only available on the Windows platform, is currently the JCOP recommended way to connect commercial controls hardware to the software level. OPC servers for custom hardware can also be easily developed using dedicated toolkits, as has been the case for the ELMB.

5.3.4 XDAQ (Cross-platform DAQ framework)

See Section E.3, *Cross-platform DAQ Framework (XDAQ)*.

5.3.5 CMW (Controls MiddleWare)

The controls Middleware (CMW), described in [BCC⁺03] and in [KAd⁺03], project provides a common software communication infrastructure for the CERN accelerator controls. This infrastructure is successively replacing existing heterogeneous software protocols and provides new facilities for the LHC era. In particular the project supports the Accelerator Device Model and device I/O services, the publish/subscribe paradigm synchronized with Accelerator Timing, and interoperability with industrial control systems.

CMW is structured as a client/server model. At the heart of CMW is the Remote Device Access (RDA) system, which defines the client and the server API and provides the communication on top of CORBA.

5.4 Supervisory level

5.4.1 Introduction

Systems used for the supervision level of industrial controls applications are called SCADA (Supervisory Controls And Data Acquisition) systems. A SCADA system includes input/output signal hardware, controllers, HMI (Human Machine Interface), networks, communication, database and software. It mainly comes in the branch of Instrumentation Engineering.

The selection of a single commercial standard framework for the controls of all the detectors in the experiment allows for the development of a homogeneous system. A typical SCADA system is based on the experience of many people in this field, of perhaps many hundreds of man-years, in the development and debugging before becoming a stable and mature system. Support and maintenance, including documentation and training, is provided by the company.

5.4.2 PVSS (Process Visualization and Control System)

A commercial SCADA system named PVSS II [ETM08] (meaning Process Visualization and Control System II; german acronym) has been chosen by all LHC experiments as the supervisory system of the corresponding DCS systems. PVSS II is a development environment which offers many of the basic functionalities needed to fulfil the tasks mentioned above. Also PVSS II components for use by all LHC experiments are either already developed and maintained, or are currently under development centrally within the framework of the CERN-wide 'Joint Controls Project', JCOP. The framework provides the tools to build a partitionable controls hierarchy as well as the PVSS II implementation of certain hardware devices.

The LHC experiments have chosen PVSS II as the SCADA system for the control of the experiments. This system is essentially a toolkit that delivers the building blocks of a control system. It will be the basic skeleton of the DCS by housing many central components like:

- Database
- Network interconnection
- Graphical user interface
- Hardware connectivity

Every PVSS II system can consist of all these components. Several PVSS II systems can be connected together to enhance the capabilities and offered resources and to distribute the load. A system can be spread over several PCs or more than one system can be run on one PC. This design makes PVSS II highly scalable and allows independent development of subparts of the DCS. PVSS II offers many features that are needed for a control system:

- Access control
- Alarm handling
- API (Application Programmable Interface)
- Archiving
- C-like scripting
- Development tools
- HMI (Human Machine Interface)
- Logging
- Networking
- Redundancy manager
- Trending

Each PVSS II system has always one event manager and one database manager. It can have multiple control managers (for the scripts), API managers, drivers (connected to the hardware) and user interface managers. Each of these managers can run on a separate processor if required. The mix of ready to use functional components and easy and open programming capabilities simplify the development and offer the flexibility required. The PVSS II system runs on Windows and Linux and one can mix both operating systems even within a single PVSS II system, as long as no external component used, like OPC, restricts to one.

5.4.3 JCOP (Joint COntrols Project) FrameWork

Over PVSS II and using its extension capabilities a specific framework for CERN technologies has been developed.

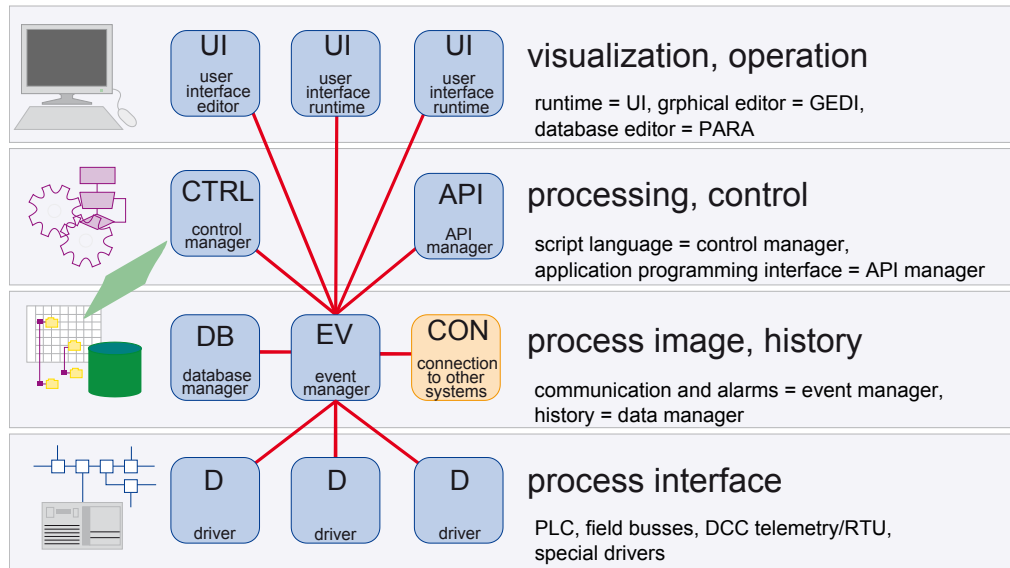


Figure 5.3: PVSS managers

The Framework is an integrated set of guidelines and software tools which is used by developers of the Control System to build their part of the Control System application. When all parts of the application have been developed and integrated these form the complete Control System (from a software point of view).

The objectives of this project are defined in [Mye99]. A set guidelines and conventions to follow when developing components of the LHC Experiment Control Systems, as well as an overview of the components part of the Framework, are given in [JCO07].

5.4.4 SMI++ (State Management Interface)

SMI++ is a framework for designing and implementing distributed control systems. SMI++ is based on the original State Manager concept which was developed by the DELPHI experiment in collaboration with the DD/OC group of CERN.

In this concept, a real-world system is described in terms of objects behaving as Finite State Machines. The object model of the experiment is described using a dedicated language State Manager Language (SML) defined in [Gas04]. This language allows detailed specification of the objects such as their states, actions and associated conditions. The objects representing concrete entities interact with the hardware they model and control through driver processes or proxies. The objects are typically organised in hierarchical structures called domains.

SMI++ objects can run in a variety of platforms, all communications are handled transparently by the DIM protocol. Additionally a binding to PVSS and LabView has been developed.

5.4.5 Command hierarchy

Detector controls will be organized in a hierarchy of FSM nodes using SMI++ as shown schematically in Figure 5.5. The implementation of this organization will be made purely in software and only represents

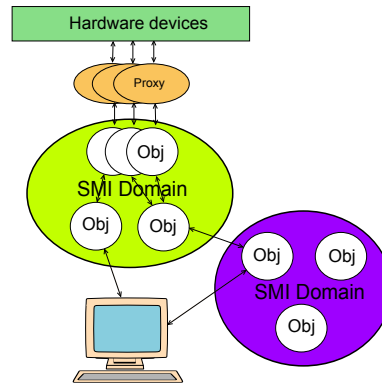


Figure 5.4: Basic concepts of SMI++

the logical structure of the detector. The interfaces of all nodes in this hierarchy are made such that each node can be connected to any other node.

Therefore each node can be placed anywhere in the hierarchy. This allows for independent development of the detector systems while it will greatly facilitate their integration into the full system.

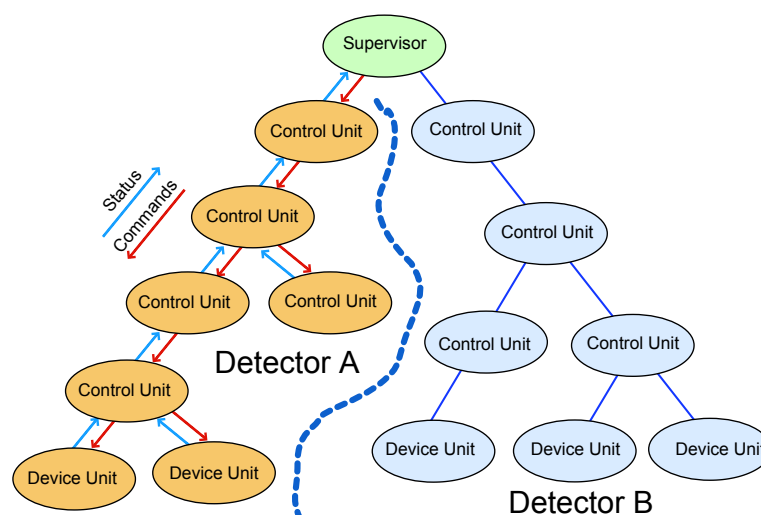


Figure 5.5: Illustration of command hierarchies in the DCS. Commands flow downwards, status flows upwards

The DCS supervisor, at the topmost point of the hierarchy, offers global commands like 'Start' and 'Stop' for the entire detector as defined in [GGV06]. The commands are propagated towards the lower levels of the hierarchy, where the different levels interpret the commands received and translate them into the corresponding commands specific to the system they represent. As an example, a global 'Start' command is translated into a 'HV ramp up' command for a detector. Correspondingly, the states of the lower levels are summarized by the upper levels and thus define the state of the upper levels. As an example, the state 'HV on' of a detector is summarized as 'running' in the global state. The propagation of commands ends at the lowest level at the 'devices' which are representations of the actual hardware. The SCADA system ensures that every action on the devices is transmitted to the hardware and that the state of the devices represent the hardware status.

5.4.6 Partitioning

Partitioning implies that a subtree is cut off from the command hierarchy. In this way components can be operated independently from the rest of the tree thus rendering the operation of the corresponding detectors independent of the rest of the system [CMS05]. This mode of operation will be used mainly for maintenance, calibration, system testing and trouble shooting. Under normal operating conditions, e.g. during a ‘physics run’, there will be only one partition, comprising all detectors participating in the data-taking.

The mechanism of partitioning is illustrated in Figure 5.6. Partitioning is introduced into the command hierarchy by declaring a node as partitionable. This adds three parameters to each node:

- actual owner of node
- possibility to ignore states of lower level nodes
- possibility to ignore commands of upper level nodes

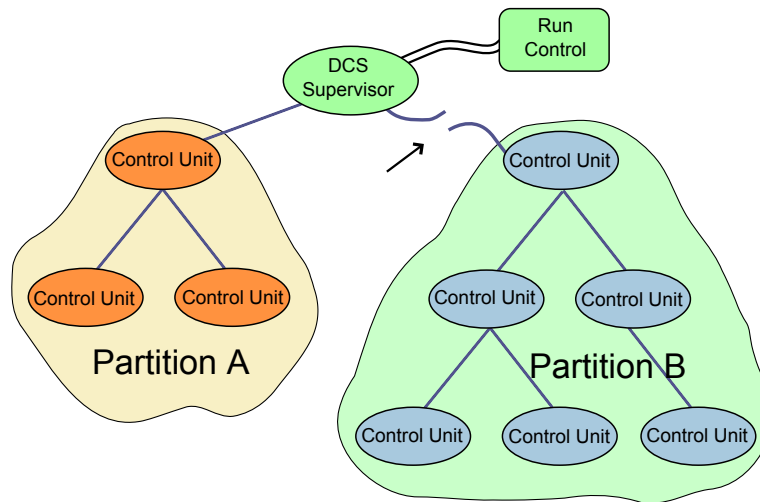


Figure 5.6: Illustration of the mechanism for partitioning the DCS

Tools to change the status of all three properties exist in the JCOP framework. The tools ensure that all nodes in a partition are owned by the same user and that the control of nodes in a partition can only be exercised through the partition's root node. The root node is the top most element in a tree or a branch and has therefore only children and no parents. For instance in Figure 5.6 the root node of detector B is the top most control unit in detector B after partitioning it from the DCS supervisor. However the owner can decide to share a partition and so a different user can also send commands to it.

5.5 Databases

5.5.1 Conditions database

The conditions database will hold the data describing the detector environment at data taking necessary for the offline reconstruction. This will allow for fine tuning the physics event reconstruction as well as for getting the status of all detector components at any point in time. Since the DCS system will control the environmental setup of the detector (HVs, LVs, Gas,...), selected data from the DCS will have to be written into the conditions database. It is foreseen to provide two write mechanisms:

- **on change:** a parameter is written whenever it changes.
- **periodically:** a parameter is written periodically after a certain time interval.

Whether a parameter has to be written and by which mechanism will be a configurable property of the parameter. To ensure the synchronization between the parameters in the conditions database and the appropriate ones in PVSS II, it will be possible to force an update of the parameters in the conditions database beyond the mechanisms described above.

5.5.2 Configuration database

The aim of the Configuration Database Tool is to provide means to store and retrieve different sets of configuration data (e.g. power supply limits and sensing) for the control system in an external (Oracle) database. These includes static and dynamic configuration data for PVSS datapoints.

This feature provides a fast way to reinstall the developed control system and to have different sets of configuration and also being able to track their changes over the time.

5.5.3 Geometry database

A detector (or any other machine) is basically a composition of parts. A part is characterised by its shape and material, and further by any part specific data like ‘manufacturer’, ‘service-interval’, ‘gas-gain’, ‘voltage’, ... Parts are composed out of other parts thus forming a hierarchy.

The Detector Description Language (DDL) defined in [CLvL05] provides generic XML constructs to describe materials (air, iron, etc.), shapes (box, cylinder, etc.), parts, compositions of parts, and specification of part specific data.

5.6 CERN Control Centre (CCC)

The LHC is not an isolated machine: it will be fed by a succession four increasingly large interconnected accelerators. The accelerators of CERN can transport several beams simultaneously and adapt each one to a given facility. It is this ability to deal with several beams at the same time that makes CERN a unique laboratory in its field of research. The old Proton Synchrotron (PS), which is the oldest accelerator in service at CERN, prepares beams for the LHC, while feeding the Antiproton Decelerator (AD) and other facilities with various particles.

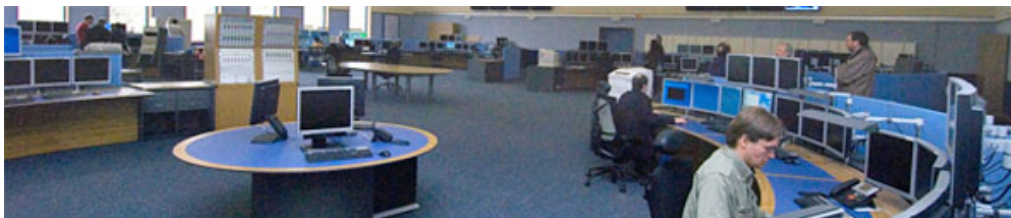


Figure 5.7: CERN Control Centre

(This page is intentionally left blank)

Chapter 6

Beam operation

6.1 Beam optics

A full description to the optics of an accelerator is given in [Bai07]. In this chapter only a few concepts will be raised out.

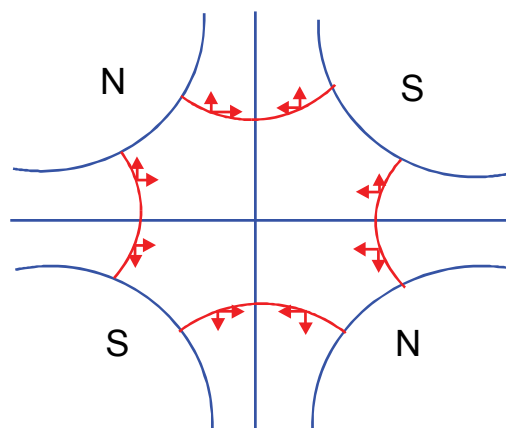


Figure 6.1: Force on a particle moving through a quadrupole

A quadrupole has 4 poles, (2 North and 2 South) arranged symmetrically around the beam, see Figure 6.1. Therefore a particle which deviates from the central axis of the quadrupole in the horizontal plane, is deflected back towards the centre of the magnet. Thus this magnet focuses particles in the horizontal plane. Unfortunately the opposite is true in the vertical plane, and it defocuses the beam vertically. This example is a Focusing Quadrupole (QF). It focuses the beam in the horizontal plane and defocuses the beam in the vertical plane.

Rotating the poles by 90° it would become a Defocusing Quadrupole (QD) by focusing in the vertical plane and defocusing in the horizontal plane. These quadrupoles can be considered as ‘thin lenses’ and the beam as a diverging light beam.

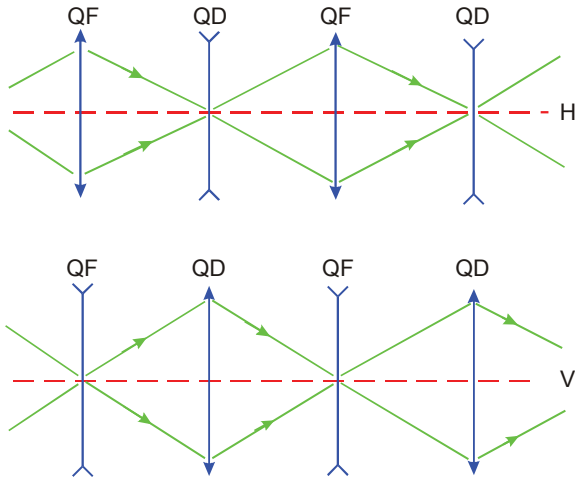


Figure 6.2: Light rays passing through a series of focusing and defocusing lenses

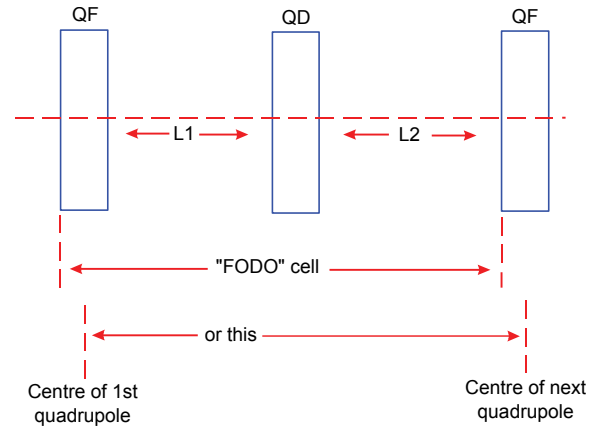


Figure 6.3: Layout of a FODO Cell

In Figure 6.2, the lenses, which are concave in one plane, are convex in the other. In both cases the concave lenses will have little effect as the light passes very close to their centre, and the net result is that the light rays are focused in both planes. The LHC accelerator consists of a series of dipoles to constrain the beam to follow some closed path, approximately circular, focusing and defocusing quadrupoles to provide the horizontal and vertical focusing needed to contain the beam in the transverse directions. A very common combination of focusing and defocusing sections is the FODO cell of Figure 6.3. This consists of alternate focusing and defocusing sections with non focusing - drift sections between them.

These transverse oscillations are called **Betatron Oscillations**, and they exist in both horizontal and vertical planes.

6.2 The LHC Machine Modes

During its operational cycle, the LHC proceeds through different phases referred as MODES [Lau06]. The mode of the machine is set manually by the machine operator or automatically by a software task that coordinates the control processes that drive the LHC from one mode to another as a response to beam conditions. A transition diagram for the modes is shown in Figure 6.4.

The most relevant modes in the TOTEM context are ADJUST, STABLE_BEAMS and UNSTABLE_BEAMS. The LHC mode information is available to the LHC experiments through the DIP data exchange protocol [Tse05] and over hardware link via the Safe LHC Parameter (SLP) System.

6.2.1 ADJUST mode

The ADJUST mode may be entered either from the RAMP mode or from the STABLE_BEAMS mode. It is normally followed by STABLE_BEAMS unless the beam is dumped before STABLE_BEAMS conditions are reached. In ADJUST mode the machine operation teams will perform large changes to the machine conditions at physics energy, like a beta squeeze or to bring the beams into collision. When this mode is entered from the STABLE_BEAMS mode, the control room crews will warn all the experiments.

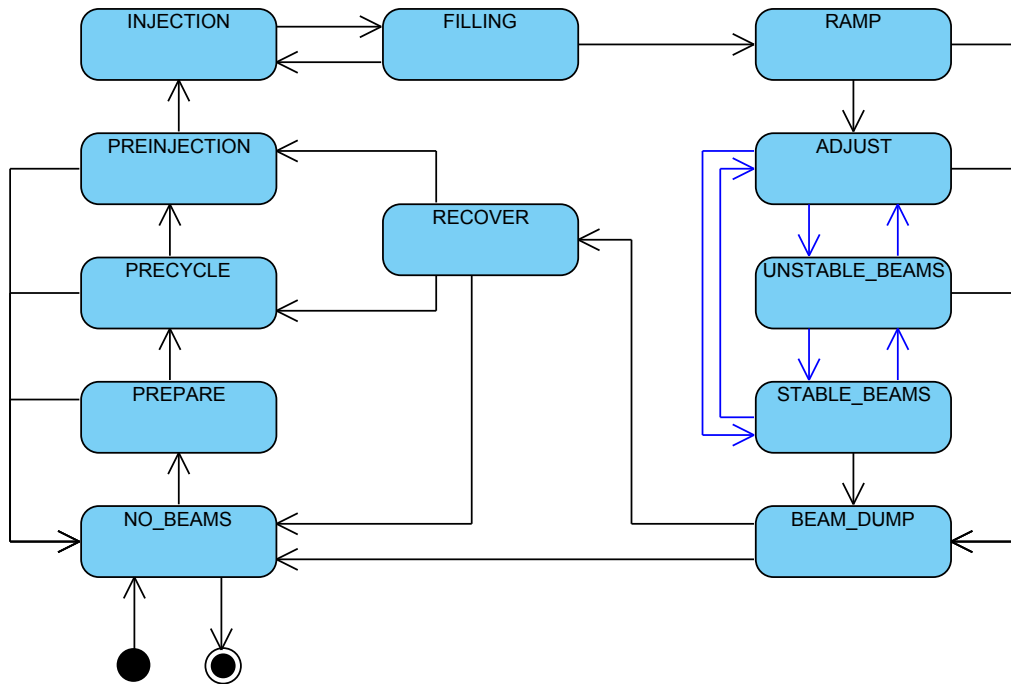


Figure 6.4: UML state model for the LHC based on [Lau06]

6.2.2 STABLE BEAMS mode

When the beam conditions and backgrounds are considered good and the beams are colliding at the interaction points, the machine mode is changed to STABLE_BEAMS which corresponds to the data taking periods for the experiments. In STABLE_BEAMS mode the background conditions must remain good and only minor adjustments are made to the beams to maintain the luminosity and background quality. It is important to realize that in the STABLE_BEAMS period, a feedback loop may be permanently acting on the beam.

6.2.3 UNSTABLE_BEAMS mode

The UNSTABLE_BEAMS mode will be entered when the machine is nominally adjusted for physics data taking although satisfactory conditions are no longer present. It will always be entered from the STABLE_BEAMS mode. The objective of this mode is to reestablish satisfactory physics conditions which may require measurements and changes to the machine settings that are more extreme than those usually done in STABLE_BEAMS mode.

This mode can be entered by the operation crews without any warning to the experiments. The detectors should be put in ‘standby’ mode (off, voltages ramped down,...) and the movable devices as the Roman Pots moved out of the beam.

6.2.4 BEAM DUMP

This mode will be entered when the operator wishes to inform the experimental teams that a scheduled beam dump is to be performed.

6.2.5 Mode Transitions

The communication between the LHC control room and the experiment will be done by a handshake mechanism built in top of DIP. This handshake will be answered by the experiment DCS as explained in Section 11.4, *LHC status and handshake*.

In general, beam conditions are expected to degrade either so suddenly that the operation crews will not have time to react and the beam will be dumped automatically or so slowly that the operation crews will have time to follow the procedure to go to the ADJUST mode which implies a pre-warning to the experiments. However, for the time being, nobody is able to say what the LHC operation will look like, see [LHC00].

6.3 Beam Instrumentation Signals

Table 6.1 provides a minimum set of data to be provided from the LHC accelerator to the experiments [Tse06] and [KRS⁺05]. As experience with the experiment and accelerator operation develops, this may result in the list or cardinality of signals exchanged to vary, thus altering the production rate and hence the data exchange rate.

Signal name	Type	Cardinality
Total Beam Intensity	Software	1
Individual beam Intensity	Software	2
Average 2D beam size	Software	2
Average bunch length	Software	1
Luminosity	Software	1
Average beam loss	Software	1
BPM	Software	12
Collimator settings	Software	8
40 MHz clock	Hardware	3
BPTX	Hardware	2
Energy	Hardware	2
Beam MODE	Hardware	1
DEVICE_ALLOWED	Hardware	1
IMMINENT_BEAM_ABORT	Software	1

Table 6.1: Signals created by LHC and sent to the experiments

The TOTEM experiment must also provide some other signals to the LHC. They are defined in [ORL06] and [Dut08] and listed in Table 6.2.

Signal name	Type	Cardinality
MOTOR_POWER_ON	Software	24
Resolver	Software	24
LVDT	Software	24
Event rates	Software	24
Radiation monitor (active)	Software	24
Detector power (HV)	Software	240
Radiation levels	Software	24
READY_FOR_BEAM_ABORT	Software	1
RP_OUT	Software	24
MOVEMENT_INHIBIT	Hardware	1
USER_PERMIT_1	Hardware	1
USER_PERMIT_2	Hardware	1
BEAM_PERMIT_1	Hardware	1
BEAM_PERMIT_2	Hardware	1

Table 6.2: Signals created by TOTEM about the RP and available to the LHC

The signals of Figure 6.3 are living only within the TOTEM domain and are not sent or received to and from other systems interacting outside the TOTEM experiment. The relationship among them and the rest of signals is explained in Chapter 12, *Integration with the motorization*.

Signal name	Type	Cardinality	Description
BACK_HOME	Software	1 per RP	Deduced by internal logic of the motorization, it triggers the extraction of a Roman Pot to its end switch at a fastest speed.
DEVICE_ALLOWED	Software	1	Defined as STABLE_BEAM or UNSTABLE_BEAM.
OVERRIDE	Hardware	1	Allows to operate the RPs in unstable beams.
RPS_PACK_OUT	Software	1 per beam	Indicates whether a set of RP on a same beam are at their parking position or not.

Table 6.3: TOTEM RP internal signals

6.4 Beam Interlocks

6.4.1 LHC Beam Interlock System (BIS)

The Beam Interlock System (BIS) of the LHC provides a hardware link from an external system such as an LHC experiment to the LHC Beam Dumping System, to the LHC Injection Interlock System and to the SPS Extraction Interlock System. Its architecture is described in detail in reference [PS06], while the operational procedure is taken from [MSW06].

The BIS permits injection into the LHC when all systems are ready for beam. When the beam is circulating in the LHC, the system transmits beam dump requests from connected systems to the LHC Beam Dumping System.

The BIS is split into a system for beam1 and another system for beam2 and handles the two independent BEAM_PERMIT signals, one for each beam. The BEAM_PERMIT is a signal that is transmitted over hardware links and that can have the values:

- TRUE
Injection of beam is allowed. With circulating beam, beam operation continues.
- FALSE
Injection is blocked. If a beam is circulating and the signal changes from TRUE to FALSE, the beam will be dumped by the Beam Dumping System.

The LHC beam operation is granted by the external systems when all the BEAM_PERMIT=TRUE and USER_PERMIT=TRUE for both beams.

The USER_PERMIT is another signal that is transmitted over hardware link (Beam Permit Loop) and that can have the values:

- TRUE
The user is ready and beam operation is allowed according to the user.
- FALSE
Beam operation is not allowed according to the user.

6.4.2 TOTEM Beam Interlocks

In TOTEM the beam related signals are directly dependent on the Roman Pots Motorization. Their generation have been defined by Mario Deile in [Dei08]. Figure 6.5, Figure 6.6 and Figure 6.7 summarizes this logic.

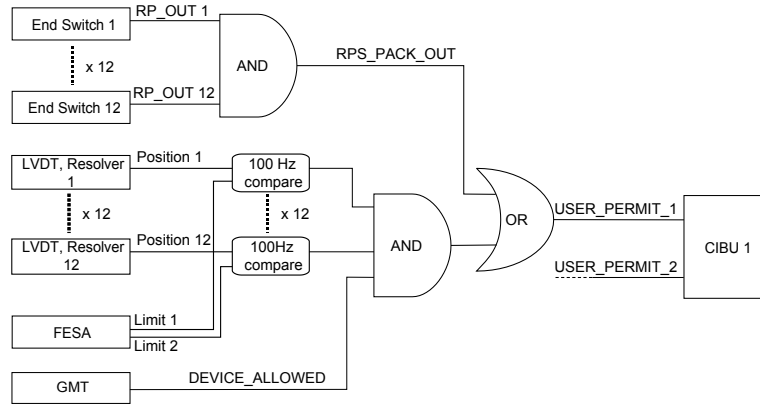


Figure 6.5: RP Position Interlock (CIBU 1)

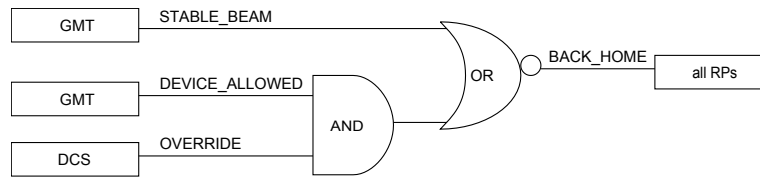


Figure 6.6: RP Retraction in UNSTABLE_BEAM Mode

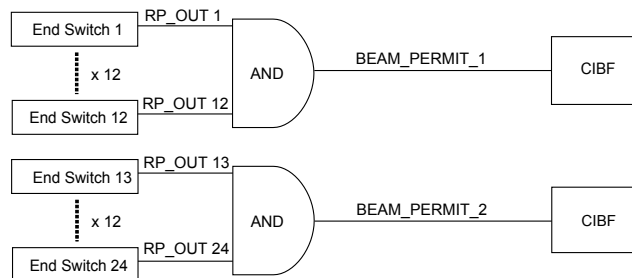


Figure 6.7: RP Injection Inhibit (CIBF)

Part III

Requirements and Solutions: Big.Brother

(This page is intentionally left blank)

Chapter 7

Thesis planning

7.1 Evolution of the thesis in relation with the TOTEM experiment

This thesis has evolved in the framework of the TOTEM experiment and the DCS project. However the objectives of the thesis and the experiment are different, but any progress in the thesis was beneficial for the developments and vice versa.

The DCS project uses Goal Directed Project Management (GDPM) as planning methodology. This methodology proposes a set of tools and principles for planning, organizing, leading and controlling projects. The method originated from PSO (People, System, Organization) projects in the IT domain. The method encourages a team oriented approach towards planning and controlling projects.

In Figure 7.1 a GDPM representation of the DCS activity plan is given. Each column in color of the table represents a different kind of activity:

- Project Management
- Hardware
- procurement of Requirements
- Development and unit testing
- Integration
- Commissioning

For each activity a list of milestones are represented in the form of bubbles. The milestones table has to be read from the top to the bottom. Linked milestones mean that the following one cannot be achieved before the previous has been completed; in other words, the finalization of the first one is necessary in order to complete the following linked milestone. Each milestone is decomposed in a detailed Activity Plan directly accessible by clicking on the Milestone name. The activities of the DCS team are there finally grouped in Work Packages that correspond to the single piece of control that has to be developed, tested implemented and commissioned to guarantee the operation of the TOTEM experiment.

In this planning many of this thesis researches and tagged as Work Packages inside Milestones, and associated to me. In a first step the comparison among the CERN ‘given’ technologies and possible alternatives was done. At the same time the process of extracting requirements and formalizing requirements started.

The next step, given the problems of development with PVSS, was the improvement of those formalizations with emphasis on automatic processing.

The creation of the connectivity tables and scripts lead to a reduction of the development times by a factor of 3, moreover it is possible to ensure that what is being produced is what has been agreed in the documents. So this now opens the doors to formal methods of analysis for the requirements.

The calculations of Chapter 13, *Information theory analysis* made possible to have estimations on the DCS needs in terms of computing resources. It was possible to make an agreement about the archiving with the DAC and CMS-DCS groups. Also it became possible to confirm the number of ELMB, number of processing nodes and so on, instead of having an estimation based on the experience of colleagues from other LHC experiments.

Also many other discussions were done at the same time, such as the formalization of the Roman Pots operational procedure, and the location of pieces of equipment.

All this points corresponds to the phase of design, where most of the researches takes place. The development is currently ongoing, when a features is being developed they are addressed as new Work Packages in the GDPM plan. In these new developments, the automatic generations tools are extended and improved to fulfill the new objectives.

Having a ‘bare-bone’ DCS prototype lead to the CMS integration issues. CMS experiment needs to have all the PVSS development packaged as PVSS JCOP component. An automatic installation procedure generates an empty project in the target computer and install the proper components. The usual practice in all the other LHC experiments is to install full projects and doing full backups of them.

Due to my requests, CMS-DCS, has implemented the proper tools to show and reinstall the versions of our developments. This opens the doors to the Configuration Managements Plan. For such a complicated system as the Roman Pots it is mandatory to track and record all the versions and states of the system.

During the LHC working days of 2008, due to a huge effort the radiation sensors were set up in place. It was critical to have the base line of the radiation sensors and the cabling ‘noise’ before any LHC operation. Unfortunately the bad soldering incident [CER08d] did not allowed to continue the tests with the others sensors.

As a final comment, GDPM has resulted to be extremely useful for the development of the DCS and the thesis. In a first approach it helps defining and prioritizing the objectives ¹, and secondarily it defines a timeline.

7.2 Feedback systems

7.2.1 Generic system

A simple form of feedback consists of two dynamical systems connected in a closed loop which creates an interaction between the systems. Simple causal reasoning about such a system is difficult because

¹‘A good plan today is better than a perfect plan tomorrow.’

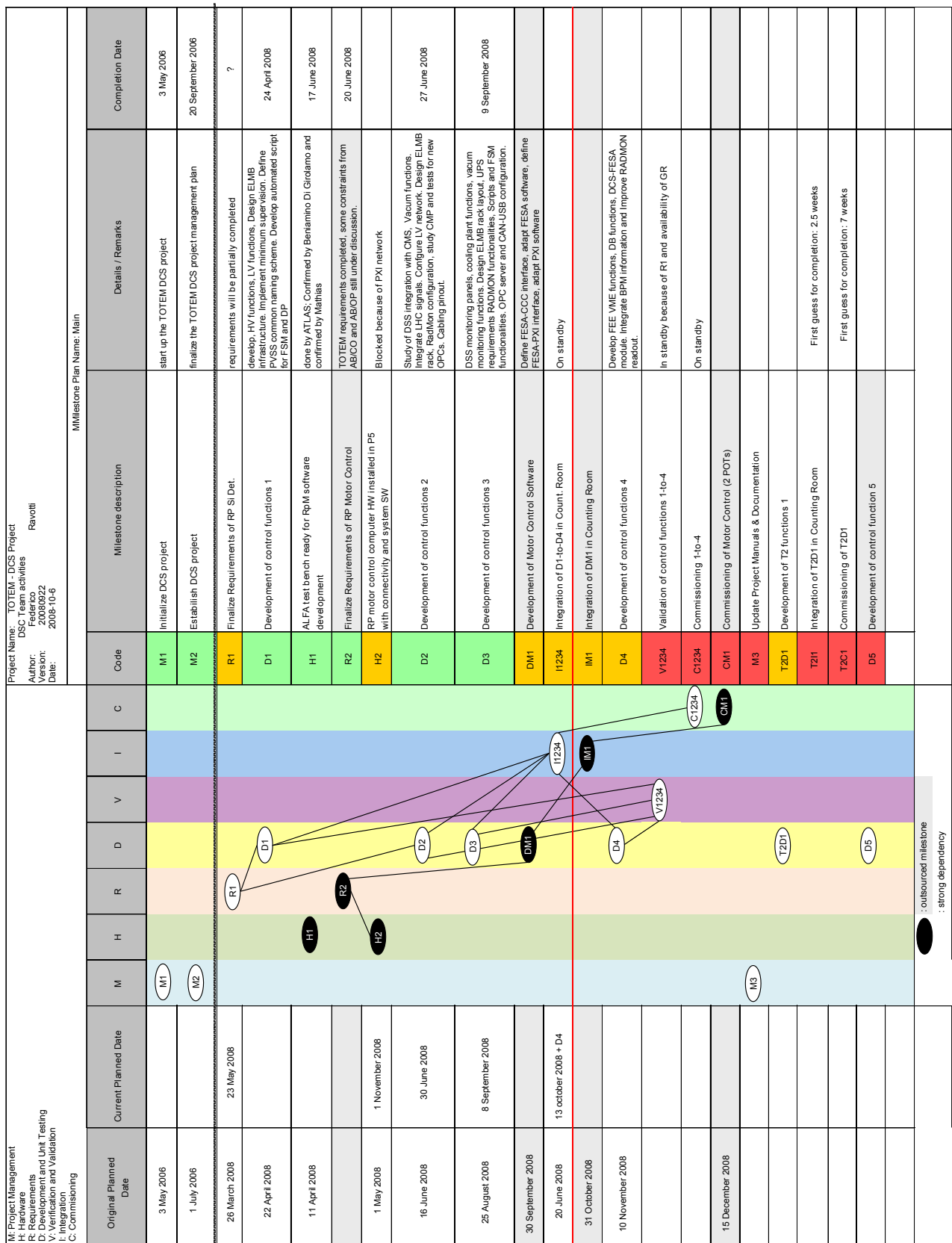


Figure 7.1: TOTEM DCS Milestone Plan

the first system influences the second and the second system influences the first, leading to a circular argument. This makes reasoning based on cause and effect difficult and it is necessary to analyze the system as a whole. A consequence of this is that the behaviour of a feedback system is often counterintuitive [Car05].

Feedback control means measuring the controlled variable (an output), comparing that measurement to the set point (desired value), and acting in response the error (difference between set point and controlled variable) by adjusting the manipulated variable (an input). The control algorithm is the set of calculations and decisions that lies between the error and the directions given to the final control element.

The function of a feedback control system is, therefore, to ensure that the closed loop system has desirable dynamic and steady-state response characteristics. Ideally, it should satisfy the following performance criteria:

1. The closed loop must be stable.
2. The effect of disturbances are minimized, providing good disturbance rejection.
3. Rapid, smooth responses to set-point changes are obtained, that is, good set-point tracking.
4. Steady-state error (offset) is eliminated.
5. Excessive control action is avoided.
6. The control system is robust, that is, is insensitive to changes in process conditions and to inaccuracies in the process model.

A generic representation of a system is shown in Figure 7.2. The system has two blocks; the ‘controlled plant’ represents the process and the ‘control system’ represents the controller.

In practice there may be many different disturbances that enter the system in many different ways. Measurement noise corrupts the information about the process variable obtained from the measurements [Joh03]. The major drawback is that feedback can create instability.

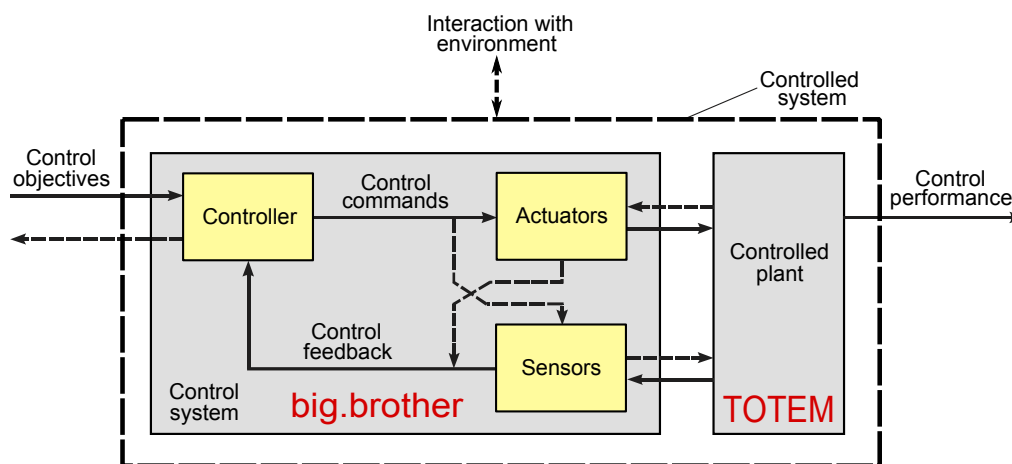


Figure 7.2: Generic control structure with error feedback; adapted from [ECS04b]

7.2.2 System identification

System identification is the experimental approach to process modelling.

It includes the following steps [DTB97]:

- Experiment design:
Its purpose is to obtain good experimental data, and it includes the choice of the measured variables and of the character of the input signals.
- Selection of model structure:
A suitable model structure is chosen using prior knowledge and trial and error.
- Choice of the criterion to fit:
A suitable cost function is chosen, which reflects how well the model fits the experimental data.
- Parameter estimation:
An optimization problem is solved to obtain the numerical values of the model parameters.
- Model validation:
The model is tested in order to reveal any inadequacies.

The field of system identification starts from experimental test data, and develops a mathematical model of the observed physical system behavior. Typically one excites the system with some rich input signal such as a random white noise, measures the response at sample times, and from this data one creates a differential equation or a difference equation that can predict the system response to arbitrary inputs.

Unfortunately TOTEM experiment in its current development state is not able to provide any kind of model or approximation for the behaviour of the system. The current DCS development approach is to prove basic control and monitoring functionalities. Some critical interlocks will be programmed in the FSM logic for preventing damage of the detectors. However the ‘fine grain’ feedback loop will be responsibility of the operator until enough experience has been accumulated.

Once that there is a clear model of the response of each detector, the control system can be improved. These new requirements would need a few iterations in the development process to build a more automatic system.

7.3 Hardware overview

The three TOTEM detectors Roman Pots Silicon Detectors, T1 and T2 are detailed in Chapter 3, *The TOTEM experiment*. They require the usual equipment of High Energy Physics experiments, see Appendix C, *Hardware components specifications*. All the three use CAEN HV power supplies, Wiener Maraton LV units, Wiener VME crates, and environmental sensors connected to ELMB or read from the DCU through the DAQ readout chain. T1 and T2 are cooled with water loops from the same primary circuit than the CMS rack cooling, while the Roman pots Si detectors use a more sophisticated cooling plant. The T1 and T2 gas systems are subcontracted, including a PC running PVSS control software compliant with the standard LHC Gas Control System [BH07].

The Roman Pots motorization is derived from the LHC collimators, and so is the corresponding control system [RM08]. The front-end is based on National Instruments PXI hardware and Labview RealTime software, adapted to the selected sensors.

The hardware is represented in the connectivity diagrams of Appendix B, *Hardware overview diagrams*. In this fashion in a single piece of paper per detector we concentrate the location of the equipment, the number of pieces, the connectivity and the number of wires involved. All this equipment is scattered around many different locations detailed in Appendix A, *Locations*. An overview of the specifications and usage are given in Appendix C, *Hardware components specifications*.

7.4 Requirements formalization

7.4.1 Templates

The requirements were collected using a set of the templates inspired by ALICE. One of such document exists for each detector, and another are the hardware overview diagrams of Chapter B, *Hardware overview diagrams*.

Other kind of document, the summary tables of [Luc08e], are inspired on the diagrams by Leszek Ropelewski for T2, and the pinout table [DL08] and the FSM hierarchy table [AL08] are defined by me.

The full list of documents is shown in the document catalog of Figure 7.3.

	File name	TOTEM ID	EDMS ID	Description	Originally defined by	File format	Process phase	PBS scope	Representation	Status				
										TOTEM	RP Mot	RP 0	T1	T2
Project management	ProjectManagementPlan	TOTEM-DCS-PM-001	828889	Project management plan (PMP)	ECSS-M-00B	doc	all	TOTEM	text					
	CesNetwork	TOTEM-DCS-PM-002		Customer Supplier Network		.xls	all	TOTEM	text					
	PhasingAndPlanning	TOTEM-DCS-PM-007	896893	Totem DCS phasing and planning	IA	.xls	all	TOTEM	table					
	DocManagementAndCatalog	TOTEM-DCS-PM-009	896895	Totem DCS document catalog	PP	.xls	all	TOTEM	table					
Engineering	Sep	TOTEM-DCS-ENG-001	???	System engineering plan (SEP)		.doc	all	TOTEM	text					
	Cmp	TOTEM-DCS-ENG-002	???	Configuration Management Plan (CMP)	ECSS	.doc	all	TOTEM	text					
	TP	TOTEM-DCS-ENG-003	???	Technology plan (TP)		.doc	all	TOTEM	text					
	ReqTotemGen	TOTEM-DCS-ENG-011	896896	TOTEM general – Functional and technical requirements	ALICE, mod by PP	.doc	requirements	TOTEM	text					
	ReqRpMotors	TOTEM-DCS-ENG-012	896897	Roman pots motors – Functional and technical requirements	ALICE, mod by PP	.doc	requirements	RpMec	text					
	ReqRpDetectors	TOTEM-DCS-ENG-013	896899	Roman pots silicon detectors – Functional and technical requirements	ALICE, mod by PP	.doc	requirements	RpDet	text					
	ReqT1	TOTEM-DCS-ENG-014	896900	T1 CSC – Functional and technical requirements	ALICE, mod by PP	.doc	requirements	T1	text					
	ReqT2	TOTEM-DCS-ENG-015	896902	T2 OEM – Functional and technical requirements	ALICE, mod by PP	.doc	requirements	T2	text					
	SummaryTables	TOTEM-DCS-ENG-016	896903	Summary tables	Leszek + ALICE	.xls	requirements	TOTEM	table					
	HwOverviewDiags	TOTEM-DCS-ENG-017	868711	Hardware overview diagrams	ALICE	.ppt	design and configuration	SubDet	diag					
	FsmHierarchy	TOTEM-DCS-ENG-018	896904	Finite state machines hierarchy	IA + FLR	.xls	design and configuration	SubDet	diag					
	PinoutTables			Connectivity between hw	FLR+EV	.xls	all							
	SensorsDistribution	TOTEM-DCS-ENG-040	896905	Sensors distribution	FLR	.xls	requirements	SubDet	table					
	EquipmentInventory	TOTEM-DCS-ENG-051	896906	Equipment inventory	ALICE	.xls	requirements	SubDet	table					
	ITResourcesAdministration	TOTEM-DCS-ENG-052	896907	IT resources	FLR	.doc	design and configuration	TOTEM	text					

Figure 7.3: Documents catalog; mainly for requirements

This table was the first attempt to show the status of the development. It was later replaced by the packaging state and the GDPM planning.

7.4.2 Detector Product Breakdown Structure (PBS)

A Product Breakdown Structure is a hierarchical decomposition. It is structured using nested levels that conforms the main system. This decomposition can be applied to each detector or to the DCS itself resulting different PBS trees.

The Roman Pot System develops on a hierarchical structure of eight levels as in Figure 7.4 which go from the whole roman pot system ('system') to its ultimate granularity ('strip').

The mirror symmetry with respect to the IP5 has been used to define the names at the different levels. In particular, for the sector is repeated the same convention used by the LHC hardware scheme ('sector 45' and 'sector 56'), for the station and the unit is used the distance from IP5 ('station 147', 'station 220' and 'unit near', 'unit far'). The pot name is derived from its position with respect to the beam axis ('pot top', 'pot horizontal',...).

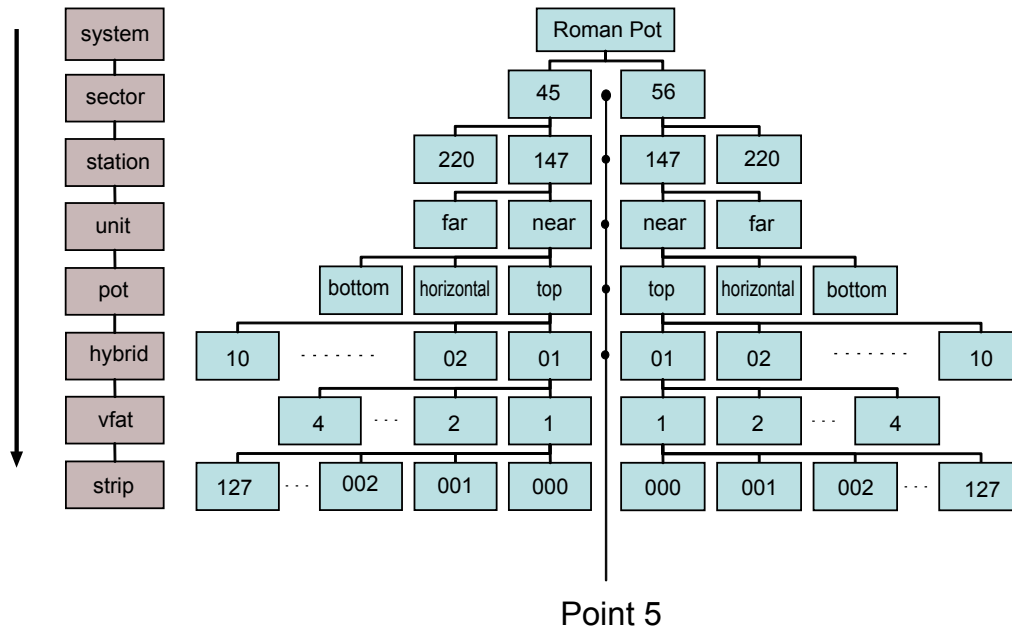


Figure 7.4: RP Naming Scheme

7.4.3 Naming Scheme

A clear naming schema is of vital importance in this kind of systems. Many sensors are installed each one of them needs a name as any of the devices there are mounted on. If this scheme does not exist each group uses different conventions, names or ordering. Resulting in systems that cannot be interfaced directly and very difficult to develop and maintain.

Figure 7.4 is an extract of [Rug08] as part of the requirements needed for building the DCS.

The naming of each subsystem of the Roman Pot detector can be derived by adding up the information of all the hierarchy levels in which the part is contained, following the order given by the arrow in 7.4 and abbreviating where possible to two letters (the first two consonants).

For example the 4th VFAT in the 2nd Hybrid of the top Pot in the far Unit of the Station at 147m of the sector 45 is *rp_45_147_fr_tp_02_004*.

Following this set of examples defined by Gennaro Ruggiero in [Rug08] it is possible to build a BNF grammar for the nomenclature:

```

<TotemId> ::= tot
| tot_ <RomanPotId>
| tot_ <Telescope1Id>
| tot_ <Telescope2Id>
| tot_ <GeneralId>

<RomanPotId> ::= rp
| rp_ <RomanPotSectorId>

<RomanPotSectorId> ::= 45
| 45_ <RomanPotStationId>
| 56
| 56_ <RomanPotStationId>

```

```

<RomanPotStationId> ::= 147
| 147 _ <RomanPotUnitId>
| 147 _ <RomanPotCoolingLoopId>
| 220
| 220 _ <RomanPotUnitId>
| 220 _ <RomanPotCoolingLoopId>

<RomanPotItemsId> ::= CoolingLoop

<RomanPotUnitId> ::= nr
| nr _ <RomanPotPotId>
| fr
| fr _ <RomanPotPotId>

<RomanPotPotId> ::= bt
| bt _ <RomanPotItemsId>
| tp
| tp _ <RomanPotItemsId>
| hr
| hr _ <RomanPotItemsId>
| <RomanPotLvChannel>

<RomanPotItemsId> ::= <RomanPotHvChannel>
| <RomanPotLvChannel>
| <RomanPotItemsSensorsId>
| <RomanPotItemsMotorizationId>
| <RomanPotPlanesId>

<RomanPotHvChannel> ::= Hv

<RomanPotLvChannel> ::= LvA
| LvD
| LvG

<RomanPotItemsSensorsId> ::= Temp01
| DssTemp
| Cool01
| Cool02
| Cool03
| Cool04
| Vacc01
| RadLaas
| RadCmrp
| RadRem
| RadBpw
| RadTemp
| RadRI

<RomanPotItemsSensorsId> ::= LvdT
| Reso
| MicrIn
| MicrOut

<RomanPotItemsMotorizationId> ::= Lv

<RomanPotPlanesId> ::= [01,02,03,04,05,06,07,08,09,10]
| [01,02,03,04,05,06,07,08,09,10] _ <RomanPotPlaneItemsId>

<RomanPotPlaneItemsId> ::= <RomanPotVfatId>
| <RomanPotPlaneDcu>

<RomanPotPlaneDcu> ::= Temp01
| Volt01
| Curr01

```

$\langle RomanPotVfatId \rangle ::= [1,2,3,4]$
 $| [1,2,3,4] _ \langle RomanPotStripId \rangle$

$\langle RomanPotStripId \rangle ::= [000,...,127]$

The names resulting are quite long, but some simplifications can be done in the User Interface of Chapter 15, *User interface*. In the FSM trees, PVSS internally uses the full names, but on the screen panels only the last level of the name is shown. The operator still have the hierarchy information inspecting the TreeView control.

7.4.4 Ordering

In some context it is also necessary to define an ordering of the elements converting them in a ordered set. An example of this need is when defining an ‘array’ with all the pots, as happens in the DIM communication with the motorization. Figure 7.5 from [Bau08] shows the agreed order.

The planes in a pot are counted starting from the one closest to the IP5 (‘plane 01’, ‘plane 02’,...). In each plane the VFAT are ordered with the 1st VFAT connected to the outer long strips of the silicon sensor and the 4th VFAT connected to the shorter strips.

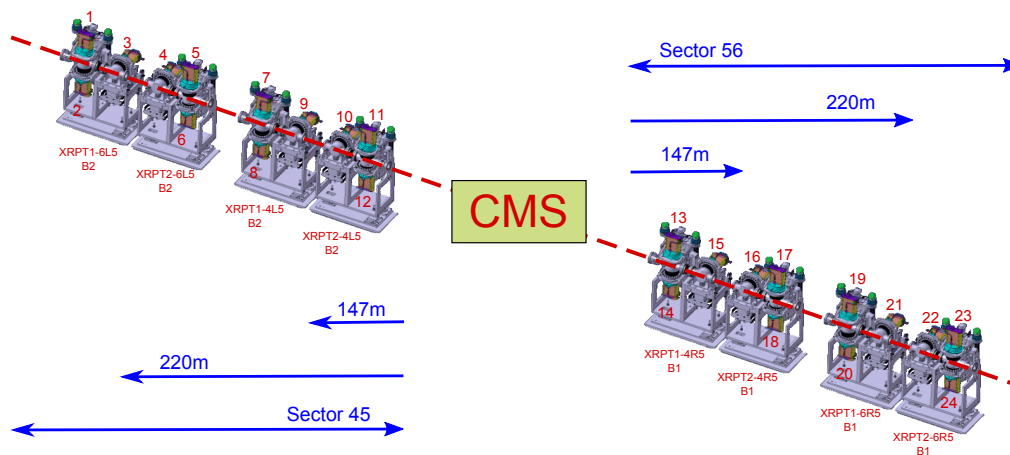


Figure 7.5: RP Order

7.4.5 Mapping

Table 7.1 shows a mapping among the different names that receive each Roman Pot. To simplify the operational tasks it is necessary to follow as much as possible the agreed naming convention, in the DCS development, electronics production, installation, but also take into account legacy names or the names given in external system such as the Collimation Application.

The importance of this mapping and trying to keep it as simple as possible is quite obvious when thinking of the operators dialog over the phone explained in Chapter 12, *Integration with the motorization*. If the TOTEM operator and the CCC operator have an screen with different names that cannot decipher, mistakes in this communication will be the daily routine.

CMS relative position	Roman Pot Product Breakdown Structure	Roman Pot order	RP mechanical design name	LHC layout name	Collimator application name	Roman Pot position	Left up	Right up	Collimator ID
	EDMS 906715	EDMS 901060	EDMS 934341	EDMS 934341	EDMS 934341	EDMS 934341			
Left	tot_Rp_45_220_fr_hr	3	Station 3	XRPT1.6L5.B2	XRPTOT04	Horizontal	X		0
	tot_Rp_45_220_fr_tp	1			XRPTOT05	Vertical up		X	1
	tot_Rp_45_220_fr_bt	2			XRPTOT06	Vertical down		X	
	tot_Rp_45_220_nr_hr	4		XRPT2.6L5.B2	XRPTOT10	Horizontal	X		2
	tot_Rp_45_220_nr_tp	5			XRPTOT07	Vertical up		X	3
	tot_Rp_45_220_nr_bt	6			XRPTOT11	Vertical down		X	
	tot_Rp_45_147_fr_hr	9	Station 1	XRPT1.4L5.B2	XRPTOT12	Horizontal	X		4
	tot_Rp_45_147_fr_tp	7			XRPTOT14	Vertical up		X	5
	tot_Rp_45_147_fr_bt	8			XRPTOT13	Vertical down		X	
	tot_Rp_45_147_nr_hr	10		XRPT2.4L5.B2	XRPTOT15	Horizontal	X		6
	tot_Rp_45_147_nr_tp	11			XRPTOT09	Vertical up		X	7
	tot_Rp_45_147_nr_bt	12			XRPTOT08	Vertical down		X	
Right	tot_Rp_56_147_nr_hr	15	Station 1	XRPT1.4R5.B1	XRPTOT25	Horizontal	X		8
	tot_Rp_56_147_nr_tp	13			XRPTOT27	Vertical up		X	9
	tot_Rp_56_147_nr_bt	14			XRPTOT26	Vertical down		X	
	tot_Rp_56_147_fr_hr	16		XRPT2.4R5.B1	XRPTOT23	Horizontal	X		10
	tot_Rp_56_147_fr_tp	17			XRPTOT28	Vertical up		X	11
	tot_Rp_56_147_fr_bt	18			XRPTOT22	Vertical down		X	
	tot_Rp_56_220_nr_hr	21	Station 3	XRPT1.6R5.B1	XRPTOT24	Horizontal	X		12
	tot_Rp_56_220_nr_tp	19			XRPTOT17	Vertical up		X	13
	tot_Rp_56_220_nr_bt	20			XRPTOT20	Vertical down		X	
	tot_Rp_56_220_fr_hr	22		XRPT2.6R5.B1	XRPTOT16	Horizontal	X		14
	tot_Rp_56_220_fr_tp	23			XRPTOT19	Vertical up		X	15
	tot_Rp_56_220_fr_bt	24			XRPTOT18	Vertical down		X	

Table 7.1: Mapping among most of the RP alternative names

7.5 Development process

The remarks about the development process are given because of the lack of any established procedure for building such kind of system. Experience accumulated during the development of this thesis shows that usual software development cycles does not match well this kind of developments and environment.

Each new iteration of the cycle can have a huge impact on the initial requirements. And how the purpose of the software is basic research, the previous experience is very limited. Is is needed to provide correct releases as fast as possible to match the new cabling of fix operational logic. Each software release helps validating the initial requirements and assumptions, and clarifies the next development cycles.

The system consists of HW and SW components, therefore it required comprehensive systems engineering. As with any control system, the development process is iterative vertically between system engineering and lower assembly engineering, and iterative horizontally between **RE**quirements engineering, **DES**ign-configuration, **VER**ification-**VAL**idation, **ANA**lysis and **INTE**gration (Figure 7.6).

Each one of these cycles in the GDPM plan represents a milestone. The cycles does not need to include all the steps. In a early stage, before providing any useful product, Validation and Analysis

are meaningless. But as the development advances they become the most critical steps.

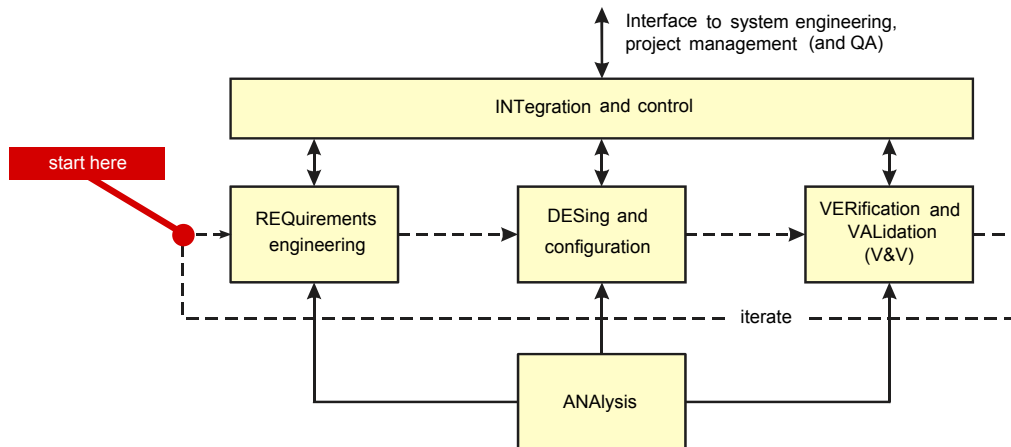


Figure 7.6: Horizontal development iteration

The CMS-DCS development model is federal, with independent detectors following their own engineering and project organization, so far that as they use the agreed technology and obey the mandatory integration rules. They are mainly a set of naming conventions and the definition of an integration node for Finite States Machines (FSM).

This situation is very different in ALICE, where the development model is centralized, and a number of engineering representations and software components have been developed centrally to be used by all detectors [Aug06]. Many of these templates have been adapted and improved as explained in Section 7.4.1, *Templates* and Appendix B, *Hardware overview diagrams*.

The way building a DCS by using the datapoints is divided in two parts. By one side there are PVSS datapoints generation for control/monitoring functions, and by the other the FSM logic is mapped in another set of PVSS datapoints. Both developments can be done in parallel after the agreement of the logic names. This has serious implications for the development of the automatizing scripts of Section 10.3, *Pinout tables and hardware generation scripts* and Section 11.3, *FSM hierarchy tables and operation logic generation script* and the packaging Section 7.6, *Packaging*.

Figure 7.7 shows the development sequence for developing new functionalities. There are four different kinds of blocks:

1. **Green blocks**

They express the requirements and the physical construction of the detector. Naming scheme, pinout tables, commissioning results (after development iterations), types of sensors and power channels conform this category

2. **Blue blocks**

They are the engineering formalization of all the requirements in a way that can be processed automatically. However, they do not attempt to be a 100% formalization of the requirements. The order of magnitude for hardware control functions or sensors (PVSS datapoints) can be near 4000 items, and the number of FSM nodes can be around 2500 items. Generate such a huge amount of items inside a PVSS project in a manual, or semiautomatic way is not good enough. The tedious JCOP procedure of manual generation of all those items can lead to human errors. Also this intermediary representation allows the physicist or any other provider of requirements to validate our development in a very early stage.

Other LHC experiments does not have such a level of automation, all the traceability of requirements into PVSS blocks has to be done manually. The verification of the products is

also done manually. The effectiveness of this new methodology does not leads doubts. An experienced PVSS developer in any other LHC experiment would need a month of work to update the requirements. Using this pioneer methodology the time from a change in the requirements until a proper PVSS component can be reduced to a pair of days.

3. Red blocks

Are the PVSS developments; datapoints, datapoint types, FSM types, scripts, panels,... Some of them are internal to TOTEM, but others are sent to CMS as packages for integration.

Configuration management is applied in all the steps of the DCS, but defining two major types of baselines:

1. DCS environment baseline

The one of pieces the DCS depend on (such as PVSS version, OPC servers, JCOP components,...).

2. DCS product baseline

The DCS development process output; the CMS-compatible components for integration (Table 7.2).

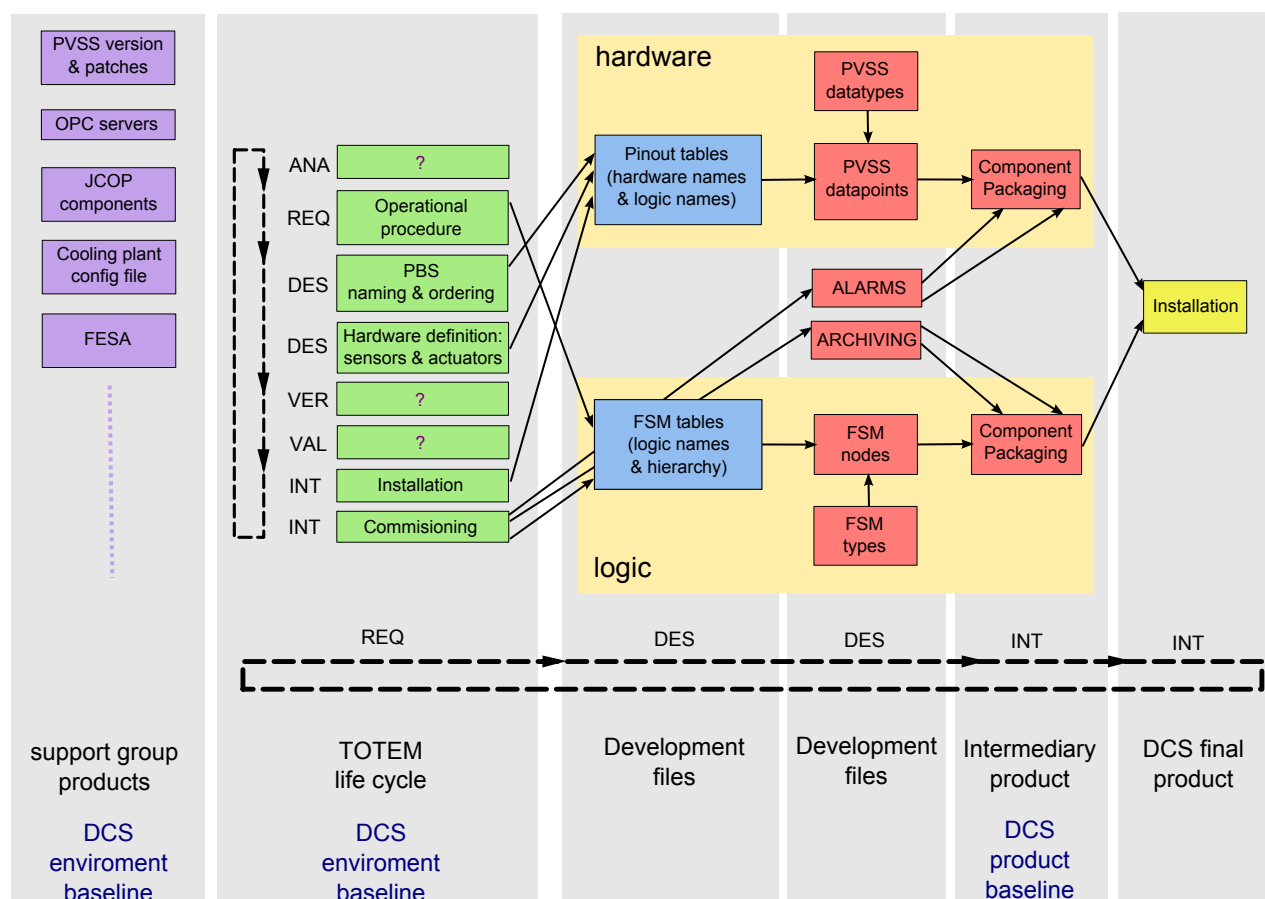


Figure 7.7: Development process for the TOTEM DCS

7.6 Packaging

CMS uses concepts of the JCOP framework and components to create its own CMS-DCS components. This DCS components does not replace the JCOP components but does use and complement it. All the DCS parts should be developed in the form of installable detector Framework components.

A software component is a system element offering a predefined service or event, and able to communicate with other components. Clemens Szyperski [Szy02] gives the following five criteria for what a software component shall be to fulfill the definition:

- Multiple-use
- Non-context-specific
- Composable with other components
- Encapsulated i.e., non-investigable through its interfaces
- A unit of independent deployment and versioning

JCOP components fulfill this criteria. They can be reused among many independent projects; if working together conforms the framework, but each one has its independent versioning.

In CMS control applications are created in the form of components. These components do have the format of JCOP framework components. Creating a control application in this format allows installing this application in any project already existing by means of the JCOP framework installation tool. Version tracking is also gained using this system. Applications can be moved from one project to another redistributing the load among the production systems.

This component system allows for easy installation of the controls, allows for easy update while keeping track of changes made and allows also to bring back an application to an older working version if needed. A central and tidy installation is possible using this system ensuring coherency (there are not several copies of same file) and maintainability (there is only one place to go to fix a bug) across the distributed system.

The aggregation of functionalities into components follows the the DCS PBS structure. There are several hardware configuration components: the HV of each detector is one, the LV another, for the sensors another,... The FSM logic of each detector is also an independent component, but this one is on specific versions of the hardware configuration components.

Table 7.2 show how the TOTEM DCS components conforms the final product. Some components are needed in all the processing nodes, other can be shared meanwhile others are exclusive.

If a detailed table of this table is colored in a traffic-light way, it is a quite convenient way of expressing the progress by complementing the GDPM plan. GDPM organizes present and future actions, but lacks a overall representation of the final product, this table completes that aspect.

Package name	NODE 1	NODE 2	NODE 3	NODE 4	NODE 5	LAB
totFsmTypes	X	X	X	X	X	X
totServices	X	X	X	X	X	X
totRadmon	X	X	X	X	X	X
totGeHv	X	X	X	X	X	X
totGeLv	X					
totGeEnvironment	X					
totGeCountingRoom	X					
totRpHv		X				
totRpLv		X				
totRpEnvironment		X				
totRpLv		X				
totRpEnvironment		X				
totRpCooling		X				
totRpSv		X				
totTtHv			X			
totTtLv			X			
totTtEnvironment			X			
totTtSv			X			
totT2Hv				X		
totT2Lv				X		
totT2Environment				X		
totT2Sv				X		
totUserInterface	X					
totSupervisor	X					

Table 7.2: Components distribution into processing nodes

Chapter 8

Comparison with commercial solutions

8.1 Levels

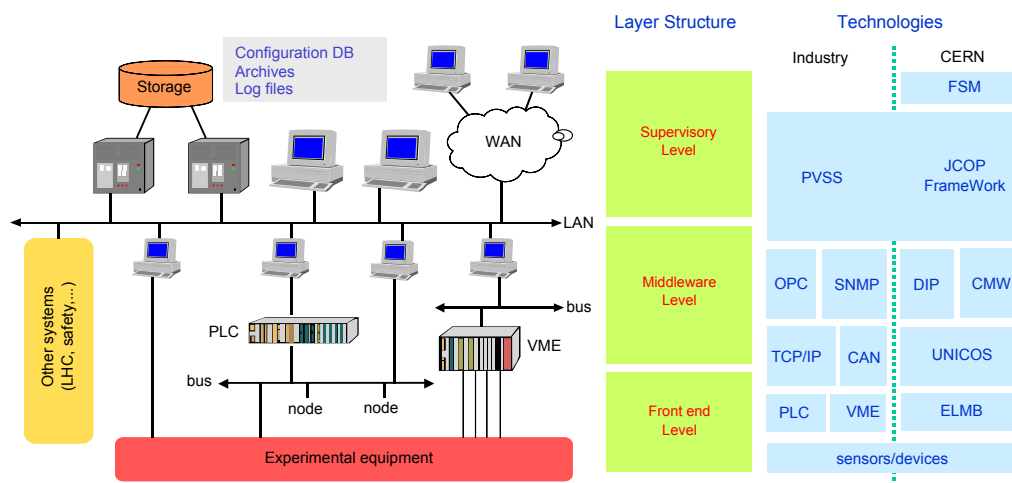


Figure 8.1: DCS levels; based on [Bur08]

Figure 8.1 represents the subsystems related to an abstract Control System in a 3 layer structure. At CERN many of these subsystems follow standards or well known industry technologies; but some of them are specific developments that suits better some problem domains.

The layers are:

- **Front end level**
The role of this level is to communicate the hardware in a more homogeneous way with the upper levels. It is necessary to define a physical level and an application level. The ELMB is an specific development but it uses ‘industry standards’ as CAN, CANopen and OPC.
- **Middleware level**
This levels takes the responsibility to make the information accessible to several computers or

many clients. Distributed software technologies are used here. In industry side OPC and SNMP are the usual technologies, meanwhile at CERN they are DIM, DIP and CMW.

- **Supervisory level**

This is the level that takes responsibility of the actions. Usually an SCADA software takes this role. The commercial product PVSS, needs to be extended with the JCOP framework. In industry, the operation logic usually is not formalized using FSM or some other way. This formalization permits some formal analysis and estimate response times as done in Section 13.7, *Analysis of the response time*.

8.2 Front end level

8.2.1 Physical Level

This section presents a comparison among many of the possible options in the physical connectivity for the front end systems. They include the readout of the sensors and the activation of the actuators. As can be seen in Figure 8.2 and Table 8.1 some of them are networks, other are fieldbuses and even another backplane buses.

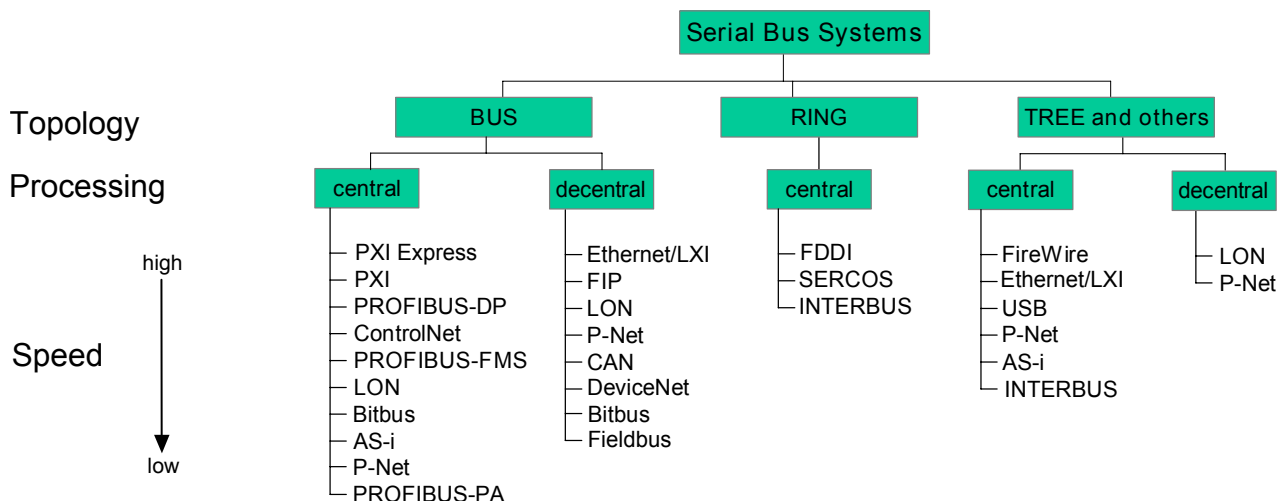


Figure 8.2: Serial buses

Three new promising instrument control buses are:

- **USB-controlled devices**

They are well-suited for applications with portable measurements, laptop or desktop data logging, and in-vehicle data acquisition; however, cable length is limited to 5 m and the connectors are not rugged.

- **Ethernet/LXI**

It is very useful when creating a network of highly distributed instruments that require remote access capabilities across large geographies. But Ethernet/LXI has the highest (worst) latency of all instrument control bus interfaces.

- **PXI Express**

Provides a high-bandwidth, low-latency interface for instrument control and data sharing in high-performance applications. However the cost is usually higher.

For the TOTEM DCS the choices have been PXI for the motorization control, Ethernet for the HV and LV power supplies control, and CAN for the VME crates control, and the readout of the sensors. It is

Technology	Master	Topology	max. Length	max. Speed	Wires	max. Stations	Standard
AS-i	single	bus, tree	100 m	167 Kb/s	2	32	EN 50295
BITBUS	multiple	bus	300 m at 75 Kb/s	375 Kb/s	2	251	IEEE 1118
CAN	multiple	bus	500 m at 125 Kb/s	1 Mb/s	2	64	ISO 11898 ISO 11519
ControlNet	multiple	bus, star, tree	5 km	5 Mb/s	coaxial	99	open specified
DeviceNet	multiple	bus	500 m at 125 Kb/s	500 Kb/s	4	64	open specified
Ethernet/LXI	multiple	bus, star, tree	100 m	1 Gb/s	4	4294967296	open specified
FDDI	multiple	ring	100 km	100 Mb/s	fiber	500	IEEE 9314
Fieldbus	multiple	bus	200 km	31.25 Kb/s	2	240	open specified
FIP	multiple	bus	200 km at 1 Mb/s	2.5 Mb/s	2	256	EN 50170
FireWire	multiple	tree	4.5 m	3.2 Gb/s	4 or 6	63	IEEE 1394
GPIO	single	bus	20 m	8 Mb/s	24	15	IEEE 488
INTERBUS	single	ring	12.8 km	500 Kb/s	2 or 8	255	EN 50253
LON	multiple	bus, tree	6.1 km at 5 Kb/s	1.2 Mb/s	2	2	ANSI
Modbus plus	multiple	bus	1.8 km	1 Mb/s	2	32	proprietary
P-Net	multiple	bus, tree	1.2 km	76.8 Kb/s	2	32 + 125	EN 50170
PROFIBUS FMS	multiple	bus	19 km at 9.6 Kb/s	500 Kb/s	2	127	EN 50170
PROFIBUS DP	multiple	bus	1 km at 12 Mb/s	12 Mb/s	2	127	EN 50170
PROFIBUS PA	single	bus	1.9 km	93.75 Kb/s	2	32	EN 50170
PXI	single	bus	10 m	1.06 Gb/s	124	4 multiplex.	open specified
PXI Express	single	bus	10 m	2.5 Gb/s	96	4 multiplex.	open specified
SERCOS	single	ring	250 m	16 Mb/s	2 or fiber	245	IEC 61491
USB	single	bus	5 m	480 Mb/s	5	127	open specified

Table 8.1: Communication technologies comparison

also quite common that the LV and HV power supplies manufacturers use CAN for the control, and the LHC machine also uses FIP for their sensors.

The idea of reusing the ELMB development of ATLAS experiment, see Section C.4, *ELMB (Embedded Local Monitoring Boards)*, had the implication of using CAN as associated fieldbus, suppressing any discussion about this topic. The advantages of the ELMB are that they provide a well tested solution up to certain radiation levels, magnetic fields, electromagnetic compatibility,... However as any *ad-hoc* development there are procurement problems.

8.2.2 Application Level

It is also necessary to standardize the application level of the communication between the sensor readout or actuator and a proper middleware, not only the physical link. In this way firmware and drivers can be reused among devices.

An example of the industry efforts in this area is the Standard Commands for Programmable Instruments (SCPI) [SCP01]. It defines a standard set of commands to control programmable test and measurement devices in instrumentation systems; but not the physical communications link, such as IEEE-488, RS232, USB, VXIbus,... It can be considered as an evolution of IEEE 488.2 (GPIO).

The SCPI commands are ASCII textual strings with an specific command structure and syntax, that are sent to the instrument over the physical layer. SCPI also includes standard command sets for several classes of instruments, e.g. power supplies, loads, and measurement devices such as voltmeters and oscilloscopes.

Two examples are:

- `SYSTem:COMMunicate:SERIal:BAUD 2400`
This command sets the baudrate of an RS232 interface to 2400 bit/s.
- `SYSTem:COMMunicate:SERIal:BAUD?`
This command queries the current baudrate of an RS232 interface.

The associated application protocol for the CAN bus is CANopen. The ELMB board uses it and is detailed in Section C.3.2, *CANopen*.

8.3 Middleware level

8.3.1 OPC Unified Architecture (OPC-UA)

The original coupling of OPC to COM/DCOM has many drawbacks:

1. Only available for the Windows Operating System
2. Frequent configuration issues with DCOM
3. No configurable time-outs
4. Security not really applicable
5. No control over DCOM (COM/DCOM is kind of a black box, developers have no access to sources and therefore have to live with bugs or insufficient implementations)

For those reasons the OPC Consortium is developing a replacement named OPC Unified Architecture. OPC-UA embodies all the functionality of the existing OPC servers and expands on top of them.

OPC-UA has a dual nature [MLD09]: it is object-oriented and it is service oriented. The object-oriented nature of OPC-UA enables provides a common object management method to support complex and flexible data models. The service-oriented nature of OPC-UA allows for broader interoperability with other platforms, as well as for increased visibility and security.

All of the *Base Services* defined by OPC are abstract method descriptions which are protocol independent and found the basis for the whole OPC UA functionality. The transport layer puts these methods into a protocol, which means it serializes/deserializes the data and transmits it over the network. Currently there are two protocols specified for this purpose. One is a binary, high performance optimized TCP protocol and as second, a Webservice-based one.

The OPC information model is not just a hierarchy based on folders, items and properties any more, but a so-called Full Mesh Network based on nodes instead. These nodes can additionally transmit all varieties of meta information. It can have attributes for read access (DA, HDA), methods which can be called (Commands) and triggered events which can be transmitted (AE, DA DataChange). The nodes be used for the sensor/actuator data as well all other types of meta data.

To facilitate the adoption of the new standard and to reduce the barrier to entry, the OPC Foundation has developed the OPC-UA Software Development Kit (SDK). In addition, a series of binary adapters can be used to grant direct access to all legacy COM-based OPC servers from a new OPC-UA client.

8.3.2 Simple Network Management Protocol Version 3 (SNMPv3)

SNMPv3 is defined by RFC 3411 and RFC 3418. It provides important security features security and remote configuration enhancements.

In typical SNMP usage, there are a number of systems to be managed, and one or more systems managing them. A software component called an agent runs on each managed system and reports information via SNMP to the managing systems.

Essentially, SNMP agents expose management data on the managed systems as variables (such as free memory, system name, number of running processes, default route,...). But this new SNMP version also permits active management tasks, such as modifying and applying a new configuration. The managing system can retrieve the information through the GET, GETNEXT and GETBULK protocol operations or the agent will send data without being asked using TRAP or INFORM protocol operations. Management systems can also send configuration updates or controlling requests through the SET protocol operation to actively manage a system.

SNMP itself does not define which information (which variables) a managed system should offer. It uses an extensible design, where the available information is defined by Management Information Bases (MIBs). MIBs describe the structure of the management data of a device subsystem; they use a hierarchical namespace containing object identifiers (OID). Each OID identifies a variable that can be read or set via SNMP. MIBs use the notation defined by ASN.1.

The MIB hierarchy can be depicted as a tree with a nameless root, the levels of which are assigned by different organizations. The top-level MIB OIDs belong to different standards organizations, while lower-level object IDs are allocated by associated organizations.

The Wiener maraton LV power supplies in the Ethernet configuration are managed using SNMP. This manufacturer also provides an OPC server that does a binding between the OPC protocol and the SNMP built-in features of the wiener remote controller.

8.4 Supervisory level

8.4.1 LabVIEW

One alternative for the supervisory level under study was the well known LabVIEW product from National Instruments.

There are many resources and documentation available; and it is also officially supported by CERN. The drawback were that any of OPC server for HV, LV or ELMB was integrated into Labview. The development of the bindings, debugging and long term maintenance was a huge effort compared with the available work force at the early stage of the project; just me. In the other hand the impossibility of any reuse of developments from other experiment also played against this option. Also integration of alarms and archiving seemed problematic.

8.4.2 CS-Framework (Control System Framework)

CS is a multi-threaded, event driven, object oriented and distributed framework with SCADA functionality [Hel08a]. It was developed for some other High Energy experiments at the GSI centre [Hel08b]. A control system can be developed by combining the CS framework with experiment

specific add-ons. This framework aims at control systems with up to 10000 process variables. Details on the implementation of the CS framework are described in [BBB⁺04].

LabVIEW has been chosen as the development tool for the CS framework. An object-oriented approach is used to provide flexibility and maintainability. Each hardware device or software module is represented by its own object. The objects do not communicate via direct method calls but via events. This has three advantages: First, one can easily replace an object by another one, since the receiver of an event is only defined by the name of the object. Second, one can replace one method by another method, since methods are identified by the name of an event. Third, if an object should not be created on the local host but on a remote PC, one just sends the necessary events to the remote node. SCADA functionality like trending and alarming is provided by the Datalogging & Supervisory Control (DSC) module of LabVIEW. However, the functionality of the DSC module is encapsulated in a dedicated class.

Although LabVIEW is not object-oriented, it is possible to use an object-oriented approach within LabVIEW. Objects are represented by Virtual Instruments (VIs) and classes by so called VItemplates (VIT). During run time, multiple objects (each object is a separate VI) can be created dynamically from a class (represented by a VIT) by the VI-server functionality of LabVIEW. Inheritance is also possible, but not as straightforward as in C++.

However this framework as pain LabVIEW lacked of the proper OPC bindings needed to control the equipment. Concluding that any preexisting SCADA-like system should need adaptations, the decision of using PVSS as the others LHC experiments was straightforward. The JCOP project had already adapted and extended PVSS to the CERN requirements, and it was supported by many CERN groups.

8.5 Management aspects

8.5.1 Introduction

Usually in the contracts among industrial partners or the ones with public entities have some clauses about the certifications of the parts or some quality assurance. ISO 9001 is the most usual standard in there areas.

ISO 9001 is a generic quality-management-system standard. ‘Generic’ means that the same standards can be applied to any organization, large or small, whatever its product or service, in any sector of activity, and whether it is a business enterprise, a public administration, or a government department. ‘Management system’ refers to what the organization does to manage its processes or activities in order that the products or services that it produces meet the objectives it has set itself. ‘Management system standards’ provide the organization with a model to follow in setting up and operating the management system. ‘Quality management’ is what the organization does to ensure that its products or services satisfy the customer's quality requirements and comply with any regulations applicable to those products or services. ISO 9001 concerns the way an organization goes about its work, and is not directly concerned with the result of this work. ISO 9001 has to do with processes, not products (at least not directly).

In contrast, IEEE standards are highly specific. They are documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics to ensure that materials, products, processes, and services are for to their purpose.

IEEE standards can be used as tools to help with the process definition and documentation (e.g., work products) required in support of the process improvement of software and systems development

efforts. Many of the IEEE software engineering standards provide detailed procedural explanations, they offer section-by-section guidance on building the necessary support material and, most importantly, they provide best-practice guidance in support of process definition.

In this way IEEE standards, as CMMI and the ESA standards match ISO 9001 and expand it with more specific requirements and recommendations for a specific problem domain. Those additional requests and suggestions comes from the experience of the Working Groups that define the standards.

An specific standard for ‘Building Automation and Control Network’ is defined in [Org03b] and [Ass09].

8.5.2 Capability Maturity Model Integrated (CMMI)

The Software Engineering Institute (SEI) developed the Capability Maturity Model Integrated (CMMI) framework in 2002 [Sof02]. Portions of this framework are described as Generic Practices (GP) and are applicable to all CMMI process areas. Generic practices provide institutionalization to ensure that the processes associated with the process area will be effective, repeatable, and lasting. In a similar purpose, ISO 9001 requires institutionalization to ensure that all business processes will be effective, repeatable, and lasting. Table 8.2 provides a cross-reference of CMMI GP to ISO 9001 requirement clauses and shows good coverage for most of ISO 9001.

Abbreviation	CMMI generic goals and practices	ISO 9001 clauses
GG 1	Achieve Specific Goals	
GP 1.1	Perform base practices	6.2.1
GG 2	Institutionalize a Managed Process	
GP 2.1	Establish and maintain the plan for performing the processes	4.1, 4.2.2, 5.4.2, 7.1, 7.3, 7.5.1, 7.6, 8.1, 8.2.3
GP 2.2	Establish and maintain an organizational policy for planning and performing the processes	4.1, 4.2.1, 5.1, 5.5.3, 7.6, 8.2.2
GP 2.3	Provide adequate resources for performing the processes, developing the work products, and providing the services of the process	4.1, 6.1, 7.5.1
GP 2.4	Assign responsibility and authority for performing the process, developing the work products, and providing the services of the processes	5.5.1, 8.2.2
GP 2.5	Train the people performing or supporting the processes as needed	6.2.1
GP 2.6	Place designated work products of the processes under appropriate levels of configuration management	4.1, 4.2.3, 4.2.4, 7.3.7, 7.5.1
GP 2.7	Identify and involve the relevant stakeholders of the processes as planned	5.1, 7.2.3, 7.3.2, 7.3.4
GP 2.8	Monitor and control the processes against the plan for performing the process and take appropriate corrective action	4.1, 7.5.1, 7.6, 8.2.3
GP 2.9	Objectively evaluate adherence of the processes against its process description, standards, and procedures, and address noncompliance	4.1, 7.6, 8.2.2
GP 2.10	Review the activities, status, and results of the processes with higher-level management and resolve issues	5.6.1, 5.6.2, 5.6.3, 7.2.2, 7.3.2, 7.6
GG 3	Institutionalize a Defined Process	
GP 3.1	Establish a defined process	5.4.2, 7.1
GP 3.2	Collect improvement information	8.4

Table 8.2: CMMI generic practices and ISO 9001 cross reference; from [LW06a]

8.5.3 European Cooperation for Space Standardization (ECSS)

The aims of ECSS, see [ECS04a], have been to develop standards which improve industrial efficiency and competitiveness in the space industry. This has been achieved by replacing the multitude of different standards and requirements unique to each contractor or space agency, which existed in

the past, with one coherent set. They are expressed as a document template and a list of issues that must be taken into account. Until now 25 standards have been published and 30 more are under development.

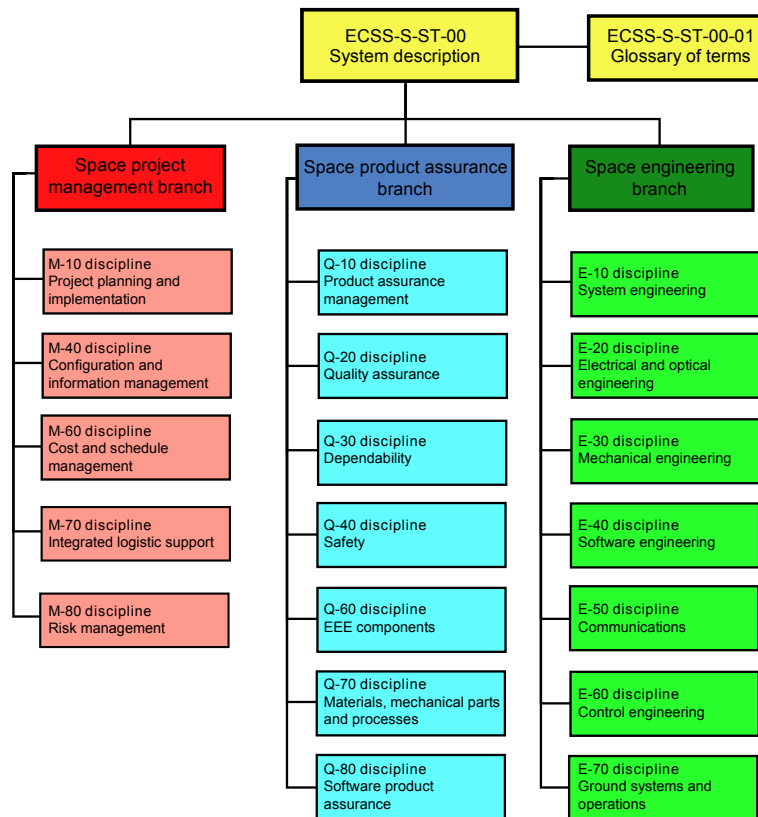


Figure 8.3: ECSS architecture

8.5.4 Development

The development process contains the activities and tasks of the developer. The developer manages the development process at the project level following the management process, infrastructure process, and tailoring process. Also, the developer manages the process at the organizational level following the improvement process and the training process. Finally the developer performs the supply process if it is the supplier of developed software products.

ISO 9001 identifies the requirements associated with development as ‘product realization’, [LW06b]. All products must be planned. That is, the development of processes associated with product realization should be documented and these processes should reflect the organization's actual method of operation. These requirements cover the processes, supporting information, and verification activities required in support of the development process. They are defined within 7 clauses:

1. Design and Development Planning
2. Design and Development Inputs
3. Design and Development Outputs
4. Design and Development Review
5. Design and Development Verification

6. Design and Development Validation
7. Control of Design and Development Changes

The development process, in conformance to ISO 9001, can be expanded using several IEEE standards:

- IEEE Std. 829, Standard for Software Test Documentation
- IEEE Std. 830, Recommended Practice for Software Requirements Specifications
- IEEE Std. 1008, Standard for Software Unit Testing
- IEEE Std. 1012, Standard for Software Verification and Validation Plans
- IEEE Std. 1016, Recommended Practice for Software Design Descriptions
- IEEE Std. 1063, Standard for Software User Documentation
- IEEE Std. 1074 Standard for Developing a Software Project Life Cycle Process
- IEEE Std. 1220, Standard for Application and Management of the Systems Engineering Process
- IEEE Std. 1233, Guide to Developing System Requirements Specifications
- IEEE Std. 1320.1, Standard for Functional Modeling Language –Syntax and Semantics for IDEF0
- IEEE Std. 1420.1, 1420.1a, and 1420.1b Software Reuse –Data Model for Reuse Library Interoperability
- IEEE Std. 1471, Recommended Practice for Architectural Description of Software Intensive Systems
- IEEE/EIA Std. 12207.0, Standard for Information Technology –Software Life Cycle Processes
- IEEE/EIA Std. 12207.1, Standard for Information Technology –Software Life Cycle Processes –Life cycle data

The development process is the largest of the 17 processes in IEEE 12207, and it is decomposed in the following activities:

1. System requirements analysis
2. System architectural design
3. Software requirements analysis
4. Software architectural design
5. Software detailed design
6. Software coding and testing
7. Software integration
8. Software qualification testing
9. System integration
10. System qualification testing
11. Software installation
12. Software acceptance support

As ISO 9001 is generic for all product and services and written at a higher level than the more detailed IEEE 12207, is reasonable that ISO 9001 has fewer subclauses than IEEE 12207.

Each one of those activities can be formalized using a document template to fulfil the objectives of ISO, IEEE, CMMI,... An example of this is given for the Configuration Management Plan in Table 8.3.

8.5.5 Software Configuration Management Plan (SCMP)

ISO 9001 fully supports the requirement to establish and maintain a plan for performing all configuration management process activities. ISO 9001 requires the documentation of project-level Configuration Management activities, but also requires the description of organizational Configuration Management activities. IEEE Std 828, IEEE Standard for Software Configuration Management Plans

Comparison with commercial solutions

(SCMP), can be used to help support this requirement. A table of contents derived from IEEE Std 828 to support the goals of ISO 9001 is given in Table 8.3.

It provides section-by-section guidance in support of the creation of a software configuration management plan (SCMP). The SCMP should be considered to be a living document and should change to reflect any process improvement activity. This guidance should be used to help define a software configuration management process and should reflect the actual processes and procedures of the implementing organization.

	Title Page
	Revision Page
	Table of Contents
1.0	Introduction
1.1	Purpose
1.2	Scope
1.3	Definitions/Acronyms
1.4	References
2.0	Software Configuration Management
2.1	SCM Organization
2.2	SCM Responsibilities
2.3	Relationship of SCM to the Software Process Life Cycle
2.3.1	Interfaces to Other Organizations on the Project
2.3.2	Other Project Organizations SCM Responsibilities
2.4	SCM Resources
3.0	Software Configuration Management Activities
3.1	Configuration Identification
3.1.1	Specification Identification
3.1.2	Change Control Form Identification
3.1.3	Project Baselines
3.1.4	Library
3.2	Configuration Control
3.2.1	Procedures for Changing Baselines
3.2.2	Procedures for Processing Change Requests and Approvals
3.2.3	Organizations Assigned Responsibilities for Change Control
3.2.4	Change Control Boards (CCBs)
3.2.5	Interfaces
3.2.6	Level of Control
3.2.7	Document Revisions
3.2.8	Automated Tools Used to Perform Change Control
3.3	Configuration Status Accounting
3.3.1	Storage, Handling and Release Of Project Media
3.3.2	Information and Control
3.3.3	Reporting
3.3.4	Release Process
3.3.5	Document Status Accounting
3.3.6	Change Management Status Accounting
3.4	Configuration Audits and Reviews
4.0	Configuration Management Milestones
5.0	Training
6.0	Subcontractor Vendor Support

Table 8.3: Software configuration management plan document outline; from [LW06b]

8.5.6 Conclusions

In the TOTEM DCS project the Project Management Plan (PMP) [Pal06] is derived from the ECSS template, as well as ideas for the cyclic development process. However for the ongoing SCMP, ECSS does not provides any proper template and IEEE Std 828 is the source of inspiration. The planning follows the GDPM methodology, see Section 7.1, *Evolution of the thesis in relation with the TOTEM experiment*, but any relation with the standards has not been studied.

Chapter 9

Frontend sensors

9.1 Sensors types

9.1.1 Temperature

The temperature sensor are usually used of detector safety interlocks. The HV does not consumes high currents, but the LV does. Temperature and the status of the electronic chips are usually quite related.

PT100

This is a very common temperature sensor. A PT100 is installed in every plane of the Roman Pots. They are also distributed inside the volume of T1 and T2. This sensor does not require calibration and is extensively used by other experiments. It will be used in the 4 wires configuration, and connected to the DCS and the DSS.

PT1000

They are required for measuring the cooling capillaries inside the pot. It will be used in the 2 wires configuration and does not need any calibration.

The usual ELMB adapter with a resistor of 1 MOhm is suited for a NTC. Using this adapter with a PT1000 requires to set up the ELMB in the range of 25 mV. This range is not the ideal due to the > 320 m long cables; signal losses and noise can affect quite significantly the readout signal. However it was necessary a new production of NTC/PT1000 adapters for TOTEM, see Section C.4.5, *Adapters*, and a modified version of this adapter was assembled. They were mounted with a resistor of 100 kOhm to fit better in the 100 mV ELMB ADC range.

9.1.2 Pressure

Vacuum

This sensor is required for the Roman Pots. Any other experiment at LHC does not have requirements for measuring vacuum so a selection process took place to find the best suited device. This process concluded in a sensor named *SM 5430* [Sil06] due to its radiation hardness and output signals compatible with the ELMB. This sensor follows a ‘Winston bridge’ configuration, as shown in Figure 9.1. Two wires provide a reference current of 5 V and other two wires provide a signal in mV depending on the pressure.

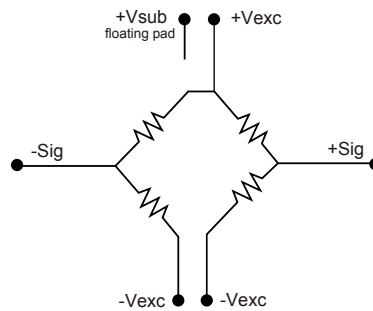


Figure 9.1: Vacuum sensor connectivity

Several tests took place in H8, see Section A.2, *Test area H8*, resulting in the calibration curve of Figure 9.2 [VVG08]. The output signal for a pressure of less than 10 mbar measures less than 25 mV. An ELMB configured in the default range of 100 mV has proved to provide enough resolution for the vacuum sensor readout. The range of 100 mV suits better the requirements than the 25 mV range. In this way the pressure can be monitored and recalibrated during maintenance operations; the pumps will be stopped, the pot opened and the vacuum will become ambient pressure.

One clear result of those tests is that calibration data is required for all the sensors. Each one of them has different offset values, but similar slope. Also the influence of the operating temperature of the pot (around -25°) has to be studied.

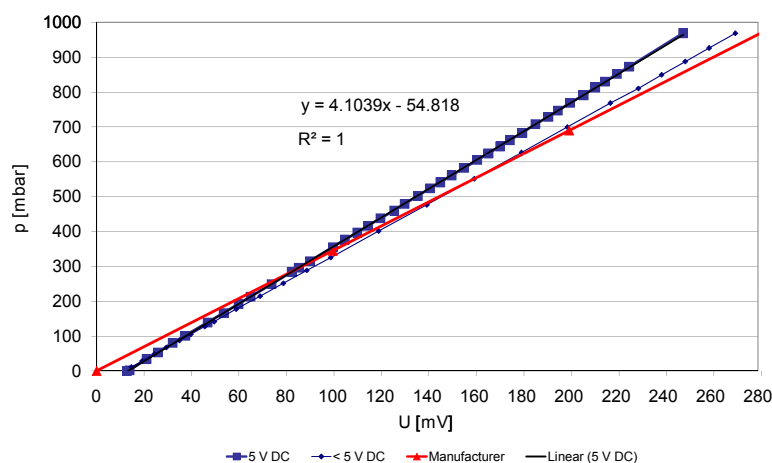


Figure 9.2: Vacuum sensor calibration curve

This sensor is responsible for some detector safety interlocks inside the pot, [ORL06]. At so low temperatures there is risk of condensation and ice. If this happens there will be shortcuts in

the electronics. When the vacuum goes over a threshold all the LV and HV channels must be disconnected. Also if a sudden vacuum lost happens during data taking, the pot must be retracted. The pot window could had been deformed too much because of the difference of pressure and touch the beam leading to a serious LHC incident.

Ambient Pressure

This sensor is required for T1 and T2. A sensor of the same family that the one for the RP vacuum will be used. This other device will have the same configuration and excitation signal, but an output of more mV.

In any case, an ELMB configured in the range of 100 mV can measure that output.

9.1.3 Humidity

This sensor is required for T2. An experimental sensor used by the CMS tracker will be used [STVW05] with an output signal in the order of 10 mV. The CMS tracker has also developed a special board to amplify the signal it into the [0,5] V range and also converting into 4 to 20 mA [Lop06].

Based on Figure 9.3, I have considered that the ELMB in the 25 mV, range is enough for the readout and no preamplification of the signal is necessary. Temperature corrections and calibrations are needed as in the pressure sensor.

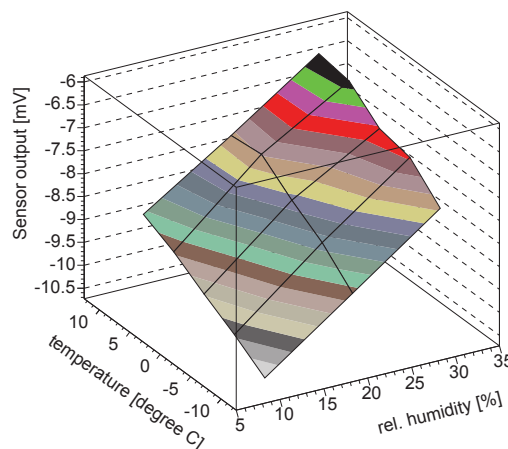


Figure 9.3: Typical calibration data of a humidity sensors

9.1.4 RADiation MONitors sensor

Concept

The RADiation MONitoring (RADMON) system aims to measure, the Total Ionising Dose (TID) and the 1 MeV equivalent particle fluence (Φ_{eq}) at various locations of the experiment itself.

Measurements of TID and Φ_{eq} are needed to understand the radiation-induced changes on the detector performances during TOTEM operation, to survey the radiation damage on electronic

components, to verify the Monte Carlo simulations in the detectors locations and to survey anomalous increases of radiation levels that may arise from accidental radiation burst such as beam losses or unstable beams.

The basic unit of the RADMON system is the *Integrated Sensor Carrier* (ISC) this is a small PCB hosting the sensors inside. The design of the RADMON readout electronics proposed here follows the ATLAS experiment where the sensors readout is based on Embedded Local Monitor Boards (ELMBs) [KCD*06]. This design has been adapted together by TOTEM together with the LHCb [CW07] and the ALICE experiments at the LHC.

A set of radiation monitoring sensors that fulfill the needs of the LHC experiments was selected, characterized and procured by the PH-DT2 and TS-LEA CERN groups. Details on these sensors, as well as calibration data, can be found in the ‘Sensor Catalogue’ [RGM05]. All the proposed sensors base their operating principle on the measurement of the voltage shift (ΔV) across the device's terminals under constant current biasing. Due to powering constraints of the DAC board, see Section 10.4.1, *The ELMB system connectivity*, the sensor's dynamic range will be referred to a maximum DV of 25 V.

Consequently two states are identified in these sensors:

- **Exposure mode:**
In this sensor both terminals of the sensor are grounded. No active measurement is done.
- **Readout mode:**
The sensor is biased to a power supply. There is a current circulating through it. The difference of voltage for a preset current determines the radiation level.

Dose sensors

TID measurements are performed with Radiation-sensing Field Effect Transistors (RadFETs) [HSRG07]. The recorded signal from these devices is the shift of the transistor threshold voltage (V_{th}) obtained at constant readout current. The V_{th} is converted to Dose (Gy) following a power-law calibration curve ($V_{th} = a \times \text{Dose}^b$) as shown in Figure 9.4 and Figure 9.5.

The sensors for the LHC experiments at CERN are of two types:

1. **Thick-Oxide RadFETs** (manufactured by CNRS-LAAS, France):
 - Range up to 10 Gy ($\Delta V < 10$ V)
 - Initial sensitivity of 500 mV/Gy (5 mV/rad)
 - Readout current of 100 μ A injected during 1 s
 - Initial $V_{th} \sim 2.5$ V
 - Accuracy of ± 10 %
2. **Thin-Oxide RadFETs** (produced by REM, UK):
 - Range up to 20 kGy [0.2 μ m] and 200 kGy [0.13 μ m] at $\Delta V = 25$ V
 - Devices with oxide thickness of 0.13 μ m were not specified in [RGM05] but were procured at CERN following a request of the ATLAS experiment
 - Initial sensitivity of 20 mV/Gy (0.2 mV/rad) [0.25 μ m] and 10 mV/Gy (0.1 mV/rad) [0.13 μ m]
 - Readout current of 160 μ A injected during 5 s
 - Initial $V_{th} \sim 3$ V
 - Accuracy of ± 10 %

Fluence sensors

Φ_{eq} measurements are performed with forward biased *p-i-n* diodes [RGM05]. The recorded signal from these devices is the shift of the diode forward voltage (V_F) readout under pulsed current injection.

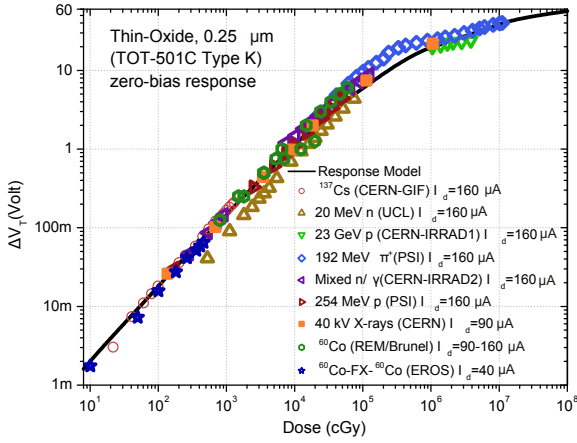


Figure 9.4: Thin-oxide (0.25 μm) RadFET calibration curve

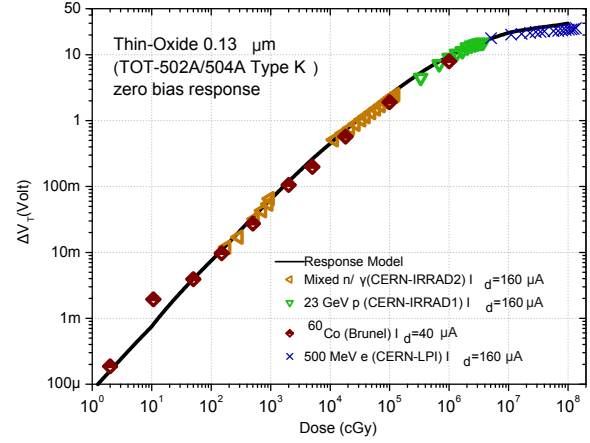


Figure 9.5: Thin-oxide (0.13 μm) RadFET calibration curve

The V_F is converted to equivalent particle fluence (cm^{-2}) following a linear relationship.

The sensors specified for the LHC experiments at CERN are of two types:

1. **High-Sensitivity Silicon Diode** (manufactured by CMRP, Australia):
 - Range up to $2 \times 10^{12} \text{cm}^{-2}$ ($\Delta V < 10 \text{ V}$)
 - Sensitivity of $1.7 \times 10^8 \text{cm}^2/\text{mV}$
 - Readout current of $1 \mu\text{A}$ injected during 50 ms
 - Initial $V_{th} \sim 3.5 \text{ V}$
 - Accuracy of $\pm 10 \%$
2. **BPW34F Silicon Diode** (manufactured by OSRAM, Germany):
 - Range from $2 \times 10^{12} \text{cm}^{-2}$ to $2 \times 10^{14} \text{cm}^{-2}$ at $\Delta V = 25 \text{ V}$
 - Sensitivity of $9.1 \times 10^9 \text{cm}^2/\text{mV}$
 - Readout current of $1 \mu\text{A}$ injected during 50 ms
 - Initial $V_{th} \sim 0.5 \text{ V}$
 - Accuracy of $\pm 20 \%$

Integrated Sensor Carrier (ISC)

The carrier (Figure 9.6) is made of a thin ($\sim 500 \mu\text{m}$) double-sided PCB [RGR⁺07]. A total of 11 devices, including a temperature sensor (10 kW NTC), can be mounted on the integrated sensor PCB with a common ground and be readout via a 12 wires flat cable [Rav06].

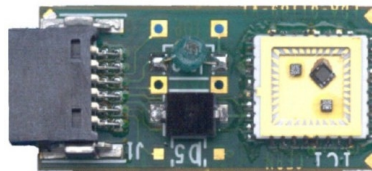


Figure 9.6: Integrated Sensor Carrier (ISC)

Several ISC configurations are possible; however two specific layouts had been evaluated for the TOTEM experiment:

1. **‘Base’ configuration:** 1 RadFET, 1 *p-i-n* diode, 1 NTC sensor. This configuration implies the use of 4-wires for the readout (3 sensors + RL).
2. **‘Redundancy’ configuration:** 2 RadFETs, 2 *p-i-n* diodes, 1 NTC sensor. This configuration implies the use of 6-wires for the readout (5 sensors + RL). With this ISC configuration, it will be possible to profit of the broad measurement range achievable combining the above mentioned four different RADMON sensors.

The redundancy configuration has been selected. The main reason was the broad range of TID and Φ_{eq} levels to be monitored. In this way it is possible to fulfill the radiation monitoring needs of TOTEM, see Section 3.7, *The TOTEM radiation environment*. This solution will be useful in order to increase the sensitivity during the low-luminosity LHC initial period.

In its final layout, in addition to the radiation sensors and the NTC, the ISC will mount a 1 kW reference resistor in one extra channel. The reference resistor will provide a reference value to check the connectivity of the ISC in its final location. Moreover this resistor can be used to check the line resistance during the commissioning on the system and in case of future replacement of the ISC itself.

Although the reference resistor signal will not be integrated directly in the Detector Control System, a total of **7-wires** (6 to the DCS + 1 reference) must be available in the counting room for the full readout of the elements of the ISC.

9.2 Measurement locations

The sensors in the TOTEM experiment will be scattered inside the detectors. This leads to the identification of **36 measurement locations**.

9.2.1 Roman Pots

Inside every pot there is need of monitoring the vacuum, temperature, and radiation. How all the sensors are internal to the pot, and as there are 24 pots, the result are 24 measurement locations for this detector.

The peculiarities of each sensor are:

- Two vacuum sensors will be mounted in each motherboard, both of them will be wired up to the RP motherboard external connectors but only one will be wired inside the patch panel because of the lack of enough wires in the long cables. This is not the ideal configuration. This sensor is not redundant, but during the first years of TOTEM operation vacuum of the 3 pots of one unit will be common. So there are 3 sensors in the same vacuum environment. Additionally, if there are problems with one specific sensor it can be rewired in the tunnel during maintenance operations.
- Although each one of the 10 hybrids has its own PT100, only two of them will be driven outside the motherboard. Usually they are the planes number 3 and 8, but this can be modified doing some manual soldering in the RP motherboard. Those sensors will be wired up to the ELMB boxes (see Section 10.5.4, *DCS rack interconnections*) and there, using discrete wiring, one is rerouted into the DSS and another into the ELMB. This allows that in case of a problem with a DSS temperature sensor the DCS one can be unplugged and rerouted to the DSS. All this is done inside USC55, where the access is normally allowed. The DCS will continue having the temperature readout through the DSS DIP publishing. Also 4 PT1000 are installed for the cooling capillaries monitoring inside the pot.

- Since the RADMON ISC is embedded in the motherboard layout, Figure 9.7, a special housing to protect the ISC from dust and mechanical damage is not required.

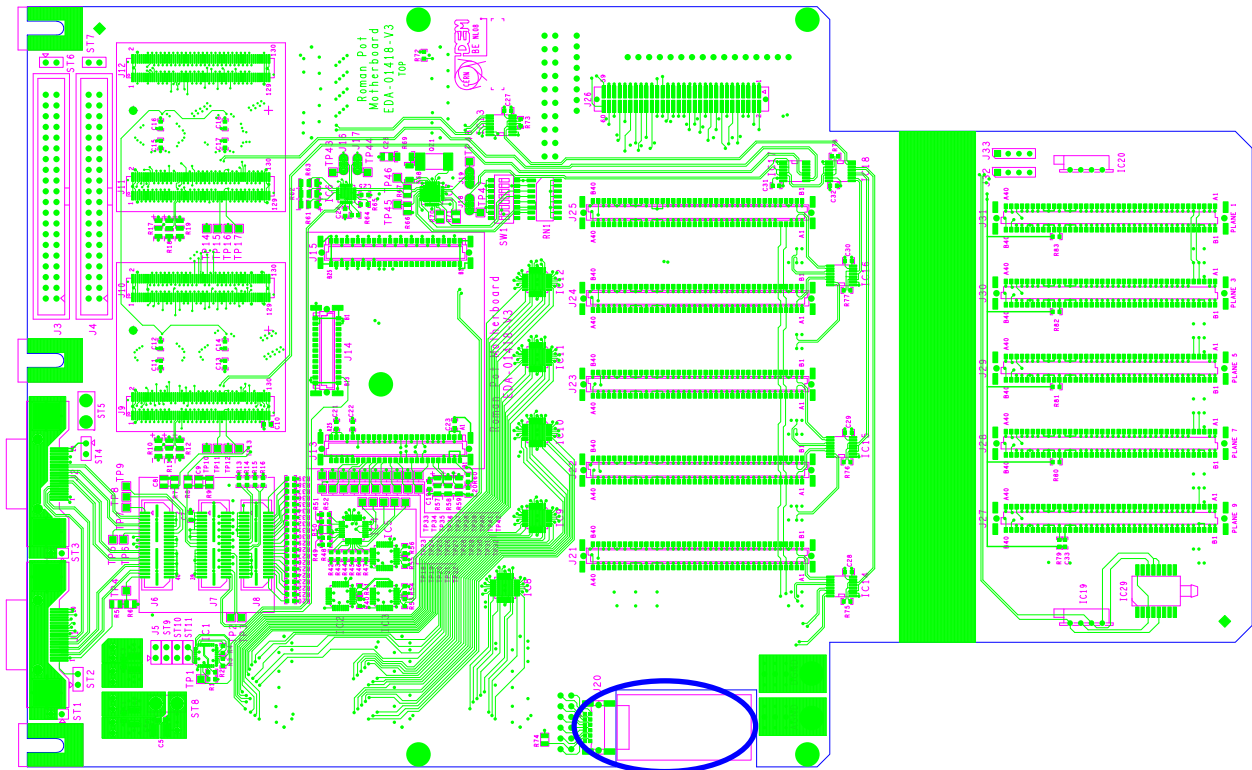


Figure 9.7: Totem RP Motherboard layout (CERN EDA-01418). The ISC is embedded in the top left-hand corner.

9.2.2 Telescope T1

The intention is to measure temperature, humidity, pressure and radiation. Due its big volume 2 RADMON sensors will be installed per quarter. So there will be 8 measurement locations for TI.

In these locations, the possibility to mount the ISC in a special housing to protect it from dust and mechanical damage may be considered.

9.2.3 Telescope T2

There is clear need to measure temperature, humidity, pressure and radiation. Each T2 quarter is quite compact (less than 2 m^3), so only 4 measurement locations are considered for the whole T2.

Most of the sensors are plugged into the '11th card', Figure 9.8. Ten 'horse shoes' boards are attached perpendicularly, so calling this board the '11th card' was quite straightforward. This card is one of the most complicated pieces of T2. The electronic design has more than 20 layers.

The peculiarities of each sensor are:

- The temperature sensors will be glued in some horseshoes as shown in Figure 9.9. The HV cooling and the Heat Exchanger will be measured.

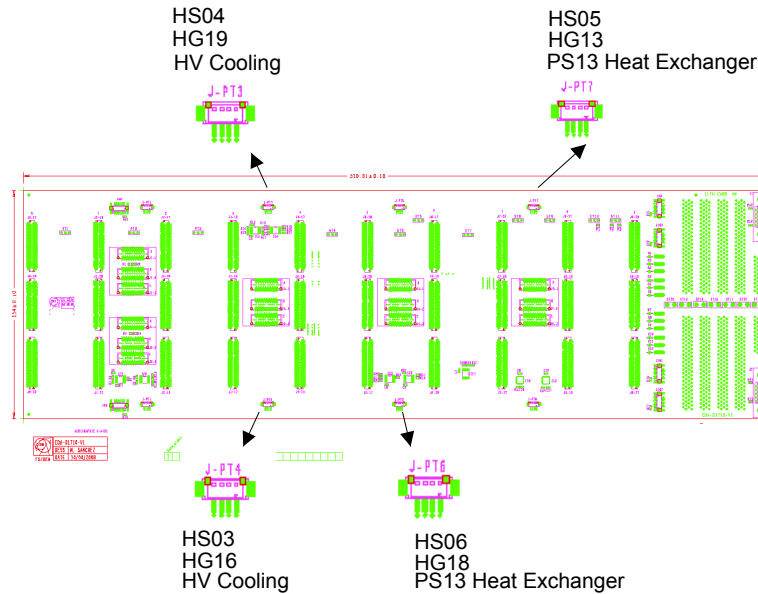


Figure 9.8: Connectivity of T2 sensors into the 11th card

- The humidity and pressure sensors need temperature calibration. For that reason all of them will be mounted in a ISC like board using the same connector type. Two of such boards will be installed in the '11th card' for the redundancy of the sensor type.
- Since the ISC is embedded in the '11th card', a special housing to protect the ISC from dust and mechanical damage is not required. The 11th card will have two connectors for the RADMON sensors, internally wired together, but only one will be used. This is done for avoiding mechanical collisions with the thick cables coming out of the 11th card; at least one of the positions should be free.

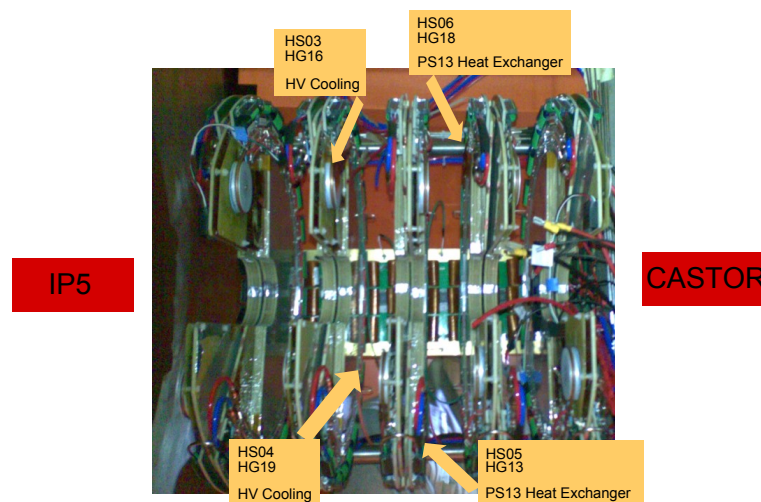


Figure 9.9: Location of temperature sensors in a T2 quarter

9.3 Distribution

Table 9.1, Table 9.2 and Table 9.3 section are an extract of [Luc08d]. They make a full description of all the wiring needs for DCS+DSS for every detector. They begin with the exact number of wires for each sensor types, and follow the PBS of each detector.

Detector PBS level		RADMON	Vacuum	Temp. PT100	Temp. PT1000	TOTAL	MAX.	FREE
Device	Wires	7	4	4	2			
Pot	Maximum number of installable devices	1	2	10	4			
	Devices connected to the DSS	0	0	1	0			
	Devices connected to the DCS	1	2	1	4			
	Wires per pot	7	8	8	8	31		
Station	Wires per station					186		
Detector	Devices connected to the DSS	0	0	24	0			
	Devices connected to the DCS	24	48	24	96			
	Wires in total	168	192	192	192	744		
	Wires rerouted to DSS	0	0	96	0	96		
	Wires to ELMB	168	96	96	192	552		
	ELMB allocated	4	1	1	2	8		

Table 9.1: Roman Pot Detector sensors distribution

Detector PBS level		RADMON	Temp. PT100	Pressure	Humidity	TOTAL	MAX.	FREE
Device	Wires	7	4	4	4			
Quarter	Maximum number of installable devices	2	10	2	2			
	Devices connected to the DSS	0	2	0	0			
	Devices connected to the DCS	2	8	2	2			
	Wires per quarter	14	40	8	8	70	108	38
Side	Wires per side					140	216	76
Detector	Devices connected to the DSS	0	8	0	0			
	Devices connected to the DCS	8	32	8	8			
	Wires in total	56	160	32	32	280	432	152
	Wires rerouted to DSS	0	32	0	0	32		
	Wires to ELMB	56	128	16	16	216		
	ELMB allocated	2	2	0.5	0.5	5		

Table 9.2: T1 Detector sensors distribution

Detector PBS level		RADMON	Temp. PT100	Pressure	Humidity	TOTAL	MAX.	FREE
Device	Wires	7	4	4	4			
Quarter	Maximum number of installable devices	2	10	2	2			
	Devices connected to the DSS	0	2	0	0			
	Devices connected to the DCS	1	8	2	2			
	Wires per quarter	7	40	8	8	63	108	45
Side	Wires per side					126	216	90
Detector	Devices connected to the DSS	0	8	0	0			
	Devices connected to the DCS	4	32	8	8			
	Wires in total	28	160	32	32	252	432	180
	Wires rerouted to DSS	0	32	0	0	32		
	Wires to ELMB	28	128	16	16	188		
	ELMB allocated	1	2	0.5	0.5	4		

Table 9.3: T2 Detector sensors distribution

As can be observed the sum is almost 1500 wires (744+280+252). This makes an idea of the complexity of the whole system in terms of connectivity and naming. Each wire needs to be plugged into an specific ADC channel, and given proper PVSS names (hardware and logical).

9.4 RADMON readout implementation

9.4.1 Readout of the radfets and p-i-n diodes

RadFETs and *p-i-n* diodes must be read out with currents ranging between 100 mA and 1 mA. These current will be injected in the sensors and the voltage drop will be measured between the contacts of the DAC. In other words, the sensor will switch form the *exposure mode* to the *readout mode* defined in Section 9.1.4, *RADIation MONitors sensor*. The range of the ELMB ADC is up to 4.5 V; therefore a differential voltage attenuator ($\sim 1:10$) is used for each DAC channel connected to a RADMON sensor. The attenuator circuit is hosted on a dedicated patch panel board. To reduce the noise on the ADC channel, capacitors are used in parallel to the resistors [KCD⁺06].

9.4.2 Readout of the temperature

One 10 kW NTC temperature sensor (103KT2125T) is mounted on each ISC. This sensor will be connected to the 2.5 V reference voltage output on the ELMB, in series with a 1 MW resistors located on the patch panel board. The corresponding voltage will be measured by one ELMB ADC channel.

9.4.3 Readout of the return line

In order to monitor the current flowing through the RADMON sensor, each Return Line (RL) is connected to ground over a 100 W resistor installed in the patch panel board. The small voltage drop on this resistor is measured by one ADC channel. When this voltage drop is converted into current, the small current (2.5 mA) that is continuously going through the NTC temperature sensor must be subtracted from the readout of the RadFETs and *p-i-n* diodes.

9.4.4 Constraints of the readout

During the *exposure mode*, the RADMON sensors must have the terminals connected to ground, but in the *readout mode* they must be biased to the current being generated by the DAC board. For this reason a series of switches (JFET transistors) are installed on the patch panel defined on Section 10.4.2, *RADMON Patch Panel board design*. Using DAC outputs they are able to ground or bias each RadFET or *p-i-n* diode.

An additional constrain is that the sensor cannot be powered during long intervals. The current circulating through the sensor affects quite a long the temperature of the sensor itself, making all the calibration curves inapplicable. Also if this powering continues too long it can lead to a physical damage of the sensor.

9.4.5 Use case

It is supposed that the ELMB are configured and initialized. Also all the power channels are properly set up. The readout procedure for reading three TOTEM ISCs connected to one patch panel:

1. PVSS commands the ELMB to set the switch array to the 'readout' position (sensors ungrounded). Using the digital output, the set of channels driving the switch array of the DAC board is set to provide 4000 μ A.
2. Loop over the three ISCs:
 - a) read ADC of the **temperature sensor (NTC)**
 - b) set the output current in the *p-i-n* diode 1 (High-sensitivity *p-i-n*) DAC channel to 1000 μ A
 - c) wait 500 ms
 - d) read ADC of the ***p-i-n* diode 1 (CMRP)**
 - e) read ADC for the **current control** on the return line (RL)
 - f) set the *p-i-n* diode 1 DAC channel to 0 A
 - g) set the output current in the *p-i-n* diode 2 (BPW34 *p-i-n*) DAC channel to 1000 μ A
 - h) wait 500 ms
 - i) read ADC of the ***p-i-n* diode 2 (BPW)**
 - j) read ADC for the **current control** on the RL
 - k) set the *p-i-n* diode 2 DAC channel to 0 A
 - l) set the output current in the RadFET 1 (Thick-Oxide FET) DAC channel to 100 μ A
 - m) wait 1 s
 - n) read ADC of the **RadFET 1 (LAAS)**
 - o) read ADC for the **current control** on the RL
 - p) set the RadFET 1 DAC channel to 0 A
 - q) set the output current in the RadFET 2 (Thin-Oxide FET) DAC channel to 160 μ A
 - r) wait 1 s
 - s) read ADC of the **RadFET 2 (REM)**

- t) read ADC for the **current control** on the RL
 - u) set the RadFET 2 DAC channel to 0 A
3. PVSS commands the ELMB to set the switch array to the 'exposure' position (sensors grounded). Using the digital output, the proper channel of the DAC board is set to its minimum current value (0 A).
 4. To calculate the 4 sensor currents, PVSS converts the voltage drop on the RL to current, subtract the 2.5 mA current that flows from the NTC biasing and display/store the currents injected in the sensor during readout. A check on this current can be done and, if its value differs more than 10% from the nominal one, a warning message can be displayed so the measurement is redone.
 5. PVSS applies the temperature and annealing corrections (when needed) to the voltages returned by the ADC from the different sensors (27 in total for the 3 ISCs). The ADC values have to be multiplied by 1.1 in order to obtain the real voltage from the RADMON sensors (to reflect the use of the attenuator). The Return Line and temperature voltages can be used as they are readout.
 6. PVSS converts the voltages to the dosimetric quantities and display them as function of the time of LHC run or any other useful 'timing' parameter from the machine.

Some additional intelligence can be applied if checking more limits. It is critical to ensure that the switches had been closed or opened properly. If a wrong state is detected with the current consumed in the return line, all the DAC gates should be closed and wrong measurements redone.

9.4.6 Readout rate

The RADMON system is a slow control system for the long-term monitoring of the radiation levels around the experiments and its electronics. This system is not meant to give feedback signals that may interlock or modify in real-time the operation of the experiment. However, the frequency of the readout sequence of the RADMON may be adapted to the needs of the experiment in the different operation phases of the machine and of the experiment itself.

Considering the layout of the ISC defined in the previous section the execution time of the commands and the data treatment by the ELMB system, see Section 14.1, *Commissioning of the RADMON-DAC*, each ISC needs at least 5 s for its readout. This calculation comes from the minimum of 3 s for the sensor biasing delay plus some experimental overhead due to the PVSS+ELMB processing.

The readout of the three ISC connected to the same patch panel board can be therefore estimated in 15 s. Therefore the practical minimum time between two consecutive measurements can be estimated in about 20 seconds.

Two DAC boards (each one with 1 patch panel) can be read in parallel, and also several ELMB can be read in parallel. *Therefore, with a proper algorithm, the radiation data from all measurement locations in TOTEM could be retrieved within 20 seconds.*

An important factor to consider is the execution delay of the SDO commands, which is dependent on the ADC rate of the ELMB. Table 9.4 shows this relation. Two SDO commands are needed to read each sensor; one for the current another for the voltage. Increasing the ADC speed allows to fulfill with more precision the biasing requirements of the sensor, but this can lead to noise problems due to the length of cables. That compromise to respect the biasing timing but avoiding possible electrical interferences restricts the ADC speed choices to 15.00 Hz or 7.51 Hz.

ADC rate	SDO delay
30.00 Hz	120 ms
15.00 Hz	225 ms
7.51 Hz	315 ms
3.76 Hz	600 ms

Table 9.4: ELMB ADC rate vs SDO command processing delay

9.5 Instrumentation problems

The > 320 m long cables seems to work properly during beam operation. No noticeable interference during the working days of 2008 has been detected. So the decision for installing the ELMB in USC55 seems a big success.

The ELMB are configured in the slowest possible ADC conversion speed (1.88 Hz instead of the usual 15.00 Hz) for being less sensitive to the noise.

LHCf also uses cables of similar length and they also confirms the absence of any significant problems.

(This page is intentionally left blank)

Chapter 10

Connectivity

10.1 Introduction

When trying to solve the problem of the connectivity between the sensor and the readout electronic several alternative comes to the mind. One of them are wireless technologies, this is an active research area and eliminates most of the cabling problems. The cabling cost is in the order of the most expensive parts of equipment. However this approach is not valid, the LHC and the High Energy experiments are environments quite special. The CMS magnet is 4 T with currents up to 18000 A [Sor08], the magnets in the tunnel are also rated >2 T. The electromagnetic interferences makes inviable many of the frequencies. When taking into account the bandwidth needed, low frequencies with big antennas also become unpractical. Even if the frequencies near the visible spectrum are considered, we have to remind that the concrete walls are several meters thick for protecting the equipment and the operators from the radiation.

Other option could be to make the ADC conversion as near as possible as the sensor, lets say, inside the detector. This is the ATLAS approach using the ELMB, but TOTEM would have need an improved version of the ELMB qualified for higher radiation levels. This kind of subproject was out of the scope of TOTEM experiment and the DCS.

The approach for the TOTEM experiment is to have all the readout electronics (as mentioned in some other chapters) in a safe area, USC55, and all the sensors in the tunnel or in UXC55. Those signals will be driven through long cables, even more than 320 m, and several patch panels as defined in Section 10.2, *Cabling* up to USC55. Inside the USC55 the cables and signals are rearranged in a suitable way for the readout electronics as expressed in Figure 10.10.

The rearrangement of 1500 sensor wires into the readout is far from trivial. Additionally, another 500 wires are needed for low voltage and high voltage. The tunnel patch panels are designed to minimize the number of long cables and connectors, while the readout electronics is designed to group similar signals. One example of this mismatch are the High Voltage cables. They are composed by 48 wires, but only 40 are used for the voltages, the remaining 8 are reused for 2 vacuum sensors. The RADMON ISC are 4 sensors with a common ground, so they cannot be powered simultaneously.

Figure 10.1 presents some cost percentages with respect of the total material cost.

Concept	Percentage
Power supplies	7 %
Cabling	7 %
ELMB % connectors	3 %
Rest of electronics	28 %
rest of DCS	3 %

Table 10.1: Material cost distribution (partial)

10.2 Cabling

10.2.1 Available cables

A peculiarity of the TOTEM experiment was that the sensor cables were installed only using an estimation of needs. No clear design or in detail study was done previously. After the long cables installation, the electronics design was finalized.

Roman Pots

For the Roman Pots all the environmental sensors are mounted inside the motherboard or plugged directly into it. From this board a series of ‘short cables’ drives the signals to a patch panel in the LHC tunnel. The idea of this patch panel is to reduce the number of cables needed, regroup many of them, and deal with the LV sensing.

From this patch panel up to the USC55, the RADMON signals are driven via 4 ND100 cables, one per RP station, with 50×2 conductors (100 wires) each. The temperature is driven by 8 NE48 cables with 24×2 conductors (48 wires) each. The vacuum signal is driven together with the HV in another set of 12 NE48 cables.

The first design done by the electronics group was to use the temperature sensor included in the DCU chip and read it using the DAQ system. This sensor is intended for safety and control interlocks. I rejected this design because the risk was unacceptable. In this situation the DCS has not a direct feedback loop and the sensor readout should be done through a cluster of DAQ PCs. The DCS would power the electronics without knowing the temperature in the board. Then if there is no electrical burnout, the DAQ systems starts with the readout, and after some time the DCS could get some information about the status of the electronics.

The resulting configuration is highly suboptimal. There are no spares and not all the sensors are redundant. Moreover if a humidity sensor is needed to prevent condensation, there is not a easy way to do it. The full list of Roman Pot sensors was defined in Figure 9.1. But for a Roman Pot upgrade it is mandatory to install more cables in the tunnel.

Telescope 1

From T1 up to USC55 the signals are driven via MCA36P cables with $12 \times 18 \times 2$ conductors (36 wires) each. For each quarter 3 cables are installed, see Table 10.3.

This configuration provides enough spares for additional sensors during the lifetime of the experiment. The full list of Telescope 1 sensors was defined in Figure 9.2.

Telescope 2

From T2 to the USC55 the signals are driven via MCA36P cables with $12 \times 18 \times 2$ conductors (36 wires) each. For each quarter 3 cables are installed, Table 10.4.

10.3. Pinout tables and hardware generation scripts

Side	Name ST-EL	Name TOT-CMS	Cable Type	Function	End
Left	1514674	TR.1_SP	NE48	signals	RP220
Left	1514675	TR.1_SP	NE48	signals	RP220
Left	1514676	TR.1_RM	ND100	radmon	RP220
Left	1514700	TR.1_SP	NE48	signals	RP150
Left	1514701	TR.1_SP	NE48	signals	RP150
Left	1514702	TR.1_RM	ND100	radmon	RP150
Right	1514754	TR.1_SP	NE48	signals	RP220
Right	1514755	TR.1_SP	NE48	signals	RP220
Right	1514756	TR.1_RM	ND100	radmon	RP220
Right	1514780	TR.1_SP	NE48	signals	RP150
Right	1514781	TR.1_SP	NE48	signals	RP150
Right	1514782	TR.1_RM	ND100	radmon	RP150

The cable 'function' labeling was assigned before the sensor distribution.

Table 10.2: Installed cables for Roman Pot stations [TL06a]

Side	Name ST-EL	Name TOT-CMS	Cable Type	Function	End
-	11	TO.1_SE	MCA36P	T1 sense control	rack X4E72
-	12	TO.1_SE	MCA36P	T1 sense control	rack X4L72
-	13	TO.1_TM	MCA36P	T1 T monitor	rack X4E72
-	14	TO.1_TM	MCA36P	T1 T monitor	rack X4L72
-	15	TO.1_Spare1	MCA36P	T1 spare	rack X4E72
-	16	TO.1_Spare1	MCA36P	T1 spare	rack X4L72
+	11	TO.1_SE	MCA36P	T1 sense control	rack X3L73
+	12	TO.1_SE	MCA36P	T1 sense control	rack X3L73
+	13	TO.1_TM	MCA36P	T1 T monitor	rack X3L73
+	14	TO.1_TM	MCA36P	T1 T monitor	rack X3L73
+	15	TO.1_Spare1	MCA36P	T1 spare	rack X3L73
+	16	TO.1_Spare1	MCA36P	T1 spare	rack X3L73

The cable 'function' labeling was assigned before the sensor distribution.

Table 10.3: Installed cables for T1 quarters [TL06b]

This configuration provides enough spares for additional sensors during the lifetime of the experiment. The full list of Telescope 2 sensors was defined in Figure 9.3.

Side	Name ST-EL	Name TOT-CMS	Cable Type	Function	End
-	17	TO.2_SE	MCA36P	T2 sense control	rack X4U72
-	18	TO.2_SE	MCA36P	T2 sense control	rack X4R72
-	19	TO.2_TM	MCA36P	T2 T monitor	rack X4U72
-	20	TO.2_TM	MCA36P	T2 T monitor	rack X4R72
-	21	TO.2_Spare1	MCA36P	T2 spare	rack X4U72
-	22	TO.2_Spare1	MCA36P	T2 spare	rack X4R72
+	17	TO.2_SE	MCA36P	T2 sense control	rack X3R73
+	18	TO.2_SE	MCA36P	T2 sense control	rack X3R73
+	19	TO.2_TM	MCA36P	T2 T monitor	rack X3R73
+	20	TO.2_TM	MCA36P	T2 T monitor	rack X3R73
+	21	TO.2_Spare1	MCA36P	T2 spare	rack X3R73
+	22	TO.2_Spare1	MCA36P	T2 spare	rack X3R73

The cable 'function' labeling was assigned before the sensor distribution.

Table 10.4: Installed cables for T2 quarters [TL06b]

10.3 Pinout tables and hardware generation scripts

The assignment of each wire from a ADC channel or each voltage channel into the detector is usually called pinout. It is necessary to map all the connectivity among all the patch panels and connectors from the detector up to the corresponding equipment in USC55.

I structured all this information in [DL08] according to the levels of the Product Breakdown Structure of each detector. The names follow as much as possible the official nomenclature described in [Rug08] and Section 7.4.3, *Naming Scheme*. The responsible for each part of the cable/control chain is also identified. A red double line divides the cabling part from the DCS configuration part.

Those tables are considered to be a ‘source’ for the configuration of the DCS. They are implemented using EXCEL, but could be converted into a web interface, Some automatic processing is done using embedded formulas in the cells.

Much of the innovation resides on this formalization. Not only the the basic cabling information, but all the connectivity into the DCS system inside the counting room is specified. This helps defining responsibilities and let the cabling technicians/responsible filling the table by their own. The DCS PVSS names (hardware and logical) are automatically generated from the other cells in the row; so as the cabling part is being filled, the DCS PVSS names are automatically constructed. I do not have any feedback from other LHC experiment using such approximation. Both teams (electronics and DCS) usually work less integrated and exchanging information in pieces of paper with incompatible conventions and philosophy. As expressed in the *Preface* of this thesis, that set of tables has made some DCS developments 3 times faster than usual, but even more important is the confidence of what is agreed in the table is being implemented.

Part name	Items	Purpose
Product breakdown structure	sector station unit pot function	Identify univocally pots, units and stations
Motherboard pinout	RPMB pad RPMB signal name RPMB connector pin	Drive signals and voltages between the motherboard PCB and external cables
Short cables	19 pins connector in pot short cable name short cable wire 19 pins connector in patch pannel	Connect the motherboards into the station patch pannel
Tunnel patch pannel	19 pins connector in patch pannel patch pannel wire 48 pins connector in long cable	Connect the all short cables of 1 station into the long cables
Long cables	48 pins connector in tunnel long cable name long cable wire 48 pins connector in counting room	Drive the signals and voltages from the tunnel into the counting room and alcoves
Patching into crates/elmb/and control devices	48 pins connector flat cable ID flat cable pos	Do discrete wiring of the cables arriving from the tunnel into the connector the counting room equipment
Configuration of hardware control devices	ELMB bus ELMB ID (decimal) ELMB channel ELMB connector pin ELMB channel polarity	Position of the boards into the crates, buses, IP addresses
DCS PVSS hardware names and logic names	DCS hardware name DCS logical name	Those names are the DataPoints to integrate with the FSM logic and monitoring functionalities

Table 10.5: Structure of the pinout table

At a later step this table is automatically processed by custom made script in PVSS. It check the logic and hardware names. If not defined already in PVSS, they are generated according to the table. Also all the crates, CAN buses, ELMB and remaining configuration is set up automatically inside the PVSS project to match the table. In the same way, the DIP information for the DSS is also generated.

10.3. Pinout tables and hardware generation scripts

Further improvement could include automatic detection of polarity mismatches, wires not plugged, wires plugged twice,...

[illegible]

10.4 RADMON Connectivity into the DCS readout electronics

10.4.1 The ELMB system connectivity

The RADMON readout uses the Embedded Local Monitor Boards (ELMBs) [KCD⁺06], [CT05] and ELMB-DACs [KTH⁺02]. This approach simplifies the integration of the RADMON sensors in the Detector Control System (DCS), and has been adopted by ATLAS after extensive tests [MCG⁺07], making it compatible with the existent JCOP DCS structure. On the other hand, this approach imposes some constraints on the flexibility of the whole system.

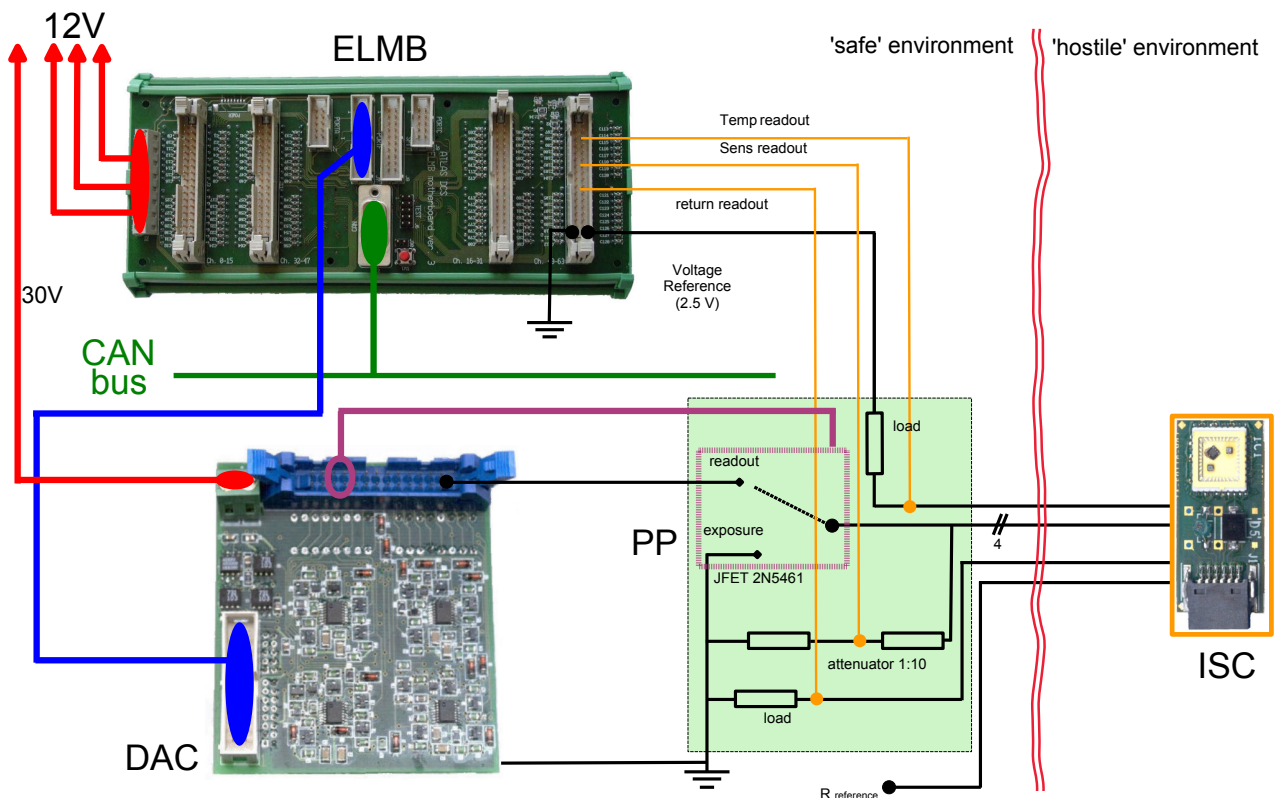


Figure 10.2: Schematic of the ELMB RADMON readout system developed by the ATLAS experiment

An example of readout is given in Figure 10.2. Four blocks can be identified: the ELMB, the ELMB-DAC, the RADMON Patch Panel and a ISC. The ISC is referred to a single RADMON sensor (*Sens*) and 1 NTC (*Temp*). The readout current injected in each sensor is monitored by measuring the voltage drop on a small resistor placed in the return line (*Return*). An extra-line is routed to the control room for the readout of the 1 kW reference resistor (*R_{reference}*).

To drive the sensors on the ISC during the readout sequence, the ELMB-DAC boards and patch panel are needed. The ELMB-DAC is a 16 channel Digital Analog Converter of 12-bit resolution. By one side connects to the digital output of the ELMB and one external power supply, and by the other it generates 16 analog outputs for the patch panel. The output current range from ELMB-DAC [KTH⁺02] can be set from 0 to 20 mA or from 0 to 1 mA, by changing one onboard resistor per channel. The maximum output voltage of a DAC channel is set by the supply voltage and is limited to 30 V due to power dissipation constraints of some components installed on the DAC itself. However, working

in the current range 0 to 1 mA as demanded by all RADMON sensors, it is not possible to reach the operating conditions where dissipation problems can appear [CT05].

Up to four ELMB-DAC boards can be connected and controlled by one ELMB. However in the RADMON readout system only two DAC boards will be chained to a single ELMB due to the relationship between ADC channels and DAC outputs explained in Section 10.4.2, *RADMON Patch Panel board design*.

To fulfill the TOTEM RADMON sensors powering needs it is necessary to apply the maximum of 30 V to the ELMB-DAC board using the external power line. The ATLAS RADMON sensors will not receive so high radiation doses, and the ELMB-DAC board can be powered using the 12 V from ELMB Analog part. This additional power line has caused some problems as are explained in Chapter 14, *Verification of the DCS*.

10.4.2 RADMON Patch Panel board design

The RADMON patch panel is the key element to adapt the different sensors signal for connecting into the ELMB motherboard. It takes the role of switching the sensor from *exposure mode* to *readout mode* and back.

The DAC board is composed of 16 outputs. 12 of them are used for driving patch panel JFET transistors performing the sensor mode switching of 3 ISC. The remaining 4, are grouped together to generate the current that the sensor needs to be readout; this is grouping called the *switch array*.

A new patch panel was designed following the one produced for the ATLAS experiment [KCD⁺06]. The ATLAS patch panel (see Figure 10.3) is used for the readout of 6 ISC in a configuration similar to the TOTEM ‘base’ configuration described in Section 9.1.4, *RADIation MONitors sensor* (two sensors + NTC + RL + resistor). In this patch panel, the signals to the ISC are provided via 6 LEMO 4-pin connectors.

We decided to change the design the the Patch Panel because of the following reasons:

- There are no ‘ATLAS Patch Panel’ available. It would be needed a full new production.
- The patch panel ISC connector is modified to match our ‘redundancy’ ISC configuration. Now a single connector is provided for all those devices (2 RadFETs + 2 pins + NTC + return line not plugged).
- With this solution we also make available the 6 free ADC channels on a separate connector.

The new TOTEM patch panel is shown in Figure 10.4.

The fact of having still 6 ADC channels free, suggests the idea of adding the connectivity for a fourth ISC; but this is not possible. Unfortunately the limiting factor in this setup is the number of DAC channels, 16 in total, to power the RadFETs, the pin sensors and to operate the switch array; and there are not 4 of them free for an additional ISC.

Already in the original ATLAS patch panel, 4 DAC channels are used to drive the *switch array*. Setting all the DAC channels in the operating range 0 to 1 mA; with 4 channels it is possible to provide 4 mA over 10 kOhm resistors to generate the 40 V needed to guarantee that all switches will properly work over the maximum ΔV for the sensors readout (30 V).

Moreover the switch array is the ‘critical’ element with this readout scheme. If all transistors are driven by a unique DAC channel, in case of its failure, it is not possible to drive the switches any longer and we ‘lose’ all ISCs connected to the patch panel. Using several channels, the risk to have all 4 channels out of order is lower.

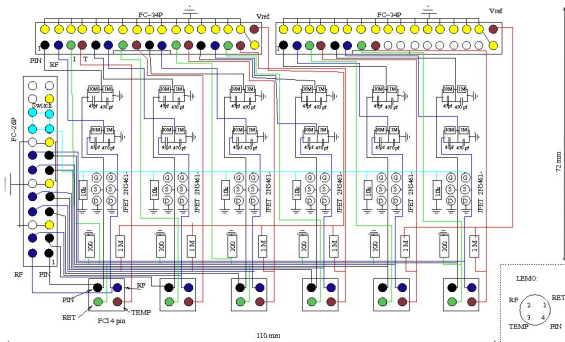


Figure 10.3: ATLAS patch panel board for the readout of 6 ISC in base configuration

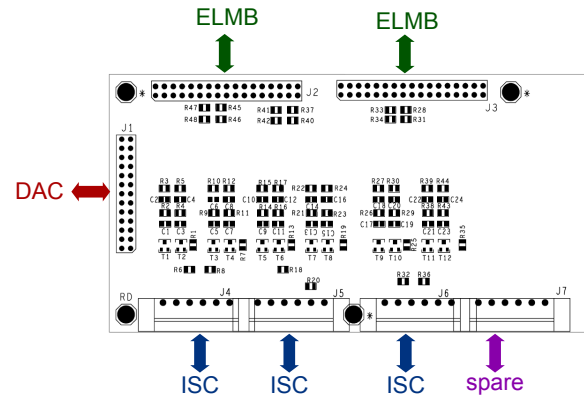


Figure 10.4: TOTEM patch panel board for the readout of 3 ISC in redundancy configuration

10.5 Rack

10.5.1 ELMB and ELMB-DAC powering

The ELMB needs 3 power inputs of 12 volts:

- for the ANALOG part (current consumption ~ 15 mA)
- for the DIGITAL part (current consumption ~ 10 mA)
- for the bus communication (current consumption ~ 20 mA)

In TOTEM the ELMB will use independent power supplies for each one of the parts. This approach is the same as the CMS-ECAL detector [Zel07]. However for each part the all ELMB of the same detector will be grouped together, see Figure 10.10. Each detector and the central part will be fully independent. The total number of Power Supplies is 21, and it is calculated in Table 10.9.

Also the ELMB-DAC board needs to be powered with 30 V (current consumption ~ 30 mA).

The current consumption values given here are obtained from measurements done on a prototype system as described in Section 10.4, *RADMON Connectivity into the DCS readout electronics* with 2 ELMB-DAC boards connected to one ELMB motherboard [Man08].

To fulfill such requirements the low voltage regulated power supply *SIEMENS SITOP power flexi* Type 6EP1-353-2BA00 has been selected.

The main features of this device are [SIE07]:

- Primary switching regulators with adjustable **output voltage** from **3 to 52 VDC**
- Setting of the output voltage through potentiometer or remote controlled through an analog signal
- Sense line connection, power good signal, current monitoring signal
- Top hat rail mounting
- Input voltage 120/230 VAC, Input current 2.2/0.9 A
- **Output current max.** 10 A (at 12 V), >4 A (at 30 V)
- **Output power max.** 120 W
- Residual ripple <50 mV
- Dimensions (W x H x D) 75 × 125 × 125 mm

Those power supplies will be grouped and mounted in a DIN rail as shown in Figure 10.5.

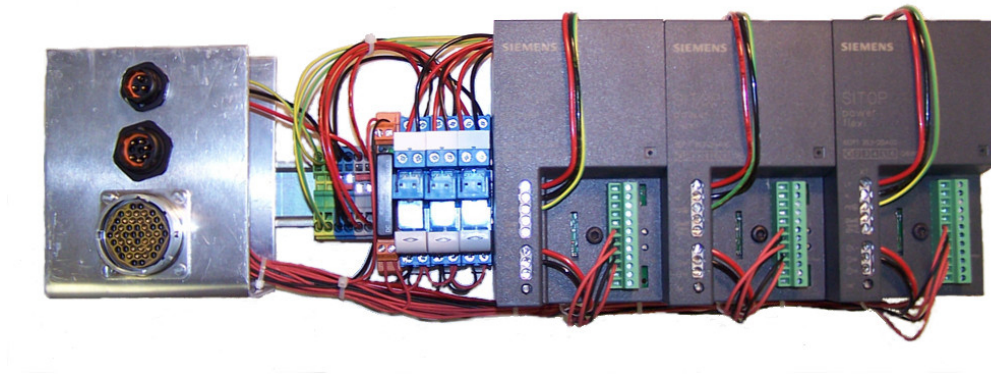


Figure 10.5: Power Supplies mounted in the DIN rail

The CERN ‘standard’ power supply for ELMB produced by PH-ESS has been also considered, see [Sch05]. This kind of module provides outputs for 2 ATLAS CAN branches that powers directly the ELMB over the bus. Internally two TRACO power supplies are used, one of them powers Analog and the other Digital+CANbus. However we do not have the need to power the ELMB trough the CAN bus, as all the equipment is located in the same rack. The advantage of using SIEMENS power supplies are:

- It is a commercial solutions, not a special production. The spares policy and maintenance should be simpler.
- Full isolation among the Digital, Analog and CAN parts.
- I have developed a system using relays for being able to disconnect the power of the ELMB, so in this way a had reset can be done remotely; see Section 10.5.2, *ELMB power relays*.
- That particular SIEMENS power supply has output signals for monitoring the I and V outputs (and manage them remotely). An additional PT1000 has been glued on top of each one.
- The overall cost 3 Siemens power supplies is less than the CERN system with 2 TRACO power supplies.

TOTEM must be able to monitor and record the evolution of radiation levels even during a power cut, and integrate this data into the LHC post-mortem analysis. In consequence, those power supplies will be connected to the CMS UPS. In any case the ELMB power relay mechanism allows to perform a power cycle (hard reset) for each ELMB box.

10.5.2 ELMB power relays

As stated by [Zel07] it is highly convenient having a way to power cycle the ELMB. Under some circumstances they stop responding to CAN commands and the reset command sent via MNT is not being executed. The only solution to recover the communication is to do a hard reset. For this reason I developed the system shown on Figure 10.6.

A DC relay (10.7) together with 3 AC relays (10.8) are used. Notice that no signal of 30 V or 5 V is needed normal operation, only for the ‘reset’ operation. When the ELMB Digital Output generates a 5 V, the DC relay output provides 30 V to the AC relay and this changes the status from ‘Normally Closed’ to ‘Open’.

The reason for not using a DC relay directly in the outputs of the Power Supplies is that the Crydom relay is only rated 3 A, while the Siemens Power Supply up to 10 A. Unfortunately there was not any

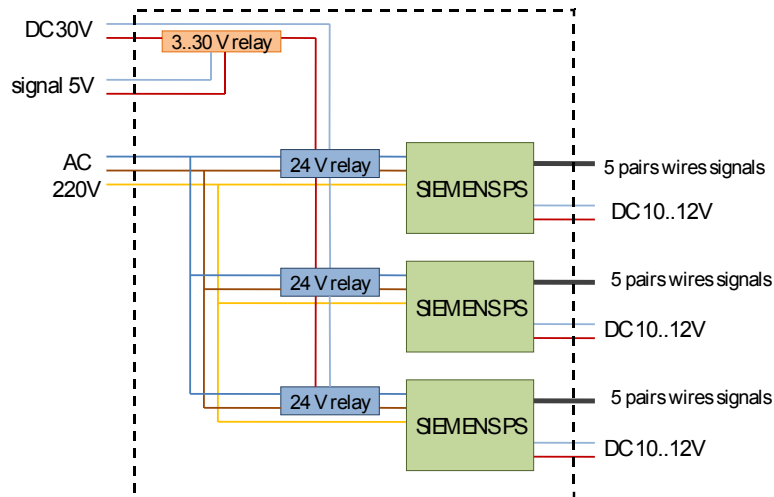


Figure 10.6: Relays schematic



Figure 10.7: Crydom Relay 3 V (solid state)



Figure 10.8: Finder Relay 24 V (mechanical)

rail mounted relay in the market rated around 30 A to branch the 3 Power Supplies in the same relay.

It is interesting to remark that the first test was done with a mechanical relay rated 6 V for switching the AC, but the ELMB is not able to provide enough current for activating this relay. Evenmore the measurements of the ELMB went out of range while just attaching that relay to the digital output, before generating any output. It seems that some electromagnetic disturbance from the AC currents flows through that mechanical relay into the ELMB circuitry.

10.5.3 CAN bus chain

All the sensors from each detector will be grouped by type, so each ELMB will only read one kind of sensors. This will allow fine configurations for ADC ranges and frequency.

Each detector will have it own dedicated CAN bus. Additional CAN bus lines are needed for the readout or the RADMON and for the common equipment such as the global monitoring for the ELMB Power Supplies.

This makes a total of 7 CAN buses. However the CAN-USB adaptor [SYS06] is able to handle 8 buses in each internal unit, and due to the proximity of the DCS computers rack and the ELMB rack, 16 CAN buses (without power) had been installed between both racks.

PC	sysworxx	unit	bus	ELMB ID (hex)	ELMB ID (binary)	ELMB ID (dec)	ELMB ID (drawing)	ELMB serial	motherboard serial	FUNCTION
TOTEM-DCS-01	TOP	00	00	01	000001	01				spare
TOTEM-DCS-01	TOP	00	00	02	000010	02				spare
TOTEM-DCS-01	TOP	00	01	03	000011	03				PS monitoring
TOTEM-DCS-02	TOP	01	08	11	010001	17				RP radmon
TOTEM-DCS-02	TOP	01	08	12	010010	18				RP radmon
TOTEM-DCS-02	TOP	01	08	13	010011	19				RP radmon
TOTEM-DCS-02	TOP	01	08	14	010100	20				RP radmon
TOTEM-DCS-02	TOP	01	09	15	010101	21				RP vacuum
TOTEM-DCS-02	TOP	01	09	16	010110	22				RP temp
TOTEM-DCS-02	TOP	01	09	17	010111	23				RP cooling
TOTEM-DCS-02	TOP	01	09	18	011000	24				RP cooling
TOTEM-DCS-03	BOTTOM	00	04	21	100001	33				T1 radmon
TOTEM-DCS-03	BOTTOM	00	04	22	100010	34				T1 radmon
TOTEM-DCS-03	BOTTOM	00	05	23	100011	35				T1 temperature
TOTEM-DCS-03	BOTTOM	00	05	24	100100	36				T1 pressure+humidity
TOTEM-DCS-04	BOTTOM	01	12	31	110001	49				T2 radmon
TOTEM-DCS-04	BOTTOM	01	13	32	110010	50				T2 temperature
TOTEM-DCS-04	BOTTOM	01	13	33	110011	51				T2 pressure+humidity
TOTEM-DCS-LAB	LAB	emulation	emulation	emulation						development
TOTEM-DCS-LAB	LAB	emulation	emulation	emulation						development
TOTEM-DCS-LAB	LAB	emulation	emulation	emulation						development

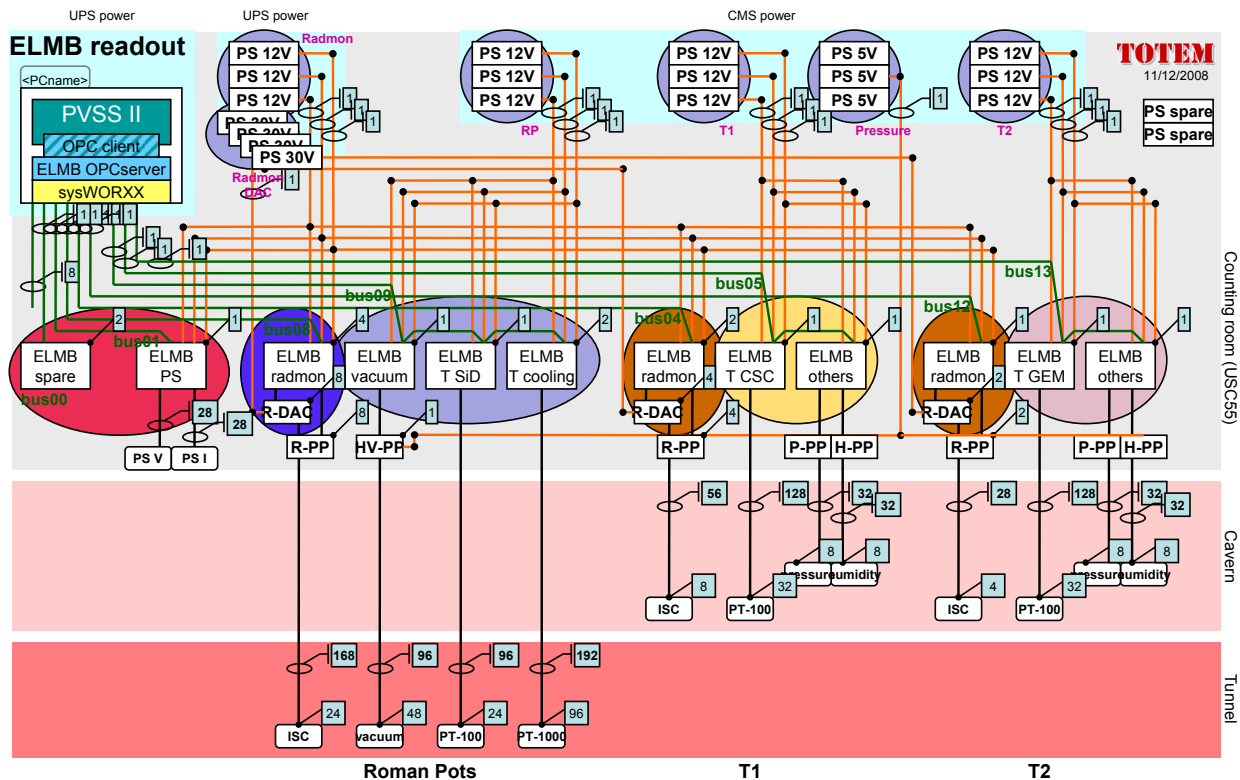
Figure 10.9: Resume of all the ELMB hierarchical configuration

This distribution of 18 ELMB into 8 buses takes into account the internal Sysworxx CAN-USB design, see Section C.3.3, *Sysworxx CAN to USB adapter*. Those adaptors internally are two units that convert 8 buses into an single USB cable. As there are two CAN-USB adapters, there will be 4 USB cables in total. Each cable will be assigned to each detector (RP, T1 and T2) and to global monitoring. When a bus ID is used in one unit, this ID is not reused in the other unit. In principle they could be reused because they are connected to different computers. But this additional constrain allows that if one unit is broken, all the cables can be replugged in the other unit. Then, the proper PVSS component, without any modification, can be reinstalled into the computer that is attached to the CAN-USB unit where the cables have been replugged. Using the distributed capabilities of PVSS the datapoints will be recognized by the rest of the PCs and the FSM logic will not be affected.

10.5.4 DCS rack interconnections

Some adaptation is needed between the thick cables arriving from the tunnel the ELMB motherboard connectors. Several alternatives were considered, as the connectors blocks of Figure 10.11. This device is used by the ALICE experiment, and also used in my systems in the laboratory and test beam. It allows a quite easy reconfiguration of connectivity but it does not scale well (remind the need of plug over 1500 wires).

Each different color of the circles from Figure 10.10 represents one ELMB box. The idea is to group all the related ELMB into the same support. Examples are the RP RADMON sensors and the rest of RP sensors. The DC current is provided to that box and also the thick cables arriving from the tunnel. That box communicates to the DCS through a single CAN bus cable. All the wire to ADC channel pinout shown in Figure 10.1 is done inside the box.



TOTEM DCS – Hardware overview diagrams

Page 8 of 11

Figure 10.10: Implementation of the global ELMB and RADMON system in the TOTEM experiment; from [Luc08c]

The final proposal for installing the ELMBs is described in [LM08]. In this document Johan Morant and I define a set of horizontal boxes to be mounted in the rack with the ELMBs inside. The connectivity to the thick cables is done using Burndy connectors on the front side of the boxes. Meanwhile the connectivity into the ELMB is done using a flat cable; see Figure 10.12. At one side there is the usual flat cable connector for the ELMB motherboard, and at the other each wire has been split and crimped. Each wire is inserted into the female Burndy connector of the front side of the box following the pinout defined in the tables of Section 10.3, *Pinout tables and hardware generation scripts*. The photography is enough descriptive about the difficulties of this assignment; a Burndy connectors is shared among several flat cables and vice versa.

The back side of the rack is filled with the Power Supplies of Figure 10.5 needed for the boxes. Each box is dedicated exclusively only to a detector or the common part. Also the boxes for RADMON sensors are independent because of the need of hosting a dedicated patch panel and the DAC. Examples of that design are given in Figure 10.13 and in Figure 10.14.



Figure 10.12: Cable used for the pinout assignment between the long cable and the motherboard



Figure 10.13: ELMB box for RP temperature and vacuum sensors

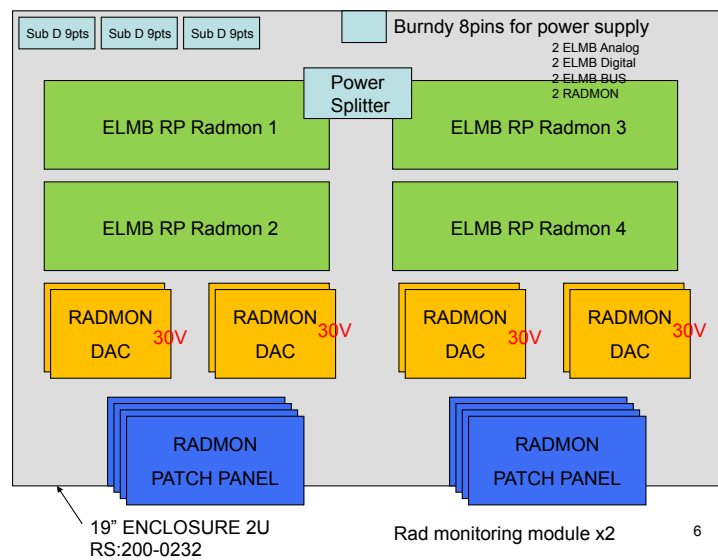


Figure 10.14: ELMB box for RP radmon sensors

10.6 Equipment used

Table 10.6 resumes the number of ELMB needed for the TOTEM sensors frontend, including spares, while Table 10.7 details that information for the additional PCB needed for the RADMON. Notice that there are only two spares in the laboratory, and this number should be increased.

Location	RADMON	Temp. Detector	Temp. Additional	Vacuum	Pressure	Humidity	PS sensing	Spares	TOTAL
Common	0	0	0	0	0	0	1	2	3
RP	4	1	2	1	0	0	0	0	8
T1	2	2	0	0	0.5	0.5	0	0	5
T2	1	2	0	0	0.5	0.5	0	0	4
Installation SUBTOTAL	7	5	2	1	1	1	1	2	20
Spares and Laboratory	0	0	0	0	0	0	0	2	2
TOTAL	7	5	2	1	1	1	1	4	22

Table 10.6: ELMB global distribution

Location	Measurement Locations	RADMON Patch Pannel	RADMON DAC	ELMB
Common	-	-	-	-
RP	24	8	8	4
T1	8	4	4	2
T2	4	2	2	1
Installation SUBTOTAL	36	14	14	7
Spares and Laboratory	-	3	3	-
TOTAL	36	17	17	7

Table 10.7: Summary of DAC boards and Patch Panels

For the temperature sensors ELMB adapters had to be procured as shown in Table 10.8. They are defined as explained in Section C.4.5, *Adapters* and in Section 9.1.1, *Temperature*.

Location	PT100	PT1000	TOTAL
Common	-	16	16
RP	16	32	48
T1	32	-	32
T2	32	-	32
Installation SUBTOTAL	80	48	128
Spares and Laboratory	48	32	80
TOTAL	128	80	208

Table 10.8: ELMB adapters

The total number of Power Supplies is 21, and it is calculated in Table 10.9.

Table 10.10 summarizes the number of relays needed according to the design.

10.7 Networks interconnection

For the TOTEM experiment 3 different computer networks are relevant. They are:

Location	ELMB	DAC	Winston bridges	TOTAL
Common	3 PS	1 PS	1 PS	5 PS
Roman Pots	3 PS	-	-	3 PS
T1	3 PS	-	-	3 PS
T2	3 PS	-	-	3 PS
Hot Spares	-	-	-	2 PS
Installation SUBTOTAL	12 PS	1 PS	1 PS	16 PS
Cold Spares and Laboratory	-	-	-	5 PS
TOTAL	12 PS	1 PS	1 PS	21 PS

Table 10.9: ELMB power supplies

Location	3 V DC	24 V DC	TOTAL
Common	1	3	4
RP	1	3	4
T1	1	3	4
T2	1	3	4
Installation SUBTOTAL	4	12	16
Spares and Laboratory	3	6	9
TOTAL	7	18	25

Table 10.10: ELMB powering relays

- **CERN General Purpose Network (GPN):**

This is the internal network of CERN. The outgoing connectivity towards Internet is ‘unrestricted’. All the office and development machines are connected here. Also the *bigbrother.cern.ch* server is located here. Incoming connections from the Internet for servers allocated inside must be declared and authorized.

- **CERN Technical Network (TN):**

All the CCC, LHC and other accelerators control equipment is connected here. It is isolated from the CERN GPN and Internet. The DIP servers and related machines are contained here.

- **CMS Network:**

This is another independent network for CMS the experiment. It is also isolated from other networks such as Internet, the CERN GPN, and the CERN TN. Specific connectivity request to machines inside the Technical Network can be authorized.

This situation affects the the TOTEM DCS in the following way:

- The DCS nodes running PVSS are connected into the CMS Network as well as the DAQ nodes.
- The motorization control (FESA and PXI) is located in the CERN TN. In this way they can be integrated into the LHC Collimator Control.
- The PLC for the cooling plant is connected also to the CERN TN.
- It is necessary to ensure the communication between the FESA machine, and cooling PLC to the DCS nodes.
- It is necessary to identify the machines from the CERN TN that generate the DIP information relevant for TOTEM.
- It is necessary to have a development machine from the CERN GPN ‘trusted’ into the CERN TN for DIP development.
- To control the DCS nodes it is necessary to use the CMS Terminal Servers and web interface. In the terminal server a distributed PVSS project is running. It will connect to the DCS nodes and

show the proper panel using PVSS. This is the only way to access the DCS from the CERN GPN.

- The *bigbrother.cern.ch* webserver for documentation and code repository must be available from the CMS Network and from the Internet.

In Figure 10.15 most of the machines are represented. Arrow shows the initial connection request.

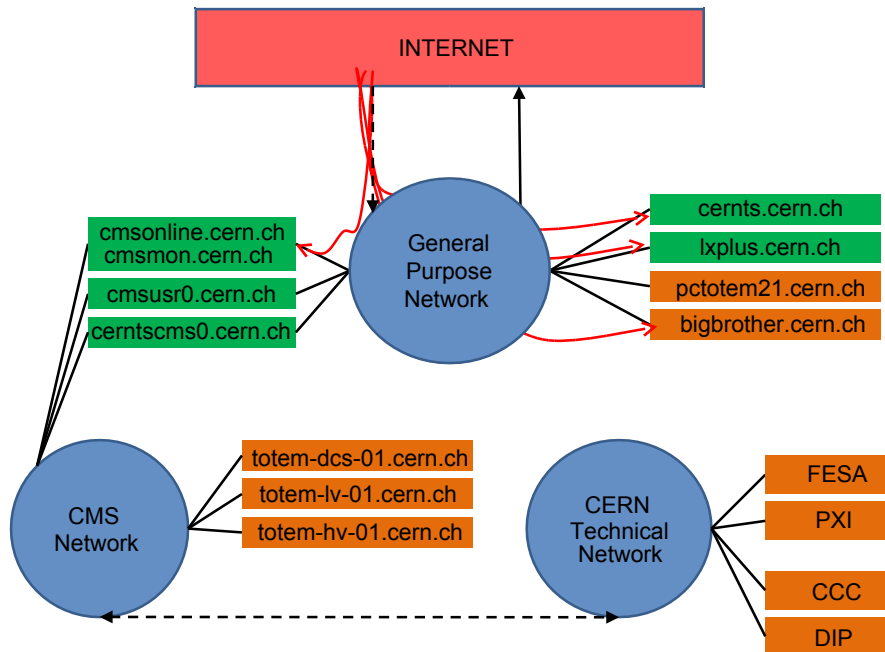


Figure 10.15: Representation of the different networks involved

It is important to notice that the linux cluster *lxplus.cern.ch* allows to do ssh tunneling and establish direct connection from Internet to specific machines and ports inside the CERN GPN. The same approach is also valid from the CERN GPN into the for CMS Network using the *cmsusr.cern.ch* linux cluster.

Operational logic

11.1 Introduction to the FSM trees

The FSM structures of the CMS detectors will be divided in two trees [GGV06]. One of the trees will be called the *FSM detector tree* and the other one *FSM hardware tree*. The TOP node used for integration in the CMS-DCS should be the same for both trees, both trees are overlapping among them. A real extract of the final tree structure is given in Figure 11.6.

11.1.1 FSM detector tree

The idea behind the *FSM detector tree* is to have a decomposition of the detector following the functionality rather than the connectivity. From the physicist point of view those are the partitions or PBS structure of the detector. In this FSM tree it is possible to operate, together or independently, the different detector's partitions.

However this is not a complete description of the detector. A HV crate state is not reflected via FSM node state in a *FSM detector tree* but in a *FSM hardware tree*.

Figure 11.1 shows the TOTEM *FSM detector tree*:

The nodes colored in light blue on the figure do refer to the PBS levels of the detector. They will become partitions in FSM terminology. The top-most node called CMS_TOTEM and is the integration interface to the CMS experiment for acting as a CMS detector. This can be called the detector supervisor.

Colored in red, deep blue, green, brown, pink,... as in the Hardware overview diagrams of Appendix B, *Hardware overview diagrams* appear the detector subsystems (HV, LV, environmental sensors, cooling, motorization,...). This is the low level control/monitoring area.

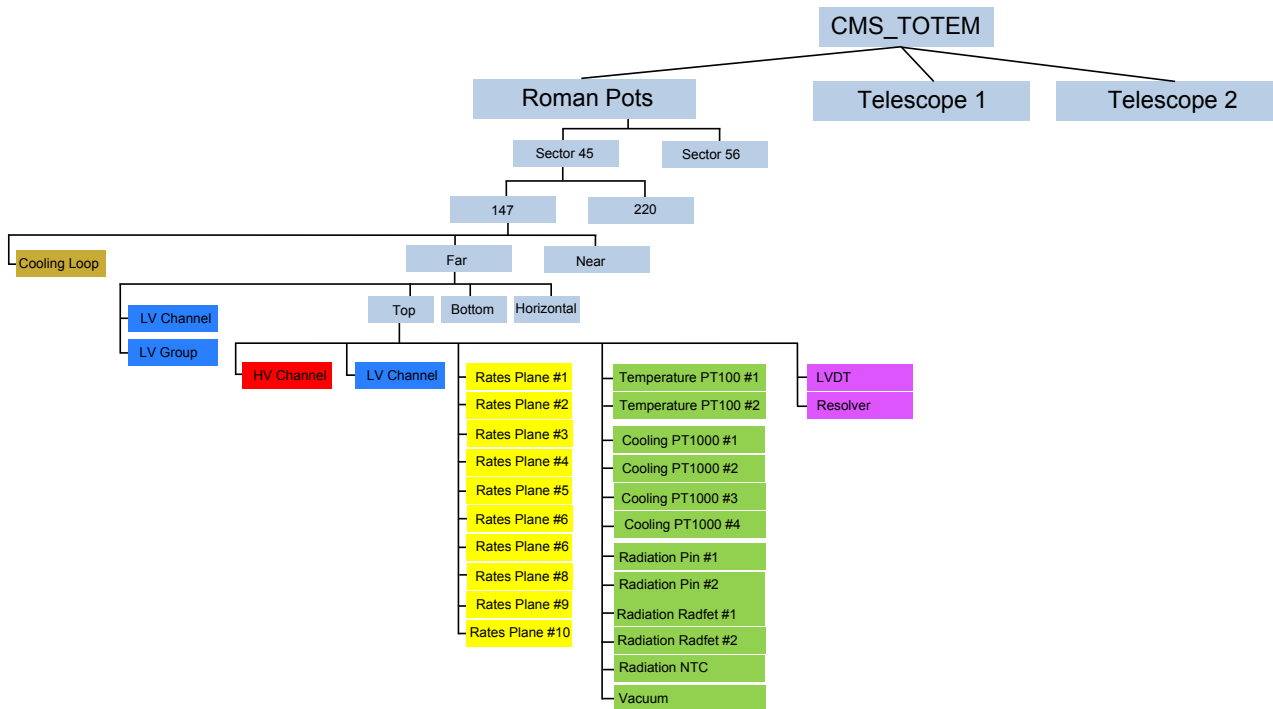


Figure 11.1: FSM detector tree

11.1.2 FSM hardware tree

The *FSM hardware tree* is useful when trying to identify a problem related to a particular subsystem (lets say HV) that could be affecting several detector partitions and therefore several branches of the *FSM detector tree*. If a problem is affecting several branches of an FSM tree, one would need to access multiple panels to correlate information and understand it. On the other hand, if the problem is concentrated in an hardware oriented FSM node or a simple structure of nodes this information is correlated in an easier and more efficient way. The *FSM hardware tree* summarizes the subsystems hardware in simple FSM structures and consequently makes these trees very efficient to solve subsystems related problems.

This tree in contrast to the *FSM detector tree* this one is complete and must represent all the control/monitoring functions of the experiment.

The TOTEM *FSM hardware tree* is represented in Figure 11.2.

In light blue there are the PBS related nodes. Children of this node (together with the partitions of the *FSM detector tree*) will be the subsystems most top nodes. These subsystems nodes colored in red, deep blue, green,... will provide an overall state of their subsystem. In this way the HV subsystem is decomposed into a Crate, with several boards, and each board contains several modules. A board is not just the sum of the states of the channels. It has some internal parameters such as temperature sensors unrelated to any specific channel that must be taken into account.

It is important to notice the General node directly under the CMS_TOTEM node. This partition represents all the equipment that is common for all the TOTEM detectors as the high voltage channels and some ELMB.

This tree is of secondary importance compared to the detector *FSM detector tree*. This is just a fast way to solve problems and be complete in the monitoring but the usual operation of the detector is done

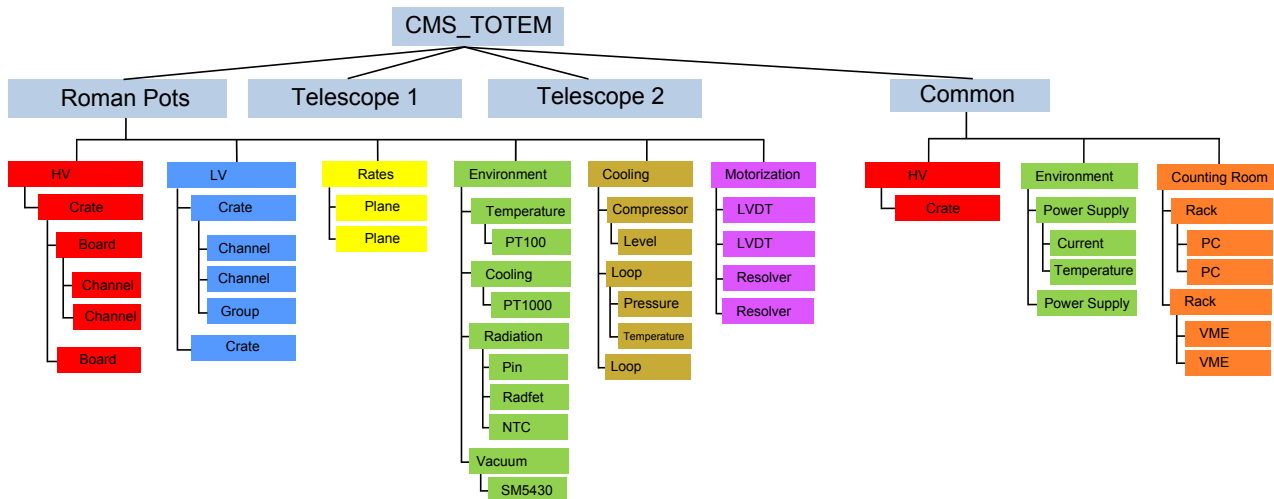


Figure 11.2: FSM hardware tree

with the *FSM detector tree*. For this reason this tree is a secondary objective for the development.

11.1.3 CMS integration

The top node of each detector in CMS should implement a set of states and commands. They are defined in [GGV06] with some tables, lists and graphics as Table 11.1. However, I have resumed all that information using UML into the state model of Figure 11.3.

States	Description
ON	When all subsystems are in state ON High voltages and low voltages are switched on. The detector is ready for data taking.
OFF	When any subsystem is in state OFF Both high and low voltage are switched off. (Long shutdown periods)
STANDBY	When any subsystem is in state STANDBY High voltage is off but low voltages are present.
ERROR	When any subsystem is in state ERROR The detector is not ready for data taking. Automatically recover is not possible. Manual intervention is required.

Table 11.1: FSM states for the CMS_TOTEM integration top node

The rest of the FSM tree nodes below this one try to propagate this semantic adding other states and commands when strictly necessary.

Complementary information must be provided using alarms and the user panels, not increasing the possible number of states. The *FSM detector tree* provides ‘physics’ related information about the detector (like when is it ready for data acquisition) and the *FSM hardware tree* provides hardware information about the detector status.

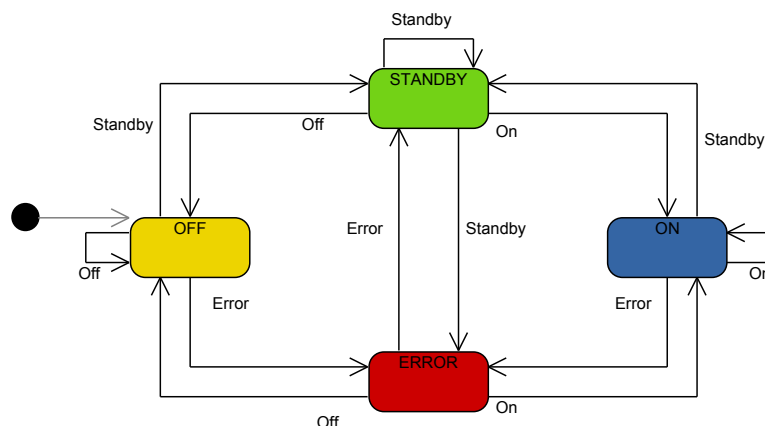


Figure 11.3: USM state model for the CMS_TOTEM integration top node

11.2 Behaviour formalization into FSM hierarchy tables

11.2.1 Using tables

In a first moment the FSM type logic was formalized as tables. An example is given in Table 11.2 and Table 11.3.

This logic maps the value of the datapoint associated to a sensor to the different states of a DU. The DU are also used to send commands to the hardware and monitor its state. The DU are grouped into CU or LU. The decision of using CU or LU is related does not have any implication in the functionality, but only CU have the full features for becoming and independent partition. Inside each CU the logic of the FSM node type check the children (DU, LU or CU) states. If the state of one child change this can escalate and change the state of the parent CU, changing the rest of the children or no other change at all.

For the rest of the tree nodes try to propagate this set adding other states and commands when strictly necessary. Reduce the set of states for the devices as much as possible. Use alarms and the user panels to give complementary information.

The FSM types names have to follow the JCOP naming scheme, that have a different philosophy of Section 7.4.3, *Naming Scheme*. However they follow the TOTEM naming as much as possible. Those names are internal to the DCS and never shown to the operator, and they have even spelling mistakes, *Wiener Maraton* is spelled without *h*.

Each state of the FSM type is colored according to the JCOP agreed colors. Also it has an associated user panel to resume the status of the node children in a graphical or tabular way as described in Chapter 15. Also the basic JCOP panels can be reused and fixed before the official releases takes place.

Table 11.4 shows how the states are propagated form the top most node into the different children.

11.2. Behaviour formalization into FSM hierarchy tables

States	Commands
ON High voltages and low voltages are switched on. The Roman Pot station is ready for data taking.	OFF brings the Roman Pot station to the state OFF
	STANDBY brings the Roman Pot station to the state STANDBY
OFF Both high and low voltage are switched off. (Long shutdown periods)	ON brings the Roman Pot station to the state ON
	STANDBY brings the Roman Pot station to the state STANDBY
STANDBY High voltage is off but low voltages are present	ON brings the Roman Pot station to the state ON
	OFF brings the Roman Pot station to the state OFF
RAMP_UP High voltages are in RAMP UP and low voltages are switched ON. The Roman Pot station is not ready for data taking	OFF brings the Roman Pot station to the state OFF
	STANDBY brings the Roman Pot station to the state STANDBY
RAMP_DW High voltages are in RAMP DOWN and low voltages are switched ON or OFF. The Roman Pot station is not ready for data taking	ON brings the Roman Pot station to the state ON
	STANDBY brings the Roman Pot station to the state STANDBY
MIXED When high or low voltage channels are not in the equal state	ON brings the Roman Pot station to the state ON
	OFF brings the Roman Pot station to the state OFF
ERROR Manual intervention is required	STANDBY brings the Roman Pot station to the state STANDBY
	ON brings the Roman Pot station to the state ON
	OFF brings the Roman Pot station to the state OFF

Table 11.2: totRpPot; TOTEM Roman Pot individual Pot states and commands

States	Commands
ON Low voltages channel is switched ON	OFF brings the low voltage channel to the state OFF
OFF Low voltages channel is switched OFF	ON brings the high voltage channel to the state ON
ERROR Manual intervention is required	OFF brings the high voltage channel to the state OFF
	ON brings the high voltage channel to the state ON
NO_COMM No connection to the Wiener Maraton Crate	

Table 11.3: FwWienerMarathonChannelTot; TOTEM Wiener Low Voltage Channel states and commands

CMS_TOTEM	tot GeSv	FwCaen CrateSY1527 Tot	tot RpSv	tot RpStation	tot RpUnit	tot RpPot	FwCaen Channel Rp	FwWiener Marathon Tot	FwWiener Marathon Channel Tot
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
OFF	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
OFF	ON	ON	MIXED	MIXED	MIXED	MIXED	ON	OFF	OFF
OFF	ON	ON	MIXED	MIXED	MIXED	MIXED	ON	ON	OFF
STANDBY	ON	ON	STANDBY	STANDBY	STANDBY	STANDBY	OFF	ON	ON
STANDBY	ON	ON	RAMP_UP	RAMP_UP	RAMP_UP	RAMP_UP	RAMP_UP	ON	ON
STANDBY	ON	ON	RAMP_DW	RAMP_DW	RAMP_DW	RAMP_DW	RAMP_DW	ON	ON
STANDBY	ON	ON	MIXED	MIXED	MIXED	ON	ON	ON	ON
ON	ON	ON	ON	ON	ON	ON	ON	ON	ON

Table 11.4: Mapping between the Top Most node for integration and the detectors nodes

11.2.2 Using UML

At a later step the tables were converted into UML state diagrams as Figure 11.4 and Figure 11.5.

The transition arrows are of two types:

- **Black and labelled**
They represent the commands of the FSM type. The label is the COMMAND name and the transition is triggered by the operator or the FSM specific logic.
- **Gray and unlabelled**
They represent automatic transitions in the system. They take place automatically as response of changes in the internal status of the hardware.

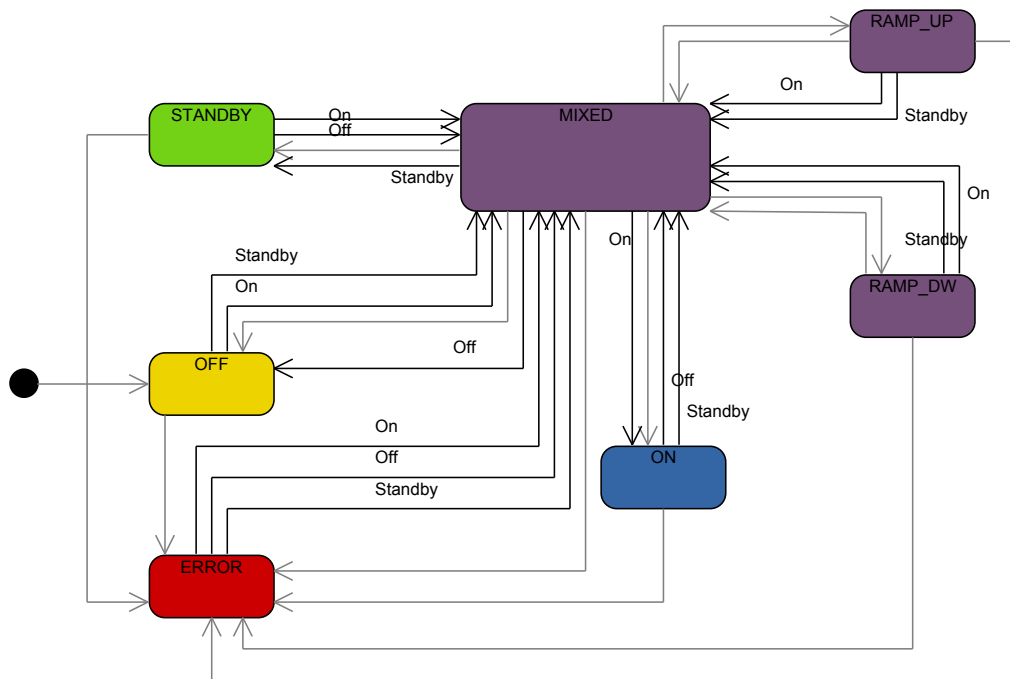


Figure 11.4: UML state model for totRpPot FSM type

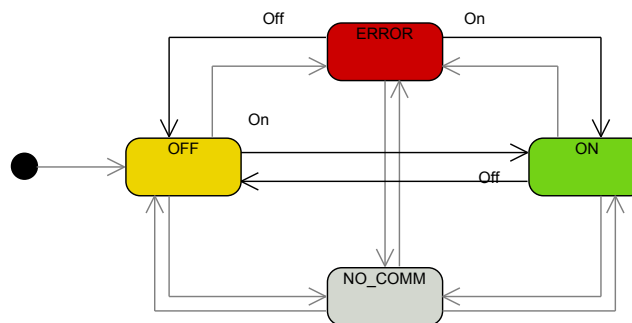


Figure 11.5: UML state model for FwWienerMarathonChannelTot FSM type

The major improvement of the UML formalization with respect to the tables that all the transitions of the FSM are expressed, giving a much better formalization of the failure sequences.

The general UML language is explained in [BRJ05]. For a more detailed review on real time references [Dou04] is a good citation.

11.3 FSM hierarchy tables and operation logic generation script

Figure 11.6 is an extract of the excel table used for defining the overlapping *FSM detector tree* and *FSM hardware tree*. It can be observed how both trees are overlapped of the final FSM hierarchy.

This table has three zones:

1. **Hierarchy level**

Here the name of the FSM node is written. Implicitly the column position (the number of empty cell before the first one filled) indicates the nesting in the hierarchy. A stack is maintained on the script with the last valid parent of each level. Also there is possibility to force the parent overriding the automatic nesting.

2. **Type of FSM node**

This part determines if it is a Control Unit (CU), Logic Unit (LU) or a Device Unit (DU). It also specifies the type of the framework.

3. **DCS Product Breakdown Structure entry**

This column is used for the analysis tool of Chapter 13, *Information theory analysis*.

The control/monitoring functions datapoints were previously generated by the pinout table and script. This FSM hierarchy table is automatically processed by another automatic script. The DU nodes generated from directly the PVSS logic names of the control/monitoring functions datapoints with some logic for mapping values into states. The LU and CU are not associated with any hardware. They are used to propagate the states and do implement some intelligence and interlocks.

The logic for assigning the initial state of the node and the transitions is defined in the FSM node type using PVSS scripting, not in each FSM node.

11.3. FSM hierarchy tables and operation logic generation script

logic name and FSM name								parent override	FSM type: CU, DU or LU	PBS
level										
A	B	C	D	E	F	G	H			
tot_Rp								CMS_TOTEM	CU:tot_Rp_Sv	E.03.01
tot_Rp_Hv										E.03.99
tot_Rp_CaenModu01									DU:FwCaenBoardSY1527Rp	E.03.01
tot_Rp_Lv										E.03.99
tot_Rp_WienMara01									DU:FwWienerMarathonRp	E.03.02
tot_Rp_WienMara02									DU:FwWienerMarathonRp	E.03.02
tot_Rp_WienMara03									DU:FwWienerMarathonRp	E.03.02
tot_Rp_WienMara04									DU:FwWienerMarathonRp	E.03.02
#	tot_Rp_Vme01									E.03.03
#	tot_Rp_Vme01_Temp									E.03.03
#	tot_Rp_Vme01_Fan01									E.03.03
#	tot_Rp_Vme02									E.03.03
#	tot_Rp_Vme02_Temp									E.03.03
#	tot_Rp_Vme02_Fan01									E.03.03
tot_Rp_Cooling								CU:tot_RpCaV	E.03.99	
tot_Rp_CoolPlant								DU:FwCaVPlantDU	E.03.06	
tot_Rp_CoolPlantLoop01								DU:FwCaVLoopDU	E.03.06	
tot_Rp_CoolPlantLoop02								DU:FwCaVLoopDU	E.03.06	
tot_Rp_CoolPlantLoop03								DU:FwCaVLoopDU	E.03.06	
tot_Rp_CoolPlantLoop04								DU:FwCaVLoopDU	E.03.06	
tot_Rp_CoolPlantLooploop01								DU:FwCaVLoopDU	E.03.06	
tot_Rp_CoolPlantLooploop02								DU:FwCaVLoopDU	E.03.06	
tot_Rp_CoolPlantLooploop03								DU:FwCaVLoopDU	E.03.06	
tot_Rp_CoolPlantLooploop04								DU:FwCaVLoopDU	E.03.06	
tot_Rp_Environment								CU:tot_GeEnvironment	E.03.99	
tot_Rp_CanBus								CU:tot_GeCan	E.03.99	
tot_Rp_CanBus08								CU:tot_GeCanBus	E.03.05.01	
tot_Rp_CanBus09								CU:tot_GeCanBus	E.03.05.01	
tot_Rp_Elmb								CU:tot_GeElmb	E.03.99	
tot_Rp_Elmb17								CU:tot_GeElmbNode	E.03.05.02	
tot_Rp_Elmb18								CU:tot_GeElmbNode	E.03.05.02	
tot_Rp_Elmb19								CU:tot_GeElmbNode	E.03.05.02	
tot_Rp_Elmb20								CU:tot_GeElmbNode	E.03.05.02	
tot_Rp_Elmb21								CU:tot_GeElmbNode	E.03.05.02	
tot_Rp_Elmb22								CU:tot_GeElmbNode	E.03.05.02	
tot_Rp_Elmb23								CU:tot_GeElmbNode	E.03.05.02	
tot_Rp_Elmb24								CU:tot_GeElmbNode	E.03.05.02	
tot_Rp_45								CU:tot_RpSide	E.03.99	
tot_Rp_45_220								CU:tot_RpStation	E.03.99	
tot_Rp_45_220_fr								CU:tot_RpUnit	E.03.99	
tot_Rp_45_220_fr_tp								CU:tot_RpPot	E.03.99	
tot_Rp_45_220_fr_tp_DssTemp								DU:tot_FwAiRpDss	E.03.09	
#	tot_Rp_45_220_fr_tp_Lvdt							DU:	E.03.08	
#	tot_Rp_45_220_fr_tp_Reso							DU:	E.03.08	
#	tot_Rp_45_220_fr_tp_MicrIn							DU:	E.03.08	
#	tot_Rp_45_220_fr_tp_MicrOut							DU:	E.03.08	
tot_Rp_45_220_fr_tp_Hv								DU:FwCaenChannelRpHv	E.03.01	
tot_Rp_45_220_fr_tp_LvG								DU:FwWienerMarathonGroupRp	E.03.02	
tot_Rp_45_220_fr_tp_LvA								DU:FwWienerMarathonChannelRp	E.03.02	
tot_Rp_45_220_fr_tp_LvD								DU:FwWienerMarathonChannelRp	E.03.02	
#	tot_Rp_45_220_fr_tp_Temp01							DU:	E.03.05.03	
#	tot_Rp_45_220_fr_tp_Cool01							DU:	E.03.05.03	
#	tot_Rp_45_220_fr_tp_Cool02							DU:	E.03.05.03	
#	tot_Rp_45_220_fr_tp_Cool03							DU:	E.03.05.03	
#	tot_Rp_45_220_fr_tp_Cool04							DU:	E.03.05.03	
#	tot_Rp_45_220_fr_tp_Vacc01							DU:	E.03.05.05	
#	tot_Rp_45_220_fr_tp_RadLaas							DU:	E.03.05.04	
#	tot_Rp_45_220_fr_tp_RadCmrp							DU:	E.03.05.04	
#	tot_Rp_45_220_fr_tp_RadRem							DU:	E.03.05.04	
#	tot_Rp_45_220_fr_tp_RadBpw							DU:	E.03.05.04	
#	tot_Rp_45_220_fr_tp_RadTemp							DU:	E.03.05.04	
#	tot_Rp_45_220_fr_tp_RadRI							DU:	E.03.05.04	
tot_Rp_45_220_fr_tp_PIO1								DU:	E.03.99	
#	tot_Rp_45_220_fr_tp_PIO1_Temp01							DU:	E.03.04	
#	tot_Rp_45_220_fr_tp_PIO1_Volt01							DU:	E.03.04	
#	tot_Rp_45_220_fr_tp_PIO1_Curr01							DU:	E.03.04	
#	tot_Rp_45_220_fr_tp_PIO2							DU:	E.03.99	
#	tot_Rp_45_220_fr_tp_PIO2_Temp01							DU:	E.03.04	
#	tot_Rp_45_220_fr_tp_PIO2_Volt01							DU:	E.03.04	
#	tot_Rp_45_220_fr_tp_PIO2_Curr01							DU:	E.03.04	
#	tot_Rp_45_220_fr_tp_PIO3							DU:	E.03.99	
#	tot_Rp_45_220_fr_tp_PIO3_Temp01							DU:	E.03.04	
#	tot_Rp_45_220_fr_tp_PIO3_Volt01							DU:	E.03.04	
#	tot_Rp_45_220_fr_tp_PIO3_Curr01							DU:	E.03.04	
#	tot_Rp_45_220_fr_tp_PIO4							DU:	E.03.99	

Figure 11.6: FSM hierarchy table

11.4 LHC status and handshake

The handshake mechanism among the experiment DCS and LHC was first explored by me in TOTEM. After this clarification of this logic the development of this handshake and the exchange of signals as in Section 6.3 was done by Oliver Homes including the BPM, BLM, LHC mode,...

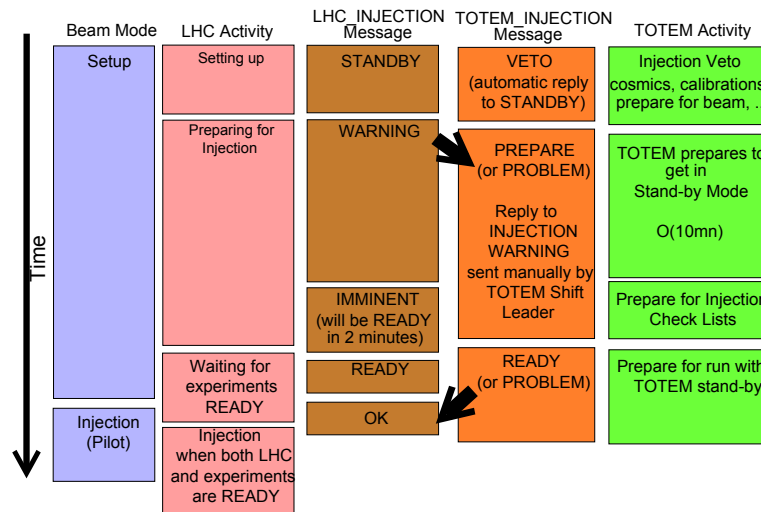


Figure 11.7: Sequence diagram for the LHC injection handshake

However the operation of this handshake has to be studied in further detail while integrated with CMS.

11.5 New datapoint types

As explained in Section 5.4.2 all the PVSS programming model is based on datapoints. They are used for storing the measurements and the configuration. The JCOP framework also uses this philosophy, and all the OPC and trending configuration among many others is stored in special datapoint types.

Following this reasoning the TOTEM DCS needs specific datapoints for the particular hardware. Examples of this specific hardware are the DCU, RADMON sensors, pressure sensors, Roman Pots motorization values,... They will include calibration data for the vacuum sensors and timing issues for the RADMON sensors and any other additional value needed.

A proposal of these datapoints is given in Table 11.8, Table 11.9 and Table 11.10

11.6 Critical actions

11.6.1 High rates versus Beam Loss Monitors

High rates in the detector are dangerous. If the rates detected by the DAQ system goes over a threshold, the pots are automatically retracted. Also the BLM can shutdown the beam. It is necessary through experimental data to determine what threshold is exceeded first (is more sensitive) and adapt consequently the logic.

DCU
TemperatureA.Value
CurrentA.Original
CurrentA.Offset
CurrentA.Slope
CurrentA.Value
VoltageA.Original
VoltageA.Offset
VoltageA.Slope
VoltageA.Value
TemperatureB.Value
CurrentB.Original
CurrentB.Offset
CurrentB.Slope
CurrentB.Value
VoltageB.Original
VoltageB.Offset
VoltageB.Slope
VoltageB.Value
TemperatureC.Value
CurrentC.Original
CurrentC.Offset
CurrentC.Slope
CurrentC.Value
VoltageC.Original
VoltageC.Offset
VoltageC.Slope
VoltageC.Value

Figure 11.8: PVSS datapoint type for the DCU

RADMON ISC
Ntc.Delay
Ntc.Voltage
Ntc.Current
Ntc.Value
Cmrp.Delay
Cmrp.Voltage
Cmrp.Current
Cmrp.Value
Bpw.Delay
Bpw.Voltage
Bpw.Current
Bpw.Value
Laas.Delay
Laas.Voltage
Laas.Current
Laas.Value
Rem.Delay
Rem.Voltage
Rem.Current
Rem.Value

Figure 11.9: PVSS datapoint type for the RADMON sensors

Vacuum / Pressure
CurrentA.Original
CurrentA.Offset
CurrentA.Slope
CurrentA.Value

Figure 11.10: PVSS datapoint type for the Vacuum sensors

11.6.2 Cooling on and vacuum loss versus shutdown all electricity and sensors powering

The speed of loosing the vacuum of 3 pots is unknown. This can happen because of a power cut in the vacuum pump, or because a mechanical damage.

If the cooling is ON (it is common for one full station, 6 pots) and vacuum is lost, there is risk of condensation and therefore shortcuts. For that reason the power of the pot needs to be disconnected.

Also if the pot is very close to the beam, the difference of pressure can make the pot window bend into the beam.

In the case that the vacuum is completely lost in the order of magnitude of the reaction time of the DCS, no warning/confirmation will be sent to the operator. In this scenario when any vacuum sensor is out of range, even slightly, the pot should be retracted and unpowered.

However if the vacuum sensor stays stable and in an acceptable range the operation could be resumed.

11.6.3 Data taking scenarios with several RP not working

If a pot is out of order, still many other are operative. The operator can stop data taking, rearrange the FSM logic, and resume operation providing an OK status to the top node.

Further researches needs to be done for the definition of a fault tolerant logic, so that the error status of a single pot can be acknowledged without propagating an error to the CMS_TOTEM node.

Operational logic

The *majority* feature of the JCOP framework is not valid because the ERROR state correspond to specific combination of pots. If 3 of the same unit are non operational operation can continue. However if those 3 pots are in the same position (e.g. horizontal) and in the same sector, the particles trajectories cannot be calculated anymore and maintenance is needed.

Integration with the motorization

The formalization of the movement control of the Roman Pots due it close interaction with the LHC required the creation a set of operational and engineering documents.

The initial operational requirements are described in [ORL06]. A discussion [DJL⁺08] took place with AB/CO in order to clarify the operation within the Collimator Control Architecture. This architecture is described in [AAG⁺05] and the application the software for the LHC collimators and movable elements has been specified too in [RM08].

Further constraints on the motorization control come from the interaction with the DCS and the the CCC. They were formalized using UML into [PLDR08]. A protocol based on DIM has been specified on [Dut08] to communicate all the Roman Pots motorization control subsystems: PXI, FESA and DCS.

12.1 Description of the collimators control architecture

12.1.1 Low Level control

The low level control is based on the same PXI hardware that is chosen for the collimation control. This decision was taken for being able to prove more easily to LHC protection committee that the system was satisfactory enough.

The specific differences in case of the Roman Pot control, are that only one extended crate is used for both motor control and position surveillance. Secondly the modules to read the Resolvers are a simplified version and only allow for an acquisition frequency of about 2 Hz. This is in contrary to the collimator modules, which are read with a frequency of 100 Hz. However the LVDT's are read at a speed of 100 Hz, in the same way as the collimators. The speed of the motors is regulated to match the readout speed of the sensors, and provide a satisfactory protection scenario.

Those differences in the hardware were done to reduce costs but introduced a problem of spares. Now it is not longer possible to have common spares program with the LHC collimators, and an additional system in common with ATLAS ALFA detector needs to be purchased.

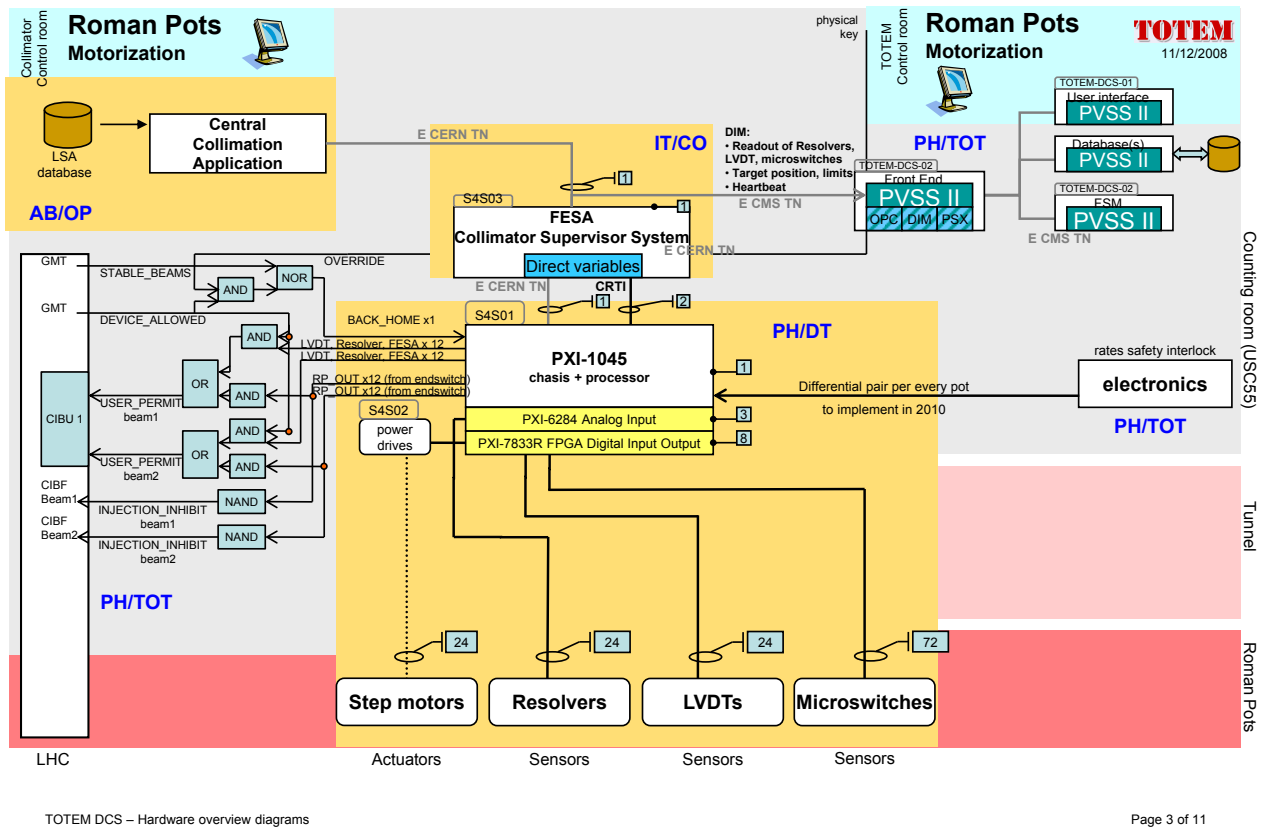


Figure 12.1: Roman Pots motorization control overview; from [Luc08c]

12.1.2 Collimator Supervisory System (CSS) level

This subsystem has the role to apply into the PXI front-end the settings and limits specified in the CCA. It also monitors the current position and generate some interlocks.

TOTEM needs to adapt the CSS because the number of motors and LVDT's do not match a standard collimator. Similar adaptations are also needed for other special collimation elements (scrapers, TCDQ, ...). For this reason TOTEM will adopt the same philosophy as the one made for these special elements using the same semantics in the CSS interface as those special elements. Extensions at a later stage such as a fast positioning based on the detector readout signals are feasible.

This specific middleware FESA software runs on a dedicated PC-gateway machine. This machine is located close to the PXI subsystem in USC55 in a location where the timing distribution is available, see Appendix A, *Locations*.

12.1.3 Central Collimation Application (CCA) level

The Roman Pot settings are fully controlled from the CCC by the standard Central Collimation Application, which represents them as a special family of collimators. The characteristic of the Roman Pot family is that these devices will not make any movements during the ramp. Under all conditions, except the machine mode STABLE_BEAM or in modes without beam, the position limits impose that the Roman Pots stay in the retracted position.

As soon as the machine modes changes to STABLE_BEAM, the position limits will relax and the Roman Pots can be moved towards the beam up to the new limit (see Figure 12.1.3).

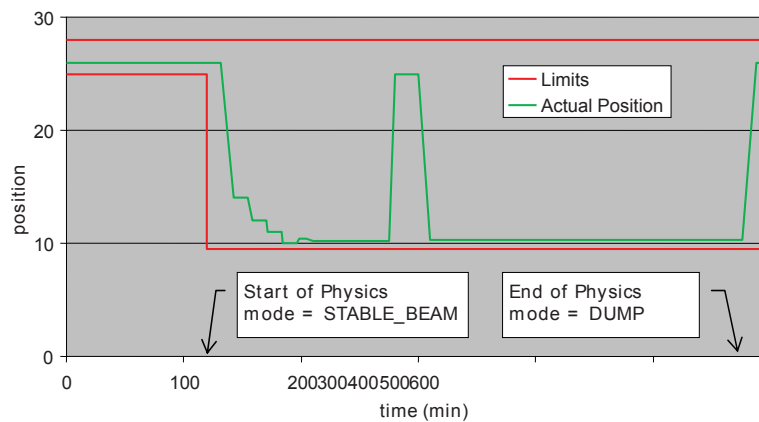


Figure 12.2: Example of Roman Pot movements during an LHC cycle

All settings and modifications of the Roman Pot position are done from the CCC through the collimator application on request by the TOTEM Control Room. Possibly when good positions are found, these positions can be saved for the next fill.

12.2 Use cases

12.2.1 Adjust Roman Pot position for data taking

The TOTEM DCS operator based on the information of the FESA and PXI systems decides to move the Roman Pots and therefore sets the MOVEMENT_INHIBIT signal to false (hardware push-button like). The operator then communicates to the CERN Control Center the list of positions to be given to each Roman Pot. The CCC operator types-in these values and associated warning and critical limits are loaded from the LSA database. These positions and limits are then sent to the FESA system, and can be viewed by the DCS operator as well.

The PXI system then provides power to the Roman Pot motor to be moved, updates related MOTOR_POWER_ON signal(s) and adjusts the motion speed of each Roman Pot to meet the requested positions. The actual position is permanently refreshed and reported by the FESA system to the DCS and the CCC Application. The process of position adjusting can be repeated multiple times on the DCS operator initiative.

Once all the Roman Pots are in an acceptable position, the DCS Operator switches the MOVEMENT_INHIBIT to true in order to prevent any data taking position to be further modified.

Finally, the TOTEM operator could store the current Roman Pot positions for future usage.

12.2.2 Adapt Roman Pots positions to failures and harmful context

This Use Case shows how the FESA and PXI subsystem shall adapt the Roman Pots positions when the latter are confronted to failures and dangerous conditions such as vacuum loss, high rates, radiations and other issues that could happen and cause damage to the TOTEM detectors and to the accelerator.

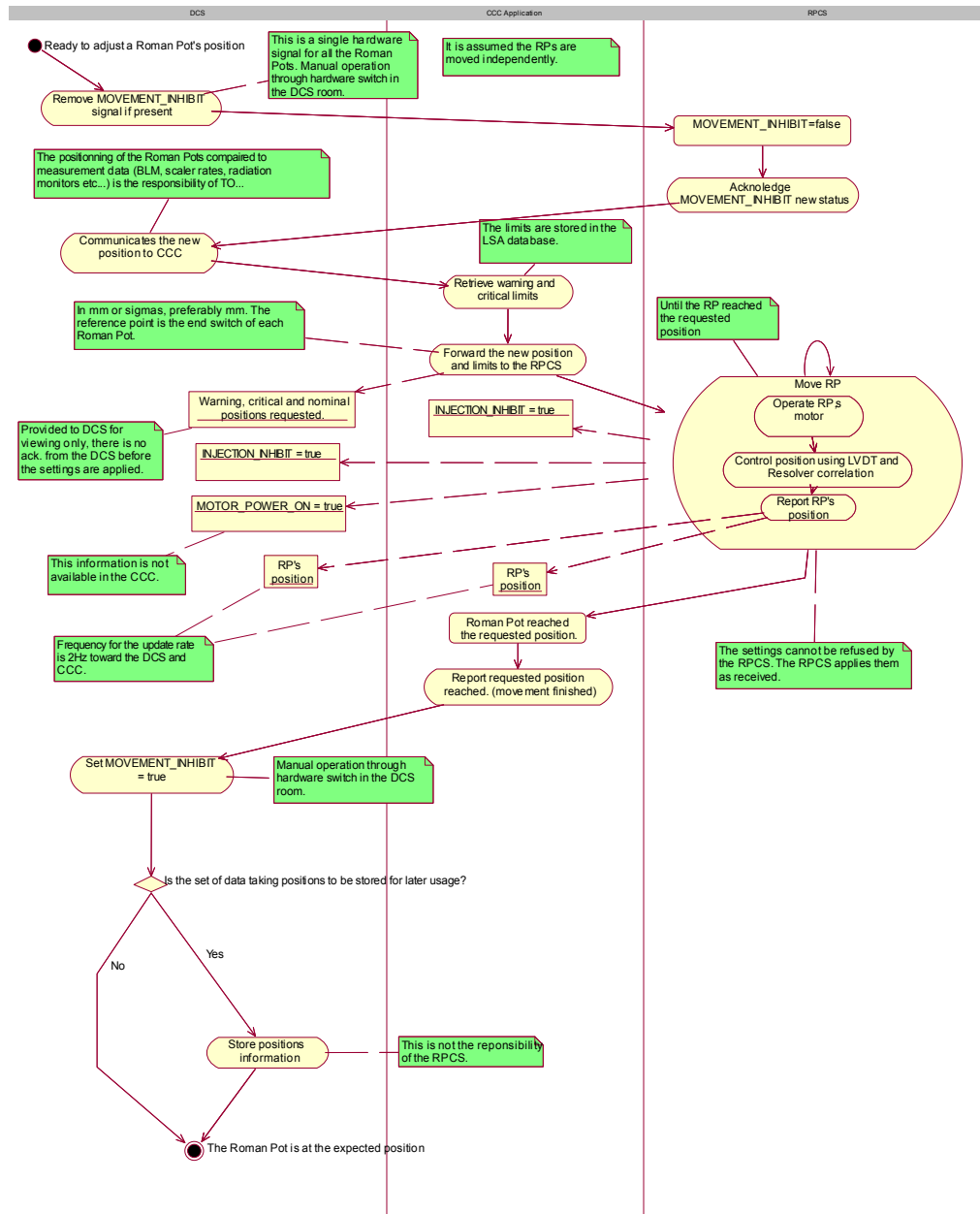


Figure 12.3: RP motorization use case for adjusting position

The information is provided to the DCS, who below a certain level of severity of the issue detected, could simply adapt the positions of the Roman Pots involved using the standard position adjustment sequence. This sequence of actions can be fully automatic or with the intervention of the TOTEM operator.

In case a predefined severity threshold is exceeded, the FESA subsystem takes over and decides whether an emergency extraction shall be combined with a beam dump request in case the Roman Pots cannot be extracted before damages occur. The Use Case ends when the Roman Pots are in a safe position.

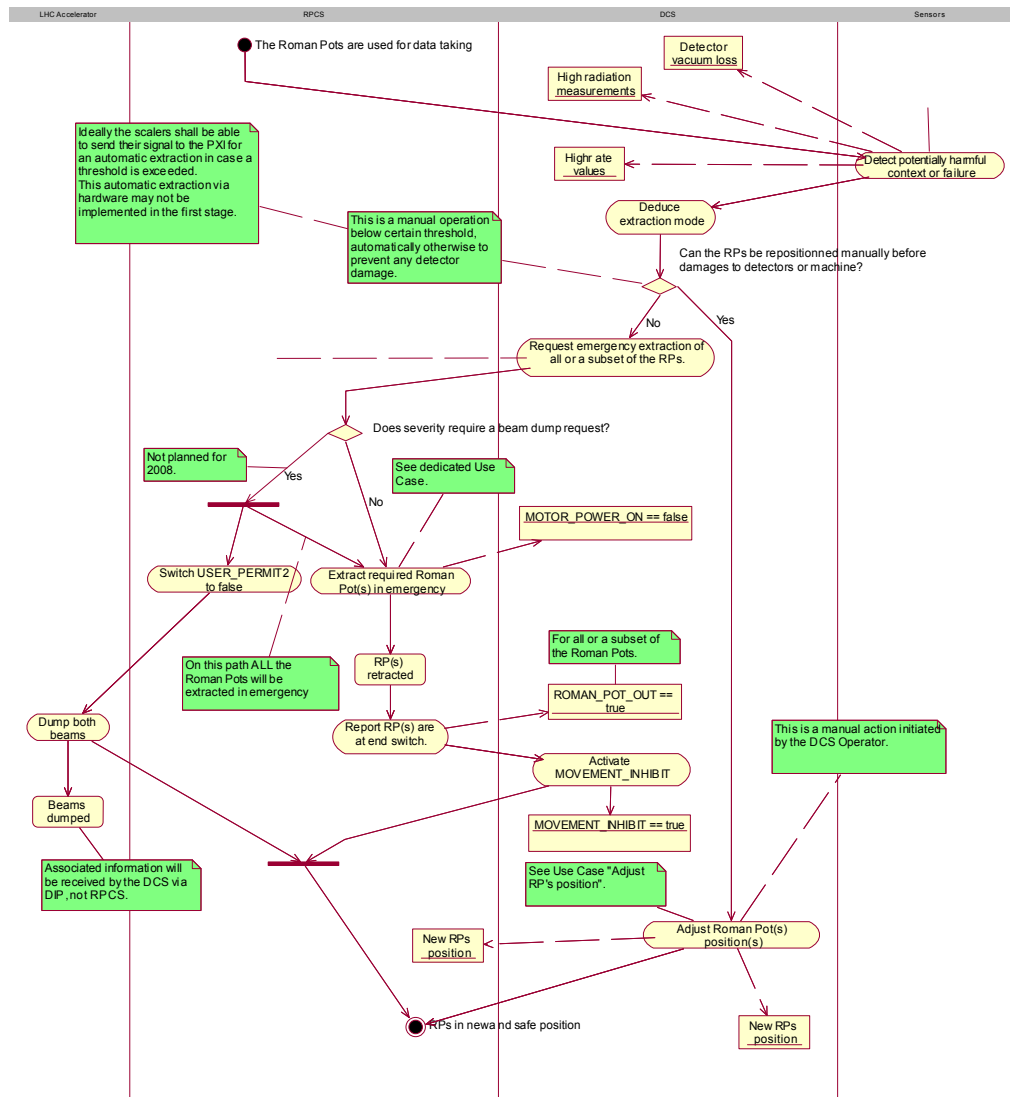


Figure 12.4: RP motorization use case for new pot position request

12.2.3 Extract Roman Pot(s) in emergency

This simple Use Case describes how to remove the Roman pots from the beam line when necessary. This Use Case can be triggered manually or automatically depending on the context.

I requested a manual hardware switch in the DCS room to trigger manually an emergency extraction of all the Roman Pots at once as a last chance mechanism. It is based in the agreement for the electrical distribution of all the subsystems DCS, FESA and PXI are powered by the CMS UPS, meanwhile the motors coils is done with the CMS diesel.

When this *PANIC* button is pressed the power inside the rack of the motor coils is disconnected. This bypass all the control levels and simply relies on the mechanical springs to extract the Roman pots from the beams. Even any if the DCS, FESA and PXI systems is responding there is still an option to get out the Pots of the beam.

If possible, the related signals are adjusted and the MOVEMENT_INHIBIT is set to true to prevent any unexpected movements of the Roman Pots.

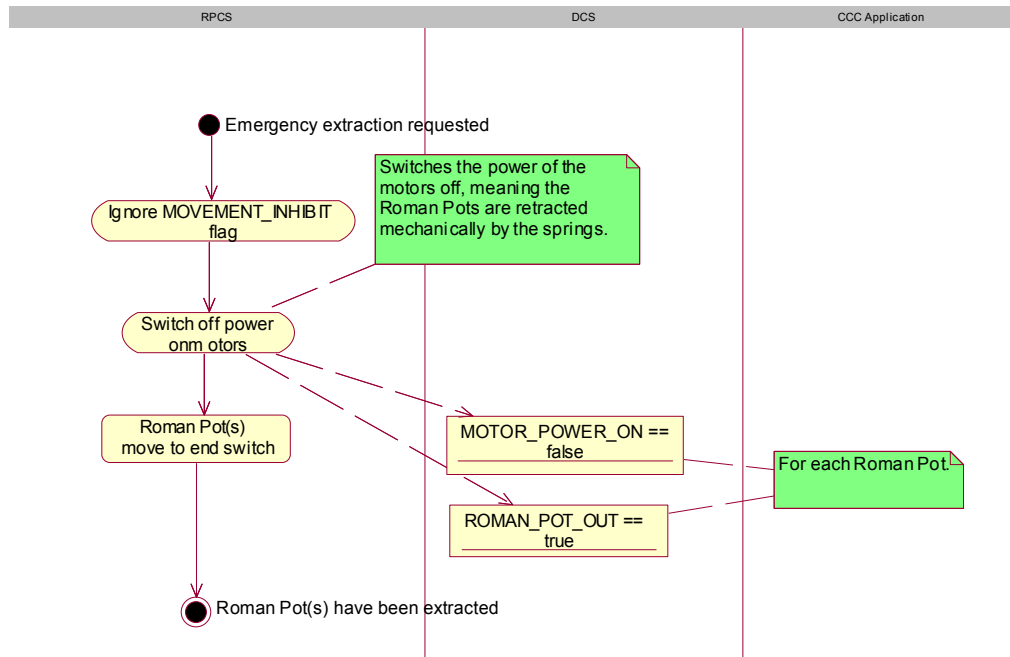


Figure 12.5: RP motorization use case for emergency retraction

12.3 Interactions between the motorization control and the DCS

For the DCS this communication is mostly a read only communication. Only a heartbeat signal is refreshed and some emergency signals for extracting the pots using the motors can be triggered.

At the very beginning the possibility of having the motorization information through the Central Collimation Application was considered, keeping all the movement problematic as a black box. But this option was abandoned, the problems were to extend that critical application for the LHC, and having so many systems stacked one on top of the other was increasing excessively the risk. The delays in transmitting the values and ensuring the communication among so many different networks could become an issue. So the approach was to access directly the FESA subsystem, rather than the PXI subsystem for having a reasonable protection scenario.

Another document named *Interface Control Document* [Dut08], was generated to map all the previous signals into the DIM protocol and specify the types of hardware signals between the FESA and PXI subsystem.

This interface and DIM were reused also for the communication between the DCS and FESA subsystem. The other alternative would had been to use CMW (Common MiddleWare); the AB technology to communicate the CCA to the FESA subsystem.

The problem of using CMW was that this technology a much less tested in the PVSS domain. Additionally the LHC logic for operating the pot is based on emulating collimators. In this way, the two vertical pots of one unit are considered as a collimator, and the horizontal one as an independent collimator. This allows to the CCA to detect some bad positions.

Also the LHC it has a different naming convention for the pots and a different ordering. The TOTEM FESA subsystem has the role to do all the proper mappings and grouping for offering two control interfaces. One using CMW for the LHC and another using DIM internal to TOTEM.

Information theory analysis

13.1 Introduction

When developing this kind of system there is need to have some estimations on the response time, archiving needs and dataflow rates. In the current state of the art for High Energy physics experiments this previous analysis is never done. It is considered that adding more resources the problems will eventually be solved without considering architectural flaws.

For that reason I developed the tool of Section 13.2, *Tool for the calculations* for helping with this analysis, and being able to quantify the DCS in terms of information exchange. Those numbers will need to be corrected by an experimental constant depending on when usual operation is fulfilled. However, the orders of magnitude will remain the same.

The usual CAN timing analysis described in [DBBL07] does not suits well to the DCS scenario. That kind of analysis suits very well to asynchronous transmissions it is need to have a very clear schedule of the situations to check, with very fine granularity.

However, in the ELMB readout case all the normal operation is done using synchronous events. This action on the ELMB is triggered on the bus level by the ‘sync interval’ [Par96]. This signals tells all the devices to flush into the bus all the acquired data.

From this point of view, if all the data acquired by all the ELMB in the bus is transmitted faster than the interval between to SYNC commands, the bus is not saturated an schedulability is possible. If this condition does is not satisfied, data will be lost or commands will not be processed.

This is a quite rough approach, but still valid, saves the need of a full detail analysis taking into account every individual command and response. Even more, in the DCS system, the SYNC interval will be one order of magnitude above the time needed to the one time needed to transmit all the data in the worst case as shown in Section 13.7.2, *Values for the RP ELMB box for Temperature and Vacuum*.

An special case is the RADMON readout. In this system no SYNC command is sent. All the data is transmitted on demand using the CANopen SDO commands. The only problem of schedulability in the bus would come if two different threads in the same computer try to use that bus almost at the same time. This can be avoided considering the bus as a shared resource, and implementing a suitable

locking mechanism that prevents sending more commands while waiting for an answer of a previous command.

For the analysis proposed in [DBBL07] there is a commercial tool named ‘Volcano Network Architect’ [Gra09] for automotive industry. Profiting of the FSM hierarchy formalization as EXCEL tables the tool described in Section 13.2, *Tool for the calculations* was developed. It allows to check the CAN timing parameters, archiving needs, dataflow rates, memory consumption estimations, overall reaction times,...

The rest of the system is evaluated checking the OPC servers refresh parameters, and the usual FSM processing times, as shown in Section 13.7, *Analysis of the response time*.

13.2 Tool for the calculations

I have developed a specific tool for the calculation of the data rate of information exchanged among all the hardware component. It uses the FSM excel table of Section 11.3, *FSM hierarchy tables and operation logic generation script* and according to the Product Breakdown Structure tag it assigns different values for the different calculation factors as seen in Table 13.1.

Property	Value	Explanation
Id	E.03.05.03	The identifier in the PBS.
Description	EnvironmentMonitors - Temperature	A text description of the PBS identifier.
Color	#76923C	Color of the PBS entry type in the FSM.
Count	120	The number of times this entry is used. It is calculated and stored after processing the FSM tree.
Information Chunk	4.00 Bytes	The information size of the value transmitted in the readout.
Variation Probability	1	The probability of changing the value between two readout intervals.
Variation Accumulated	0	The probability of changing the value but considering the nodes below. Only useful in FSM nodes.
Archiving Frequency	00:05:00	This value is multiplied by the probability of change and the information size to calculate the archiving values.
Archiving Node	4.70 GBytes	Is the information that this node has to archive only by itself.
Archiving Accumulated	0.00 Bytes	Is the information archived by this node and its children.
Archiving Overhead	16.00 Bytes	Represents the overhead in the structure, metainformation and indices to store the InformationChunk in a database.
Readout Rate Frequency	00:00:00.5000000	This value is multiplied by the probability of change and the information size to calculate the readout values.
Readout Rate Node	37.50 Kbits/s	Is the information that this node is generating only by itself.
Readout Rate Accumulated	0.00 bits/s	Is the information generated by itself and the sum of all the generated in the children entries.
Readout Rate Overhead	8.00 Bytes	Represents the overhead of the InformationChunk in the communication protocol.
Time Send Response	00:00:00.1000000	Time needed to send a response through the physical communication channel.
Time Send Command	00:00:00.1000000	Time needed to send a command through the physical communication channel.
Time Execute	00:00:00.1000000	Time needed to execute a request.
Time Internal Update	00:00:00.1000000	Time needed to encode/decode a request.

Table 13.1: PBS configuration entry for Temperature Sensors

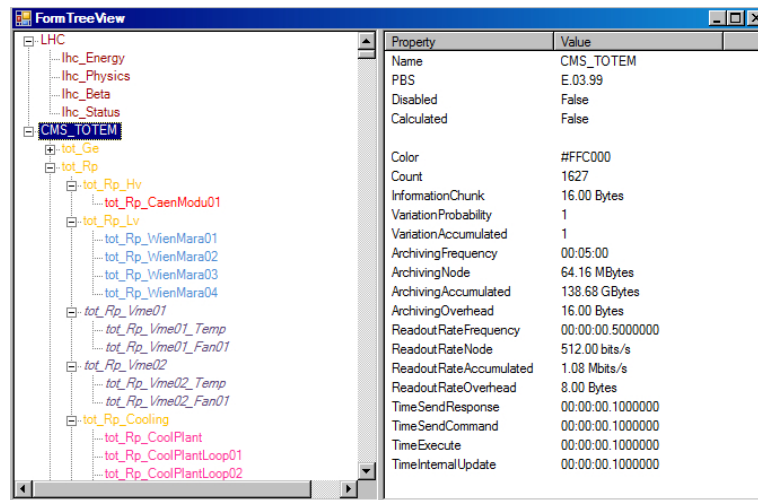
This tool is highly modular, and the process of calculation has three steps:

1. Parse the tables and build the FSM tree.

2. Assign and match the PBS entries in the tree.
3. Execute correspondent algorithm for the calculations.

Each algorithm is contained in a independent class that is dynamically loaded. It explores the FSM tree and the PBS items to update the proper information. Moreover, the PBS items are also dynamically constructed from the table, and an algorithm could generate new properties in the PBS entry during execution time.

The user interface can be seen in Figure 13.2. It generates the charts of this chapters as well as the summary table.



13.3 Orders of magnitude of the information exchanged among HW levels, nodes and buses

Table 13.2 gives a total overview of the DCS expected workload. Values are expressed as absolute and percentages.

The main result of this table is that ‘only’ 150 GB are needed for the TOTEM DCS archiving needs in the scenario of uninterrupted operation during 20 years. And the total data-flow of the whole system is slightly above 1 Mbit/s.

The data-flow results are not a CPU load estimation, but are directly related. Even if the data-flow increases up to 2 Mbits/s taking in consideration T1 detector, a usual computer of nowadays could handle the requirements of the whole TOTEM experiment. In principle they will be mapped into 4 computers, but this number can be increased if necessary.

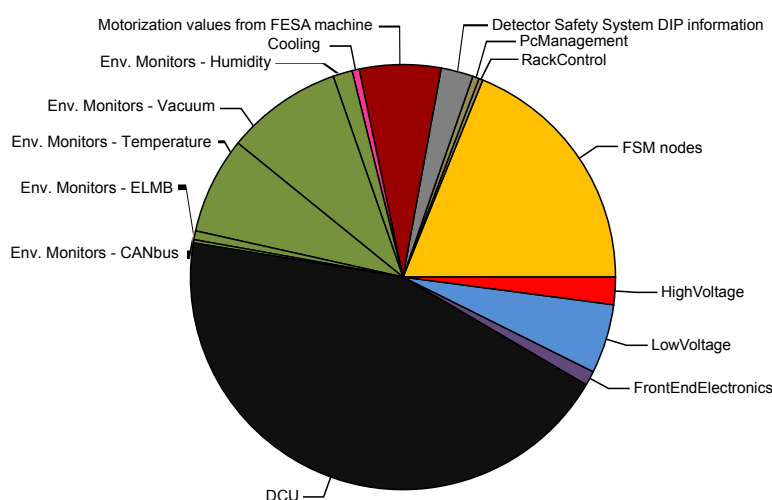
For the main supervisor of TOTEM, T1 and T2 no performance problem is expected. However the machine assigned to Roman Pots is studied in detail. If there is any problem of performance, two machines will be used, one per sector. The need is to follow the PBS decomposition of the detector as much as possible when assigning more machines.

The alternative could be to use one machine for the motorization and another for the Silicon Detector. Using distributed datapoints information can be exchanged among machines. Due to the fact that the status of the motorization and the Silicon Detector are coupled in the FSM level of the pot this will lead to having half FSM in each machine. But this is highly suboptimal, all the information has to be exchange among the two machines, and will not provide a performance improvement.

PBS	Description	Count	Count Percentage	Archiving Total	Archiving Percentage	Readout Total	Readout Percentage
E.03.01	HighVoltage	34	2.08 %	5.33 GBytes	3.82 %	42.50 Kbits/s	3.82 %
E.03.02	LowVoltage	86	5.27 %	8.08 GBytes	5.80 %	64.50 Kbits/s	5.80 %
E.03.03	FrontEndElectronics	18	1.10 %	1.13 GBytes	0.81 %	9.00 Kbits/s	0.81 %
E.03.04	DCU	720	44.12 %	67.67 GBytes	48.52 %	540.00 Kbits/s	48.52 %
E.03.05.01	Env.Monitors - Canbus	4	0.25 %	160.40 MBytes	0.11 %	1.25 Kbits/s	0.11 %
E.03.05.02	Env.Monitors - ELMB	11	0.67 %	441.10 MBytes	0.31 %	3.44 Kbits/s	0.31 %
E.03.05.03	Env.Monitors - Temp.	120	7.35 %	4.70 GBytes	3.37 %	37.50 Kbits/s	3.37 %
E.03.05.04	Env.Monitors - Vacuum	144	8.82 %	5.64 GBytes	4.04 %	45.00 Kbits/s	4.04 %
E.03.05.05	Env.Monitors - Humidity	24	1.47 %	1.50 GBytes	1.08 %	12.00 Kbits/s	1.08 %
E.03.05.06	Env.Monitors - Pressure	0	0.00 %	0.00 Bytes	0.00 %	0.00 bits/s	0.00 %
E.03.06	Cooling	9	0.55 %	1.41 GBytes	1.01 %	11.25 Kbits/s	1.01 %
E.03.07	Gas	0	0.00 %	0.00 Bytes	0.00 %	0.00 bits/s	0.00 %
E.03.08	Motorization values	101	6.19 %	15.82 GBytes	11.34 %	126.25 Kbits/s	11.34 %
E.03.09	Detector Safety System	40	2.45 %	6.27 GBytes	4.49 %	50.00 Kbits/s	4.49 %
E.03.10	RackControl	8	0.49 %	1.25 GBytes	0.90 %	10.00 Kbits/s	0.90 %
E.03.11	AccessControl	0	0.00 %	0.00 Bytes	0.00 %	0.00 bits/s	0.00 %
E.03.12	UserInterface	0	0.00 %	0.00 Bytes	0.00 %	0.00 bits/s	0.00 %
E.03.13	PcManagement	5	0.31 %	802.00 MBytes	0.56 %	6.25 Kbits/s	0.56 %
E.03.14	RunControl	0	0.00 %	0.00 Bytes	0.00 %	0.00 bits/s	0.00 %
E.03.99	FSM nodes	308	18.87 %	19.30 GBytes	13.84 %	154.00 Kbits/s	13.84 %
E.03	SUM	1632	100.00 %	139.47 GBytes	100.00 %	1.09 Mbits/s	100.00 %

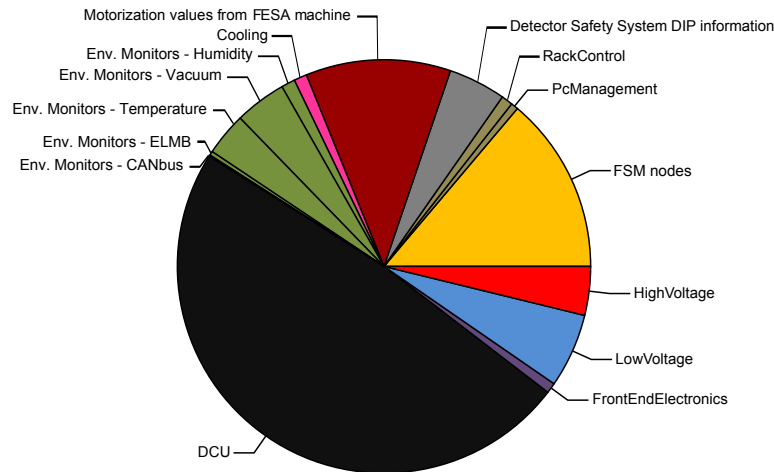
Table 13.2: DCS items distribution

Having more OPC servers connected to the same hardware is not a problem. For example, an OPC server will be installed in each computer. That OPC server will only request information about the specific board it has configured in PVSS. DIP, DIM and PVSS distributed datapoints behaves in the same fashion; only that specific value is transmitted to any number of clients. The problem comes when using ELMB, they can only be read by a computer because of the CAN/USB conversion. In our current ELMB layout, there enough empty buses available for any reconfiguration, the only issue is that some additional ELMB would be needed.



13.4 Data generation for the DCS

Figure 13.4 provides a first approximation of the data-flow rate in the DCS.



The fronted data rates are estimated as follow:

- The HV and LV servers it is based on the OPC client update rate.
- The sensors data-flow is based in the syncInterval parameter of the ELMB.
- The DIP values of the DSS will be considered as the probability of change in the ELMB temperature sensor.
- DIM values for the motorization are specified to be 2 Hz/s.
- For the PVSS distributed datapoints (such as the ones generated by CMS central time) the data-flow will be an experimentally estimated value. How is a very low set of datapoints the differences on the estimation will not be significant.

This calculation is implemented using C# in the tool as:

```
double totalbits = node.Value.InformationChunk.Value + node.Value.ArchivingOverhead.
    Value;
double datarate = (node.Value.VariationProbability.Value * totalbits) / node.Value.
    ReadoutRateFrequency.Value.TotalSeconds;
```

13.5 Archiving the data

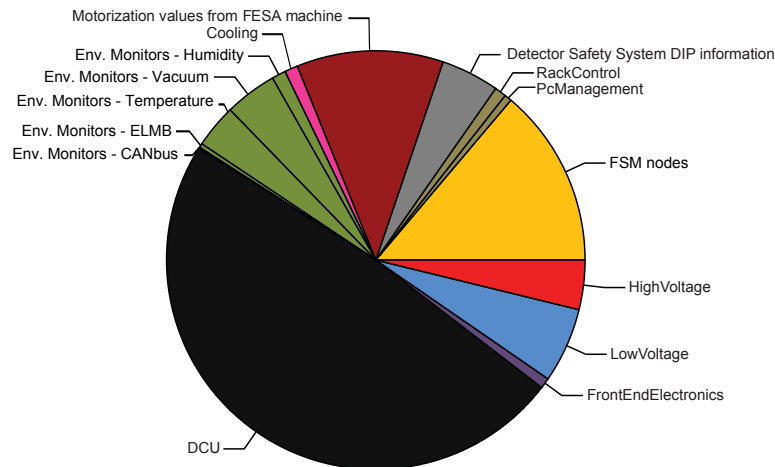
The archiving is done pooling the values of the PVSS datapoint at intervals. If the value is different or there is an specific request the value is recorded.

The estimations shows on Figure 13.5 consider that always there is need of a new archiving. The relevant result is the distribution of the values. Half of the requirements or archiving will come from the DCU values that had been by the DAQ system. When DAQ system will be fully operational it must be decided what system (DAQ or DCS) must archive those currents and temperatures. If that data is relevant for offline analysis, it must be stored together with the tracks already stored by the DAQ.

Also the FSM status will be archived. This is not strictly needed for historical analysis; with the scripts logic it cab be reconstructed form the hardware datapoints. However, if there is need of debugging debugging it must be stored as an standalone entity. After a software version change it would be much difficult to track the old logic.

This calculation is implemented using C# in the tool as:

```
double totalbits = node.Value.InformationChunk.Value + node.Value.ArchivingOverhead.
    Value;
double datarate = (node.Value.VariationProbability.Value * totalbits) / node.Value.
    ArchivingFrequency.Value.TotalSeconds;
```



Notice that using a programming language for expressing requirements or an specification is an established practice, as in [Gro08].

13.6 Memory for execution

Each one of the production nodes is equipped with 1 GB of RAM. The consumption having Windows XP SP3, PVSS and OPS servers running is less than 190 MB.

In a final DCS most of the memory is needed for the FSM logic. DU and LU have a very small footprint, but each CU is estimated in about 5 MB.

As a first approximation of the memory usage for the Roman Pots node (me most 'busy' node) we can take 200 MB as the OS+utilities baseline. Count the number of *Control Units* foreseen (24 pots + 8 units + 4 stations + 2 sector + 1 detector) giving a total of 39 CU that translate into another 200 MB for FSM operation.

In total the memory consumption is estimated to be below 500 MB. In this situation the machines are no doing any pagination and the performance results should be quite homogeneous under different loads, given enough CPU power.

Even more, if pagination happens and it still match the response time requirements it should not be a problem.

User interface has to be considered independently. It consumes also quite big amounts of RAM. Having the User Interface running on an independent node isolates many performance problems.

13.7 Analysis of the response time

13.7.1 Model of the CAN bus

SyncInterval Validation

The first step is to calculate the time need to do a full readout of the bus.

First, some definitions are needed:

AdcRate = Time spent on the ADC conversion of one channel	1/s
NumberChannels = Number of channels used in one ELMB	
FrameLenght = Lenght of the data frame in bits in the CAN protocol	bit
BitToBauds = Statistical value to reflect the conversion between bits and bauds	baud/bit
NumberElmb = Number of ELMB attached to the bus CAN	
CanSpeed = Speed of the bus CAN	baud/s

After that two parameters can be defined

BaudsInElmbReadout = NumberChannels * FrameLenght * BitToBauds	* * bit * baud/bit = baud
BaudsInCanReadout = BaudsInElmbReadout * NumberElmb	baud
AdcInterval = (1/AdcRate) * 1000	ms
ElmbInterval = NumberChannels * AdcInterval	ms

So the final value can be expressed such as

$$\text{BusExtractionInterval} = (\text{BaudsInCanReadout} / \text{CanSpeed}) * 1000 \quad \text{baud} / (\text{baud} / \text{ms}) = \text{baud} * \text{ms} / \text{baud} = \text{ms} \quad (13.1)$$

(13.2)

3 requisites must be satisfied to be a valid SyncInterval

- *BusExtractionInterval must be less than ElmbInterval*
The bus has to be faster than the ELMB generation of data.
- *SyncInterval must be bigger than BusExtractionInterval*
Is is reasonable to wait the time needed for a full readout an ELMB before triggering a new one.
If not the bus utilization would be over 100%.
- *SyncInterval must be less than ElmbInterval*
For not loosing measured values it is necessary to trigger and transmit the values faster than they are being generated in the ELMB.

All those 3 requisites can be resumed in the following condition

$$\text{SyncInterval} \in (\text{BusExtractionInterval}, \text{ElmbInterval}) \quad (13.3)$$

So the bus occupancy can be defined as:

$$\text{Occupancy} = (\text{BusExtractionInterval} / \text{SyncInterval}) * 100 \quad (13.4)$$

13.7.2 Values for the RP ELMB box for Temperature and Vacuum

This is the worst case scenario in TOTEM layout. It is a bus configured at 500000 bauds/s with 4 ELMB with 64 channels configured at a ADC speed of 1.88Hz. The frame length is 29 bits, the increase of the conversion from bit to bauds is factor 1.1.

The ADC rate is so low to reduce the noise problems, see Section 9.5, *Instrumentation problems*.

The interval is (17ms,34042ms). So an interval of 400 ms fulfills all the requirements and leads to a bus occupancy of 4.1%.

It is important to remark that changing the ADC rate does not affect the bus occupancy. This parameter is only affected by the SyncInterval. What happens that the change in the ADC rate can lead to a no longer valide SyncInterval as stated in equation 13.3

13.7.3 Model of the OPC

The CAEN and Wiener OPC servers use the concept of OPC groups for refreshing the values, while the ELMB one does not.

The two main parameters in a OPC group are:

- Update rate
This is a parameter between the OPC client and server. It determines how frequently the client is notified of new values.
- Refresh timer
This a parameter in the client. If no update is received in this interval, it forces a poll to the server. Consequently 'UpdateRate' must be greater than 'RefreshTimer'. This parameter can be 0 and this functionality is disabled.

The newest specification of OPC allows to handle OPC items individually without any group, as is the case for ELMB. with automatic notification to the client.

OPC server	Groups	Update rate	Refresh timer
CAEN	Yes	2000 ms	0 s
Wiener	Yes	3000 ms	0 s
ELMB	No	—	—

Table 13.3: OPC update rates

However in the supported DCS design, every machine has its own OPC server and client. The distributed capabilities of OPC are not used. One could say that this has the implications on the CPU load of the machine. That the same machine has to do the OPC server side filtering and the propagate to the client. But in the other hand the networking overhead is no longer applicable.

13.7.4 Model of the FSM

Each FSM node has a processing time after one of the children changes or after receiving a command for the parent. This time is specific for each node, because it is directly related to the number of children, and the number of states of the node itself. Furthermore it can even include network requests, ...

On average a reasonable estimation can be 500 ms for this internal processing time on the node.

13.7.5 Global chain

The global idea behind the DCS timing estimations is to be able to handle the soft 'real-time' needs of the system. It is necessary to execute always the actions. But if any is processed with more delay that

the desired, this should not be a critical issue. So this timing is defined between the observation of a sensor (or any other datapoint in PVSS) and the execution of an action.

Having an estimation of this timing is crucial for evaluating when for some fast responses needed does it worth to have a warning or if the first wrong value must be a trigger for a software interlock.

An example of this is to correlate the vacuum degradation in the sensor when the pump is disconnected with the time to disconnect the cooling plant for avoiding ice on the electronics. This situation is calculated in Table 13.4.

The approach is straightforward:

1. Determine the RefreshRate of the sensor and the actuator. It will be the syncInterval of the bus CAN, or the UpdateRate for Wiener and Caen OPC servers.
2. Calculate the CommonNode in the FSM hierarchy between the SensorNode and the ActuatorNode.
3. Let be Measure the number of levels of difference between the CommonNode and the sensor node.
4. Let be Command the number of levels of difference between the CommonNode and the actuator node.
5. The total FsmProcessing time is $(\text{Measure} + 1 + \text{Command}) * 500 \text{ ms}$
6. The final ReactionTime for a hypothetical soft interlock is $\text{RefreshRateSensor} + \text{FsmProcessing} + \text{RefreshRateActuator}$

It is important to mention that this timing does not consider hysteresis in the sensors. This calculation is only since the instant the sensor react, not since the physical magnitude changes.

	Situation A	Situation B
Sensor	Vacuum sensor inside a pot	Temperature sensor inside an hybrid
Actuator	Cooling circuit for a full station	Wiener Maraton Low Voltage Group
SensorNode	tot_Rp_45_220_fr_tp_Vacc01	tot_Rp_45_220_fr_tp_Temp01
ActuatorNode	tot_Rp_45_220_CoolPlantLoop	tot_Rp_45_220_fr_LvG
CommonNode	tot_Rp_45_220	tot_Rp_45_220_fr
RefreshRateSensor	400 ms	400 ms
RefreshRateActuator	500 ms	3000 ms
Measure	3	2
Command	1	1
FsmProcessing	2500 ms	2000 ms
TOTAL ReactionTime	3400 ms	5400 ms

Table 13.4: Soft interlocks estimation

The reaction time for stopping the cooling circuit after the vacuum is lost is explained in *Situation A*. The FSM states will propagate upwards 3 levels in the FSM hierarchy up to the proper station. Here a command is sent to the cooling child node to close the loop.

13.7.6 Experimental results

Due to the fact that the detector are not fully mounted still, the current test are limited to real hardware for 8 Roman Pots and 2 Quarters of T2, but having all the datapoints and the full tree of the FSM generated.

No performance problem has been detected. The delays in the refreshing and updating of values are valid for the usual operation tasks.

(This page is intentionally left blank)

Verificaton of the DCS

14.1 Commisioning of the RADMON-DAC

To test each ELMB-DAC, a system as detailed in Section 10.4, *RADMON Connectivity into the DCS readout electronics* was used. The software allows to set the current value in each channel in a range of 20 mA (12 bits - ADC) and to readout the voltage drop over the sensors and on the resistors placed in the return lines (e.g. current measurement). The 29 produced ELMB-DACs were tested [Din08] in a setup constituted by 1 ELMB, 1 ELMB-DAC and 1 PP board that allows the readout of 3 ISCs in the configuration with 2 RadFETs, 2 diodes and a temperature probe. These channels were labelled as follows: RadFET 1, RadFET 2, Pin 1, Pin 2 and temperature sensor. All channels have been tested systematically by setting the current, for each channel, in the range from 0 to 4000 ADC counts in order to explore the whole dynamic range of the DAC channels.

In the test bench the 4 radiation sensors were simulated by four 1kOhm resistors, while the temperature sensor has been simulated by a 10 kOhm resistor.

The ELMB-DAC card number 5 was tested in more detail for low currents, and with small increments of current until reach the maximum. The results are shown in Figure 14.1. The results confirm a linear relationship between current and voltage for all the channels.

For the readout of the radiation sensors, the time in between the current injection and the voltage readout has to be setup properly. Therefore was important to measure the DAC time response with different ‘delays’ between those two moments. Figure 14.2 shows the U-I curves recorded from the ELMB-DAC card number 5 (channel RadFET 1 of the ISC 1) with different time delays. In this test, the ELMB measurements were compared with the voltage measurements done with external Keithley multimeter. The results show a decrease of U with the time delay (see 100 ms data). To test the repeatability of the data for the U-I curve at 100 ms, a statistical test was made with 10 measurements per point. Figure 14.3 shows a maximum standard deviation of 0.3919 V.

To complete the time performance study of the ELMB-DAC system, the current and voltage delivered with different delays was also measured with an oscilloscope. The tests took place on the PIN diode 1, ISC 3, ELMB-DAC card number 5.

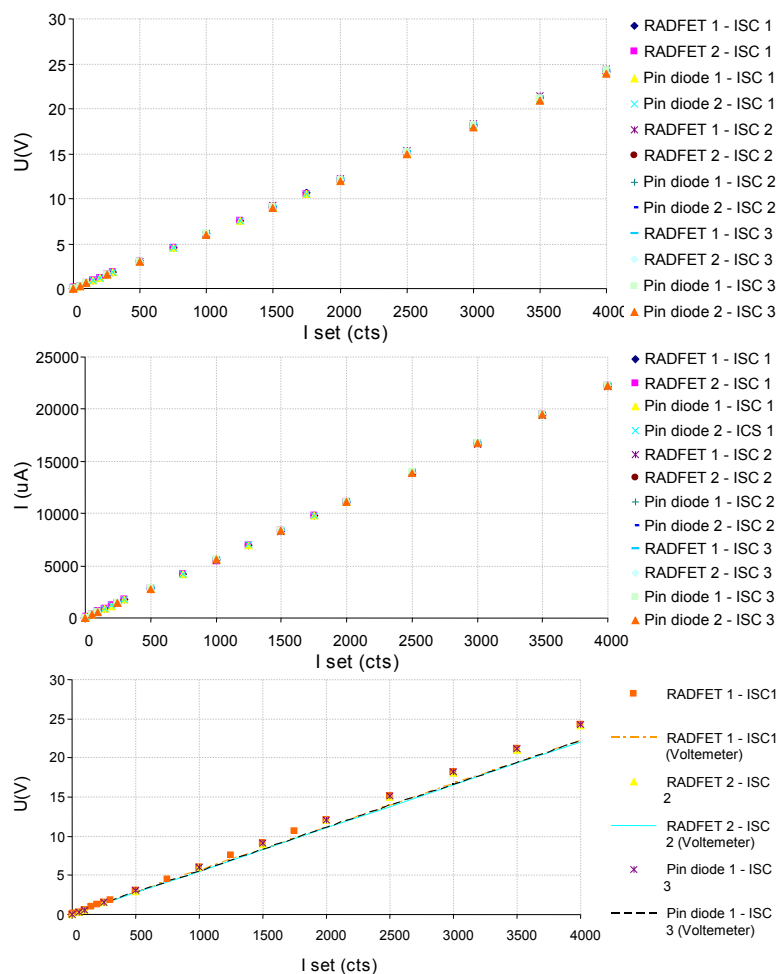


Figure 14.1: Measurements of performance (I-V curves) of the ELMB-DAC card number 5

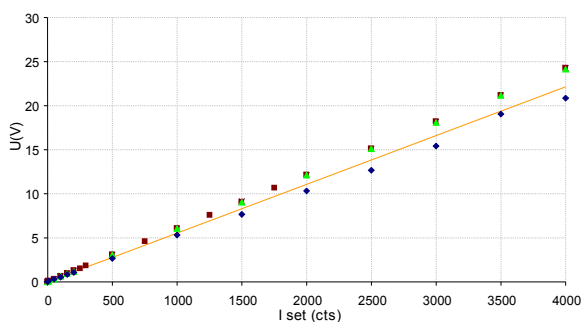


Figure 14.2: Time delay analysis: U vs I set

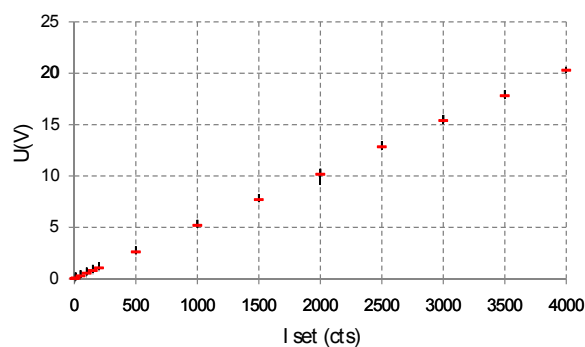


Figure 14.3: Statistical test of the repeatability of the data, which shows the error bars (black) and the mean value for every measurement (red)

The commands sent via the PVSS system and read through the ELMB analog inputs were crosschecked with an oscilloscope in Figure 14.4 and in Figure 14.5. In this comparison is observed that time delay cannot be decreased from 250 ms due to the ELMB SDO commands needed for the readout. Also for a delay lower than 500 ms, the voltage level recorded by the ELMB analog input is lower than the

oscilloscope mean value. We also could measure a pulse rise time of 18.2 ms and fall time of 8.64 ms.

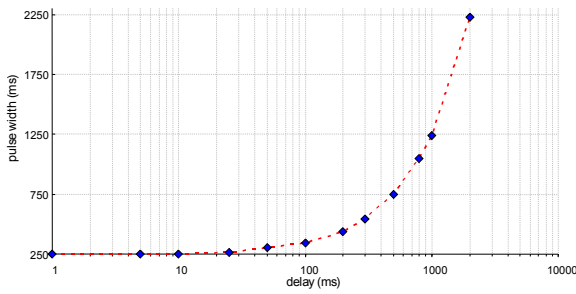


Figure 14.4: Time delay analysis: pulse width vs time delay

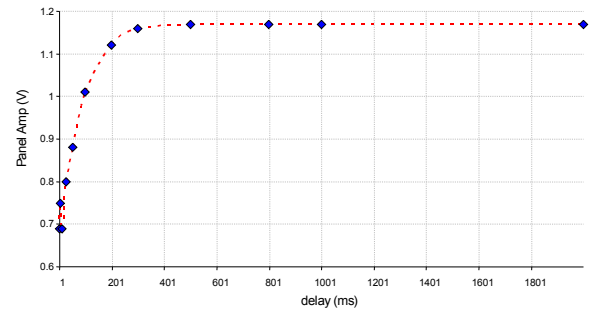


Figure 14.5: Time delay analysis: ELMB analog input measured voltage vs time delay

The system was put in place in the CMS counting room (USC55) for the first beam circulation during the 10 September 2008 (LHC inauguration). At the same time a second system was running permanently in the development lab. After some days of measurements, the data quality of the system was not satisfactory; from time to time some values totally wrong, including the currents injected. However, results from the system in the laboratory were better than the ones in the final system

The hardware was exchanged and tests including all the ELMB software layers were done. The conclusion was that the switches in the DAC board were not responding properly, and it was a hardware problem. The problem is that the logic insided the DAC board is powered with a 12V Power Supply (Analog input of the ELMB) while the 30V for operation of the sensors are provided from a different one. A common ground line was put together between the two power supplies; although the schematics of the DAC assures that both grounds are put in common. The improvement of the measurements were dramatic, as can be seen in Figure 14.6. So after this connectivity between both Power Supply grounds, the RADMON system was considered satisfactory.

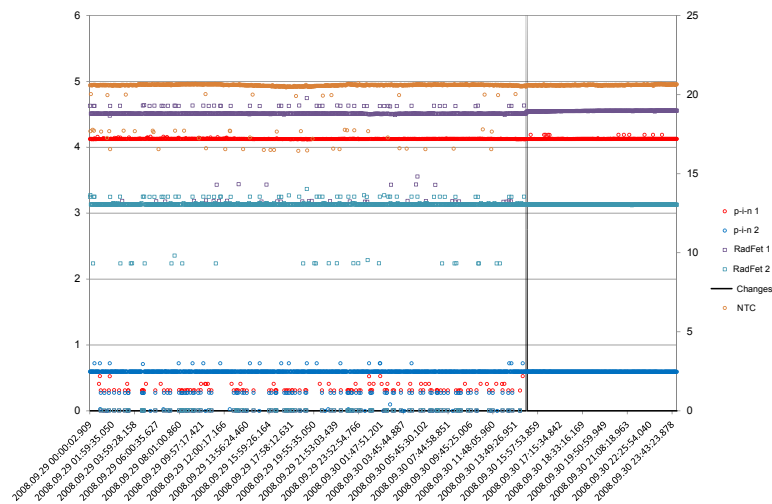


Figure 14.6: Grounding effects on RADMON readout

14.2 Replicability of the system

The system has been proved to be replicable. The components generated during the development, are tested in several computers in the development laboratory and in the final production nodes,

without changes in the behaviour.

14.3 Response of the critical actions

Most of the subsystems have been on maintenance after the LHC incident of 2008. For that reason there was not an active data flow among them, just the monitoring of the stand-by status.

However, the stand-alone DCS test have always behaved better than in the calculations of Chapter 13, *Information theory analysis*; both in reaction-time and memory usage.

Chapter 15

User interface

15.1 ALICE interface

The TOTEM DCS uses the ALICE main interface as a first approach for the User Interface. The equivalent in the CMS experiment is not so mature. A significant number of modifications took place to be compliant with the CMS policies for component.

The main screen can be seen in Figure 15.1. As can be observed there are separated areas for the FSM navigation, control/monitoring functions, log and 'know-how', external systems,...

In Figure 15.2 the FSM tree is shown in detail. Each FSM node is represented via a small icon representing the node type followed by its name:

1. yellow toothed wheel for CU
2. cyan toothed wheel for LU
3. green rectangle for DU

When a node is selected, the main area of the screen is updated with an specific control panel for that FSM node. In this place is where the integration of panels take place. There are three kinds of panel integration

1. JCOP framework panels for basic hardware control; Section 15.2, *JCOP integration*
2. TOTEM specific panels for summarizing the states of each Product Breakdown item of the detector decomposition; Section 15.2, *JCOP integration*
3. CMS 3D view; Section 15.4, *CMS 3D view*

Also it is possible to send the commands to the FSM from the treeview.

Additionally there is an FSM Expert Control Panel, Figure 15.3, that presents the the FSM detailed status and full control.

15.2 JCOP integration

In order to provide a fast development of control and monitoring functions the the many JCOP panels have been integrated directly. However, some panels have been modified to correct some bugs.

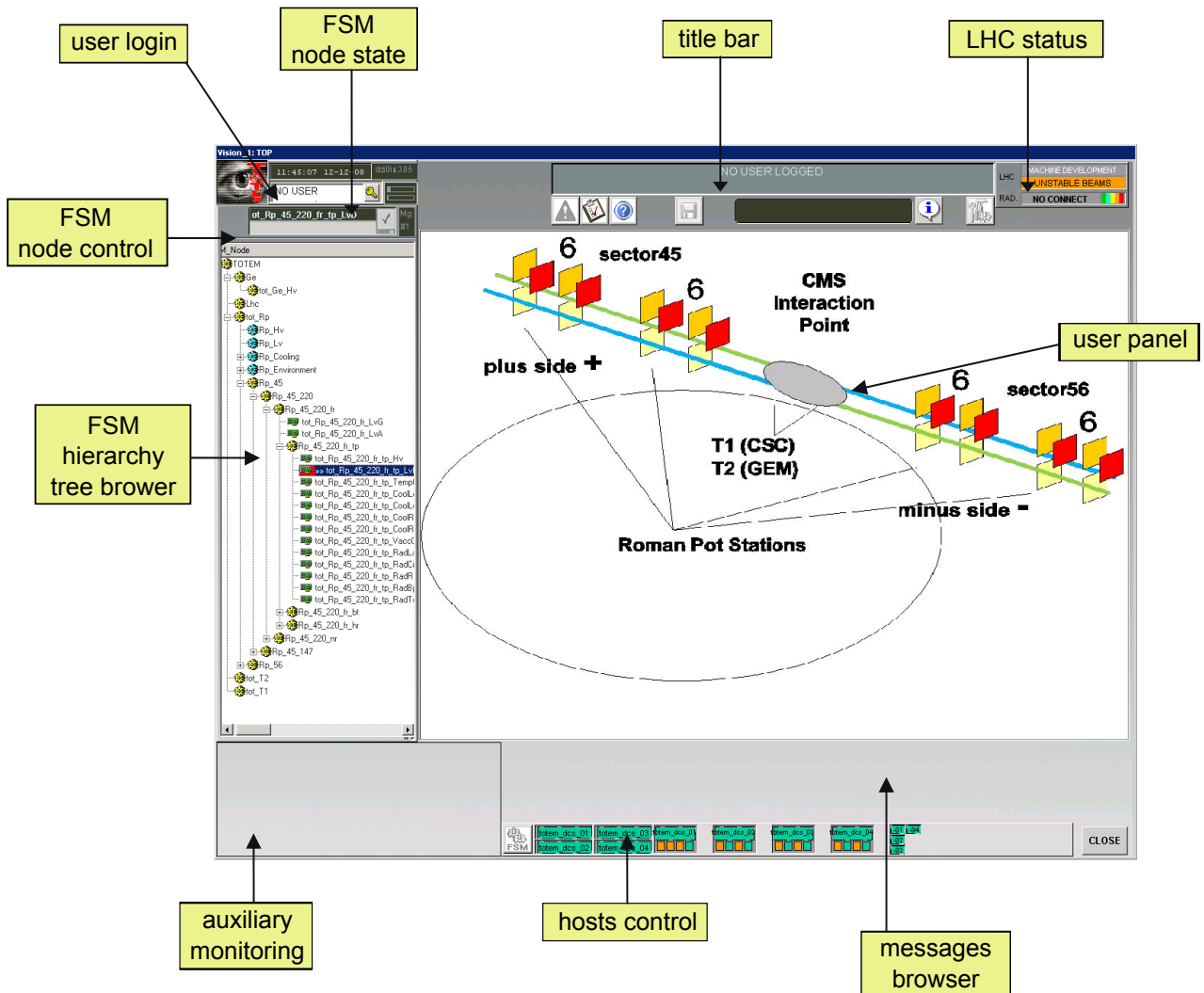


Figure 15.1: The main screen based on the standard DCS User Interface provided by the ALICE Central Team

A example of them are for crates and channels operation from CAEN Figure 15.4 and Wiener Figure 15.5.

15.3 TOTEM specific panels

As the detector is decomposed following the PBS, specific panels for each one of the levels are needed. Figure 15.6 shows a panel for a chamber detector such as T2. The idea is to present the information in a structured way following the layout of the detector. This area can be sensitive to the user actions and provide status information through colors.

The overall panel provides ways to access panes such as Figure 15.7. In this one trending capabilities, reconfiguration actions or any other kind features of expert operator take place.

The development of fully functional panels following a geometrical layout is considered about the 20% of the development time. To provide a first release in reasonable timescale but fully functional panels a shortcut took place. Using the self-inspection tools of PVSS, the generic panel of Figure 15.8 was developed. The logic implemented is the following:

1. It takes as an argument the name of the current PBS level.

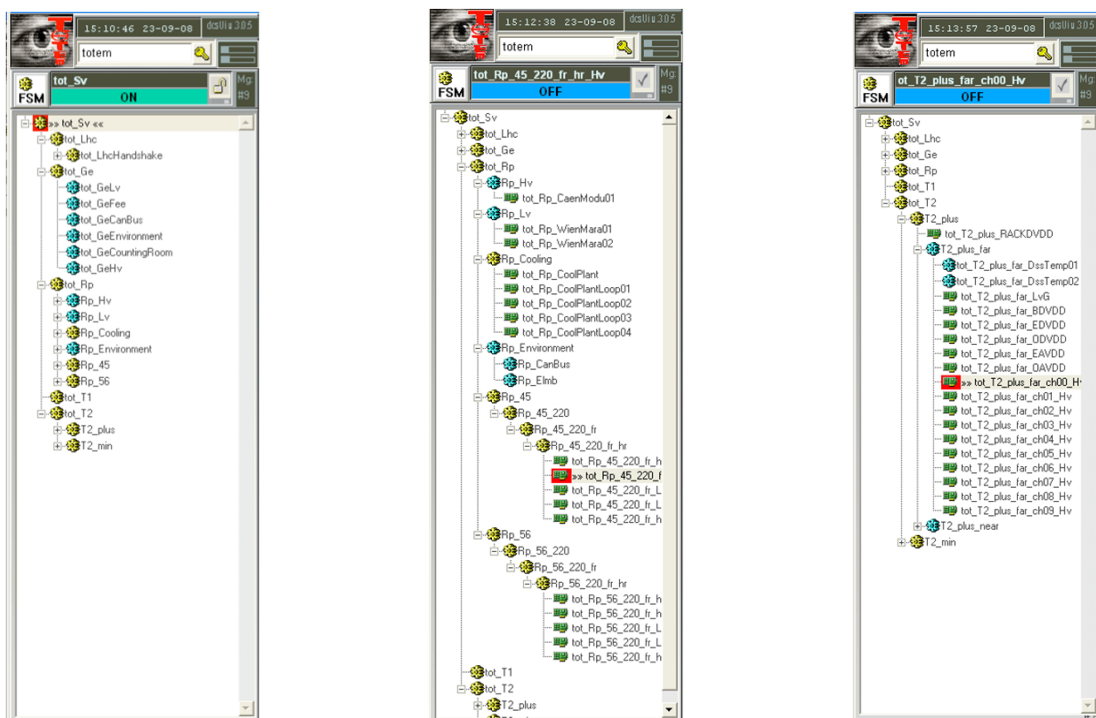


Figure 15.2: FSM tree expansion in the main environment



Figure 15.3: FSM control

2. Searches in the distributed project all the logic names that are in that PBS node.
3. From the logic names finds out the hardware name.
4. Links the hardware name to callback routine to updates the values in a table.

User interface

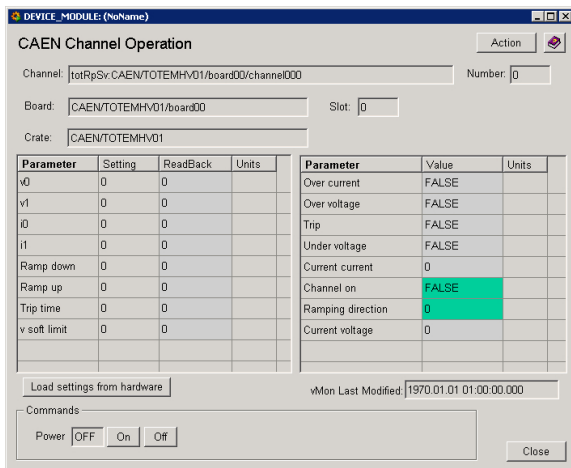


Figure 15.4: CAEN High Voltage Channel control panel

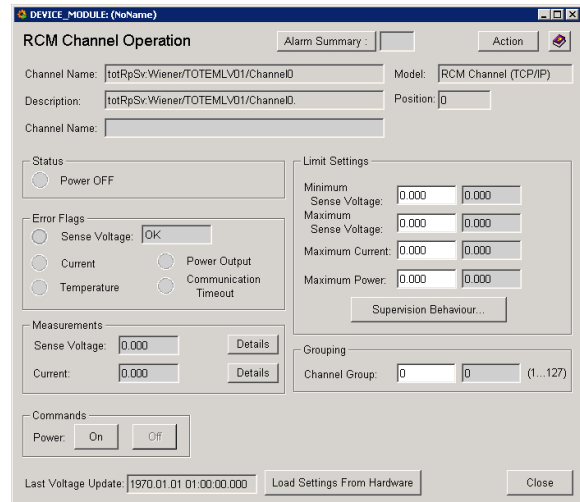


Figure 15.5: Wiener Maraton Low Voltage Channel control panel

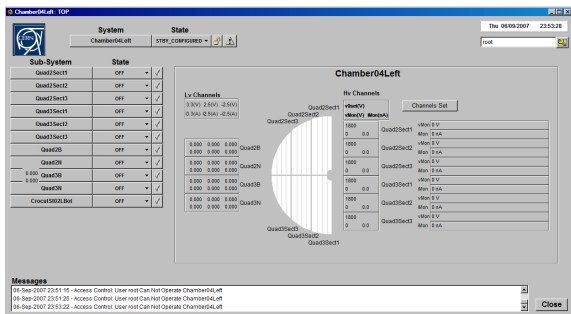


Figure 15.6: FSM explorer

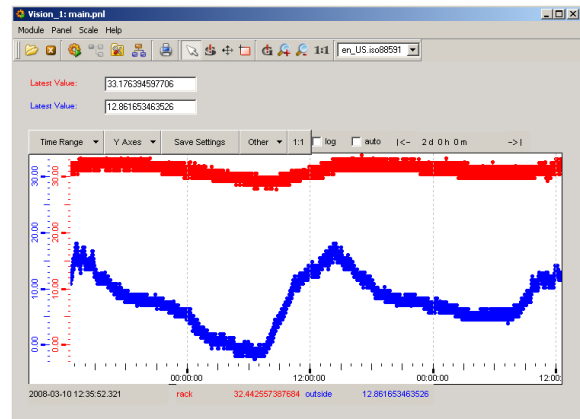


Figure 15.7: Temperature plot

The advantage of this panel is that it is not needed any maintenance to provide all the information implemented in the project. When new development takes place they follow the naming convention, and are automatically detected by the user interface and shown properly.

It is important to make clear that this panel does not provide control functions, only a simple monitoring takes place. For having an in detail monitoring or control functions is possible to continue navigate further down in the FSM tree up to the desired sensor or channel.

15.4 CMS 3D view

The goal of this panel is to provide a 3D interface automatically built using the geometry database of the experiment and the FSM tree. This development is not still operational in TOTEM, but the results should be similar to the Figure 15.9 from the CMS Tracker. The parts of the detector can be made transparent and they are colored according to the FSM states.

Then a part of the detector is in the state ERROR or CRITICAL they are colored in red or yellow, and all the OK parts are made transparent.

(This page is intentionally left blank)

Chapter 16

Conclusions

16.1 New results

The main results accomplished in this thesis are:

- **Hardware definition**

Chapter 8, *Comparison with commercial solutions*

Chapter 9, *Frontend sensors*

Chapter 14.1, *Commissioning of the RADMON-DAC*

The current CERN technologies, and possible alternatives have been evaluated. CS-Framework and Labview were under study, but PVSS was selected for the reuse of CERN and JCOP technologies. The logical connectivity of all the ELMB for the whole experiment has been defined, identifying in number and type all the different environmental sensors that will be used by the different detectors. Study the impact of the tunnel electromagnetic noise in the 320 meters long cables used for environmental sensors. Commission the developed radiation monitoring system, and implement satisfactory the improvements.

- **Define representations**

Appendix B, *Hardware overview diagrams*

Section 10.3, *Pinout tables and hardware generation scripts*

Section 11.3, *FSM hierarchy tables and operation logic generation script*

Section 11.2.2, *Using UML*

The operation logic for the Roman Pots motors and Silicon Detector has been specified. All the hardware configuration has been represented with a diagram notation developed in ALICE, showing graphically all the interconnection of the hardware devices to be controlled and monitored.

- **Standardize the inputs and develop automatic processing tools**

Section 10.3, *Pinout tables and hardware generation scripts*

Define most of the pinout for all the signal and power cables used in the experiment (together with Evangelia Dimovasili). Define tables for expressing the hierarchy of the DCS

Finite State Machines and the relationship with all the hardware elements (together with Ivan Atanasov). This table is processed automatically to build major parts of the DCS software. This step of having a common format for all the connectivity allows the use of automatic development tools.

- **Use generic self-inspection tools**

Section 15.3, *TOTEM specific panels*

The generic panel of the user interface can be use as the first approach in the development to provide an usable product. But further more can be used as a debugging tool to verify more 'user friendly' panels.

- **Define process of developing the DCS**

Section 7.1, *Evolution of the thesis in relation with the TOTEM experiment*

Section 7.5, *Development process*

The development is structured in work packages, and milestones following the GPDM methodology. The most important conclusion is that in this environment requirements changes daily. It is necessary to adequate the software as fast as possible prioritizing correctenes to completeness. If some features for making simpler the operational procedure are not available, are a minor problem compared with having a system that will produce damages into the LHC. For that objective is necessary all the information structured in a formal way, and develop automatic generation tools.

- **Information Theory analysis**

Chapter 13, *Information theory analysis*

It has been provided estimations and a model for the computing resources and monitoring devices. They are calculated with a custom tool analyzing the system structure and devices based on information flows exchanges. It also provides estimations for archiving storage needs and timings.

- **Configuration management**

Section F.2, *SubVersion*

The SVN infrastructure set up for the website, documentation and software development has been studied by IT for the future CERN-wide SVN service. Also a Configuration Management Policy according to the project lifecycle is defined (expressed through Work Packages and milestones of the GDPM plan).

This thesis has shown the importance of the DCS for TOTEM because of the complexity of the experiment and the large diversity of components utilized. The work presented here is the definition of the TOTEM DCS and it has permitted radiation measurement during the LHC start up in 2008 and basic functionality in 2009.

16.2 Future researches

There are quite a few ways of continuing the work in the DCS:

1. **Need for Standarization**

It is unreasonable that each experiment has it source of information for the DCS configuration in different formats and different tools. This only leads to duplication of needed human resources. This effort could be coordinated with organization as IEEE or generate an internal CERN specification for that under the JCOP initiative.

2. Development of better automation tools

One great success of the work done during this thesis is the definition of the pinout tables, FSM hierarchy tables and the corresponding scripts. In this way the DCS logic can be reviewed by the detector responsible on paper, and then processed automatically. The usual way of proceeding is that the detector expert/responsible provides that information in a piece of paper using an own nomenclature. The DCS developer has to understand that nomenclature and map everything into the PVSS nomenclature. All this procedure is done manually and takes many intermediary steps. This leads to mistakes and having to redo significant part of the work, delaying consequently the development. This kind of automation tools can also check the integrity of the input data and warn of possible problems, as power channels not plugged, sensors connected wrongly, ...

This way of working probably has saved several months of work in TOTEM DCS.

3. Strengthen system robustness to partial failures in hardware and software

Also in the dynamical rearrangement of the individual detectors.

4. Study the interaction among the Roman Pot data rates, beam loss monitors and the other radiation monitoring sensors

Section 14.3, *Response of the critical actions*

5. Study the reaction time, improve if necessary

Section 14.3, *Response of the critical actions*

It could be interesting to provide a expected maximum response time. In this way the DCS could be treated as a 'soft' real time system. Having built a tool for calculating the information produced in each node of the experiment and how it is propagated to the computing resources is the first step of this idea.

16.3 Development recommendations

From the project definition point of view, the following topics are the most relevant:

- **Need of defining a proper Product Break Down structure and Naming Scheme**

Section 7.4.2, *Detector Product Breakdown Structure (PBS)*

Section 7.4.3, *Naming Scheme*

In TOTEM there was no naming agreed for the different parts of the detectors. Even more every team was using a nomenclature completely different. This was a serious problem for the DCS, as this system has to interact with most of the other parts and the operator. It is needed to identify univocally all the parts of all the detectors.

- **Identify responsables for each piece of information and requirements**

Section 7.4.2, *Detector Product Breakdown Structure (PBS)*

Section 7.4.1, *Templates*

The ideal situation would be to have a matrix of Person/Information. However all attempts to formalize this information have failed. The solution was to have a 'Requirements Document' managed by the Detector Responsible where the description of the detector, naming and operational procedure should be detailed. This person interacts with the needed contacts and summarizes the information. For other pieces of information such as the pinout tables has been done in much more detail, due to the criticalness of that information.

- **Need for a proper Configuration Management Plan**

Section 7.5, *Development process*

Conclusions

This is the way of tracking what is installed on each computer, why has been set up in that way, and the lifecycle of that installation. It is needed to define baselines, versions, revisions and so on. CMS central team does not define any anything similar. In fact, they upgrade the software even without notifications. The IT-CO group define versions, but not properly the dependencies among them neither baselines. This can lead to serious problems during the lifetime of the DCS. Having a SubVersion repository helps us to manage our versions but in this environment it not enough. It is needed to improve drastically this situation, at least to know exactly what version of the software is running in each device.

Part **IV**

Appendices

(This page is intentionally left blank)

Locations

A.1 The laboratory in building 555

Most of the development and testing takes place in the building 555, in the laboratory space at the southern edge of the building, shared with Electronics and DAQ. It is operational since July 2007 and as can be seen in Figure A.1 a replica of most of the control and monitoring functions has been assembled.

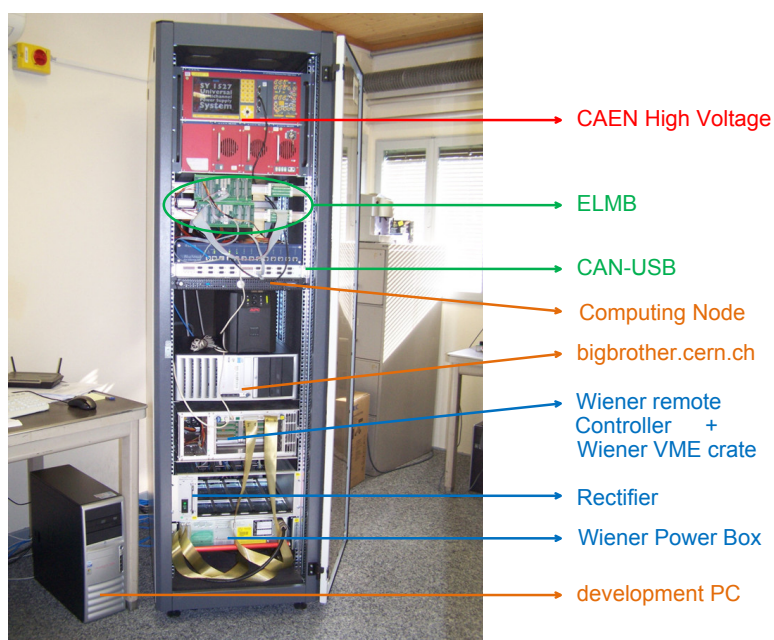


Figure A.1: DCS rack in the laboratory of building 555

A.2 Test area H8

This is the test area for TOTEM. All the detectors, Roman Pots, GEM and CSC are tested here with full electronics before installation in the tunnel or the experimental cavern. This is the so called commissioning process.

The lack of human resources in the DCS team, and the immature developments lead to not having any commissioning activity for the DCS control during 2008. The power supplies were operated manually and sensors monitored with different systems by the detectors production groups.

However some other groups (electronics and DAQ) had been working there actively in this environment.

A.3 USC55: counting room

This is the final destination of big.brother is the CMS counting room (at Point 5).

This area is usually accessible, even during beam operation. However the magnetic field of CMS is noticeable.

The location of the ELMB for TOTEM was a non trivial problem. They are qualified for operation up to 2 T and they are also radiation tolerant. For ATLAS and CMS, they are used inside the cavern; but the radiation levels for the tunnel and the forward region of CMS are 1 or 2 levels of magnitude about the radiation tolerance tests. The solution was to install all of them in USC55 and to drive the signals using long cables. For T1 and T2 they are around 120 m long meanwhile for the Roman Pots they are up to 320 m long.

Inside rack S2B19 (Figure A.3) the 5 DCS PCs, the DAQ PCs and the CAN-USB controllers. Rack S2B03 hosts the HV crate and HV patch panels; while S2B02 (Figure A.4) hosts all the ELMB boxes, and corresponding power supplies. Rack S2F05 hosts the Wiener Maraton Rectifiers and Remote Controllers. VME crates are in rack S2E11.

The motorization control is assembled in 3 different racks (S2S01, S2S02 and S2S03) shown in Figure A.5.

A.4 UXC55: experimental cavern

This is the experimental cavern of CMS in Point 5. All the CMS detectors, T1 and T2 are located here.

Figure A.6, Figure A.7 and Figure A.8 shows the CMS insertion with the beampipe.

In this area between the CMS endcaps and the beampipe is located the platform described in Figure A.10. The CMS HF calorimeter, TOTEM T1, TOTEM T2 and CMS Castor are installed on it. The other side of CMS is simmetrical.

In the racks situated on the platform is situated part of the DAQ electronics and the maraton power box, Section C.2, *Low voltage Wiener power supplies*.



Figure A.3: Rack for the DCS computers and CAN-USB adapters

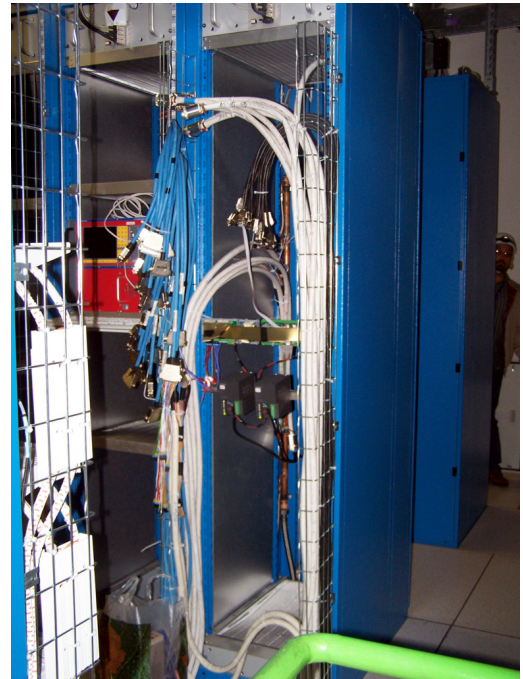


Figure A.4: Rack for ELMB

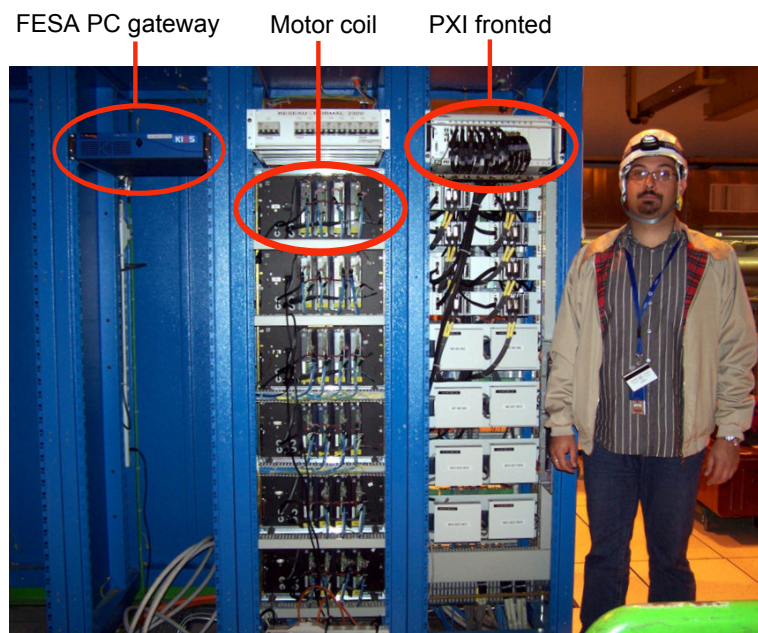


Figure A.5: Racks for the motor control

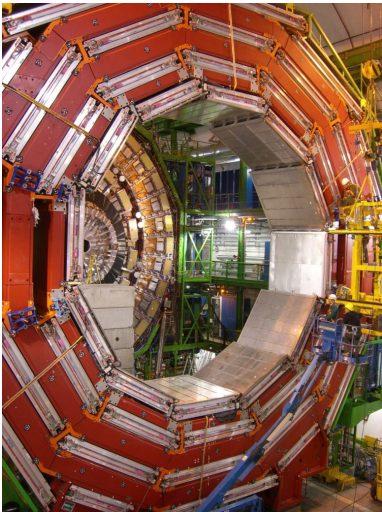


Figure A.6: CMS cavern:
wheel

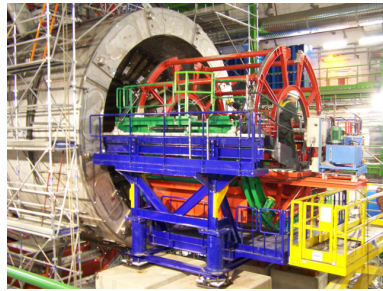


Figure A.7: CMS cavern:
platform

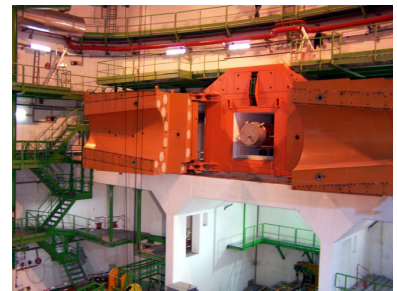


Figure A.8: CMS cavern:
insertion

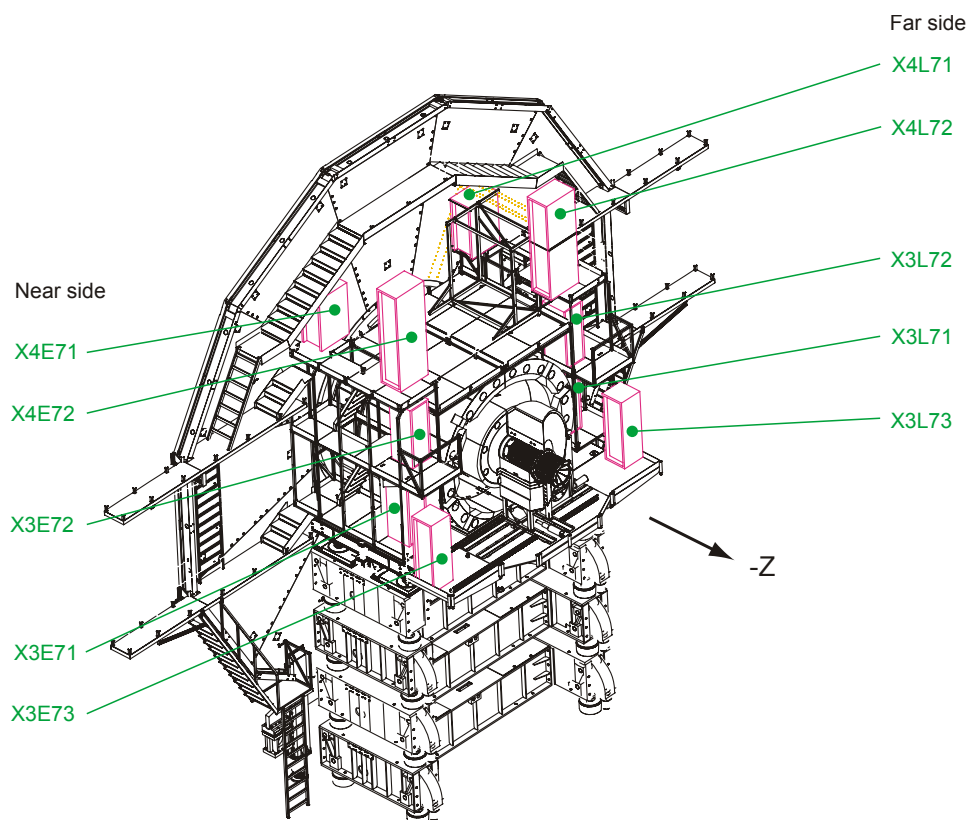


Figure A.9: HF Minus

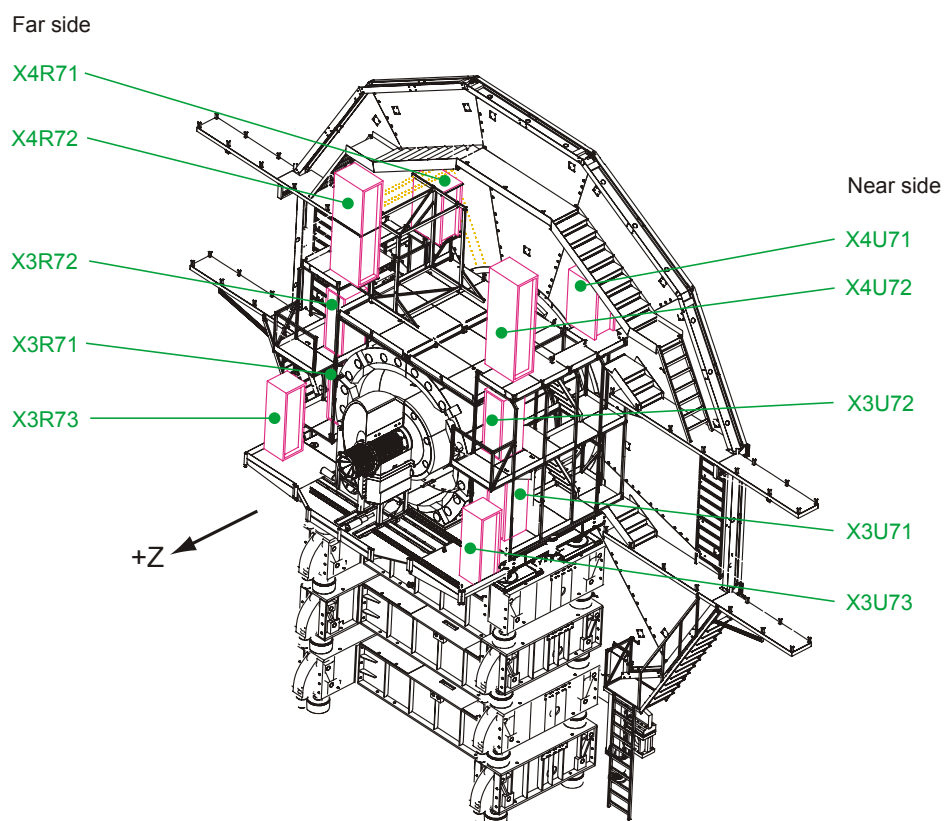


Figure A.10: HF Plus

A.5 Sectors 45 and 56

This is the location of the Roman Pots. As described in Section 3.3, *RpMe* (*Roman Pot Mechanics*), there are two stations at each side of IP5.



Figure A.11: Installation place for a RP station



Figure A.12: Roman Pots Patch Panel

A.6 Alcoves UC53 and UC57

Those are ‘service rooms’ between the straight section and the curve section in the tunnel at each side of CMS.

In them are located the maraton power boxes for the Roman Pots of that sector. The DC power comes from the primary rectifier in USC55 through the tunnel and is delivered back into the corresponding pots of that sector.

(This page is intentionally left blank)

Chapter

Hardware overview diagrams

This chapter presents the hardware overview diagrams for all the subsystems of TOTEM DCS. It has been assembled with the input provided by the detector responsible, the electronics group and DAQ group .

The format is derived from an equivalent document designed by André Augustinus for the ALICE experiment, see [Aug06].

Each diagram has to main parts:

- **Frontend**

This match the frontend level of Figure 8.1.

The vertical axis represent the 3 different locations: inside the detector, the tunnel or cavern, and the counting room. The horizontally axis represents the different control/monitoring functions. The hardware is connected using different color lines according to the function. The location of the crates in racks, the cardinality of hardware links and crates or sensors is also specified.

The nomenclature is detailed in Figure B.7.

- **Backend**

This match the middleware level and the supervisory level of Figure 8.1.

This represents the SCADA functionality, with PVSS, FSM, databases, communication protocols,... However, these rectangles represents software processes and not physical machines.

The nomenclature is detailed in Figure B.7.

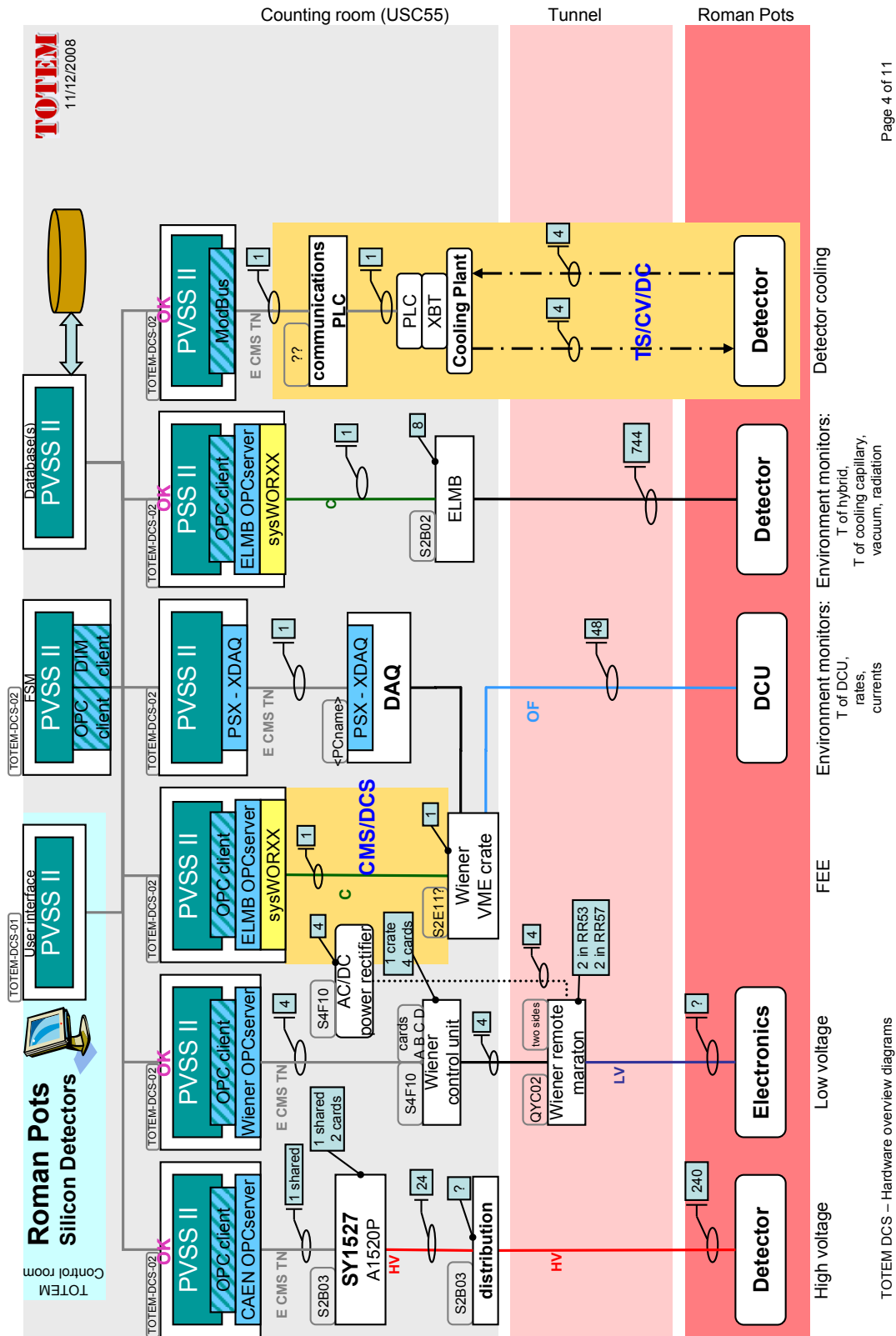
The yellow blocks represents subcontracted developments outside the TOTEM group, being the responsible group mentioned in purple.

There is a compromise in the detail level of the diagrams. The ‘one-to-one’ patch panels and connectors that keep wires without any logical change are not represented. In the other case, the patch panels that merge several wires into a single one, such as the Roman Pots HV distribution box, are specified.

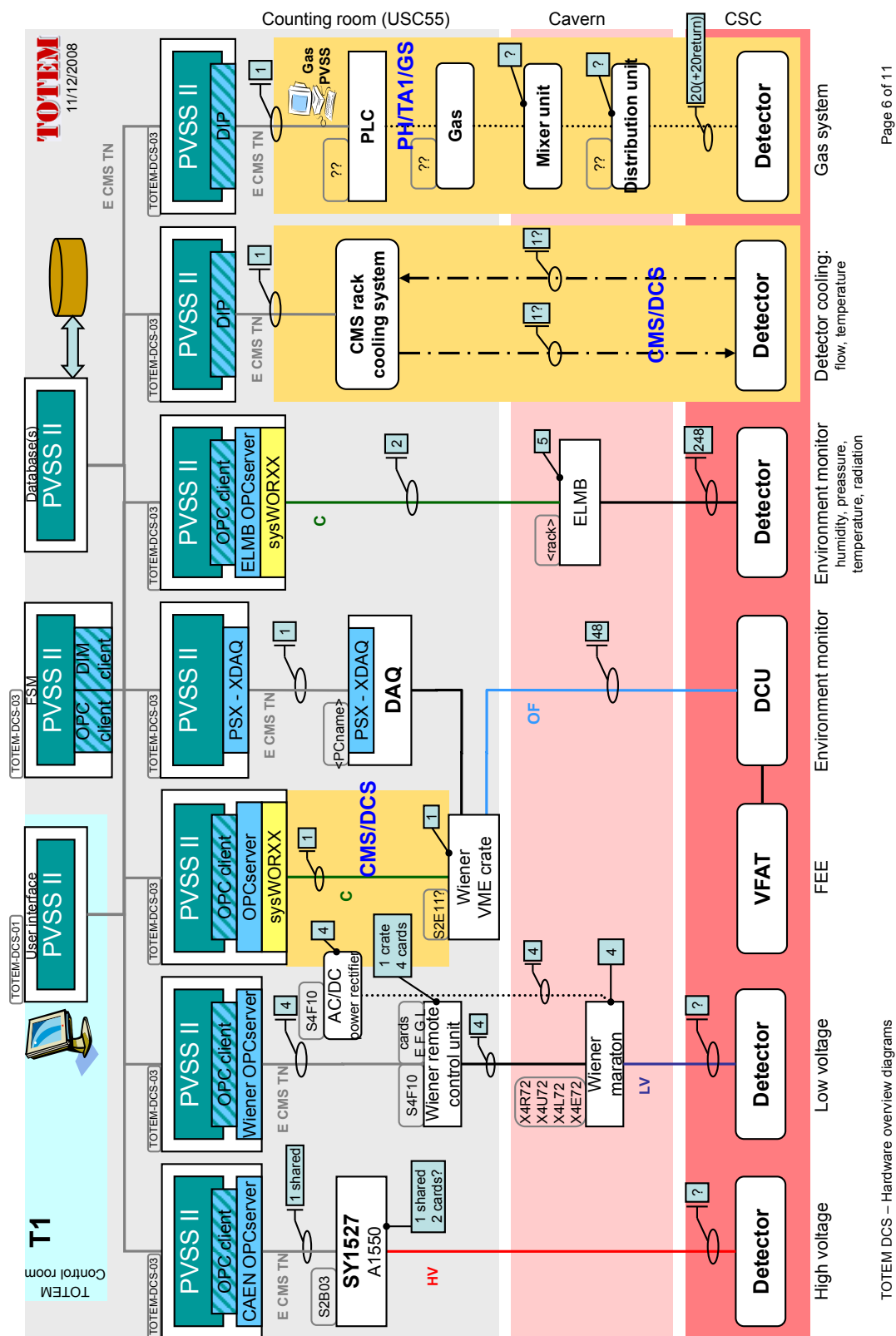
Page 3 of 11



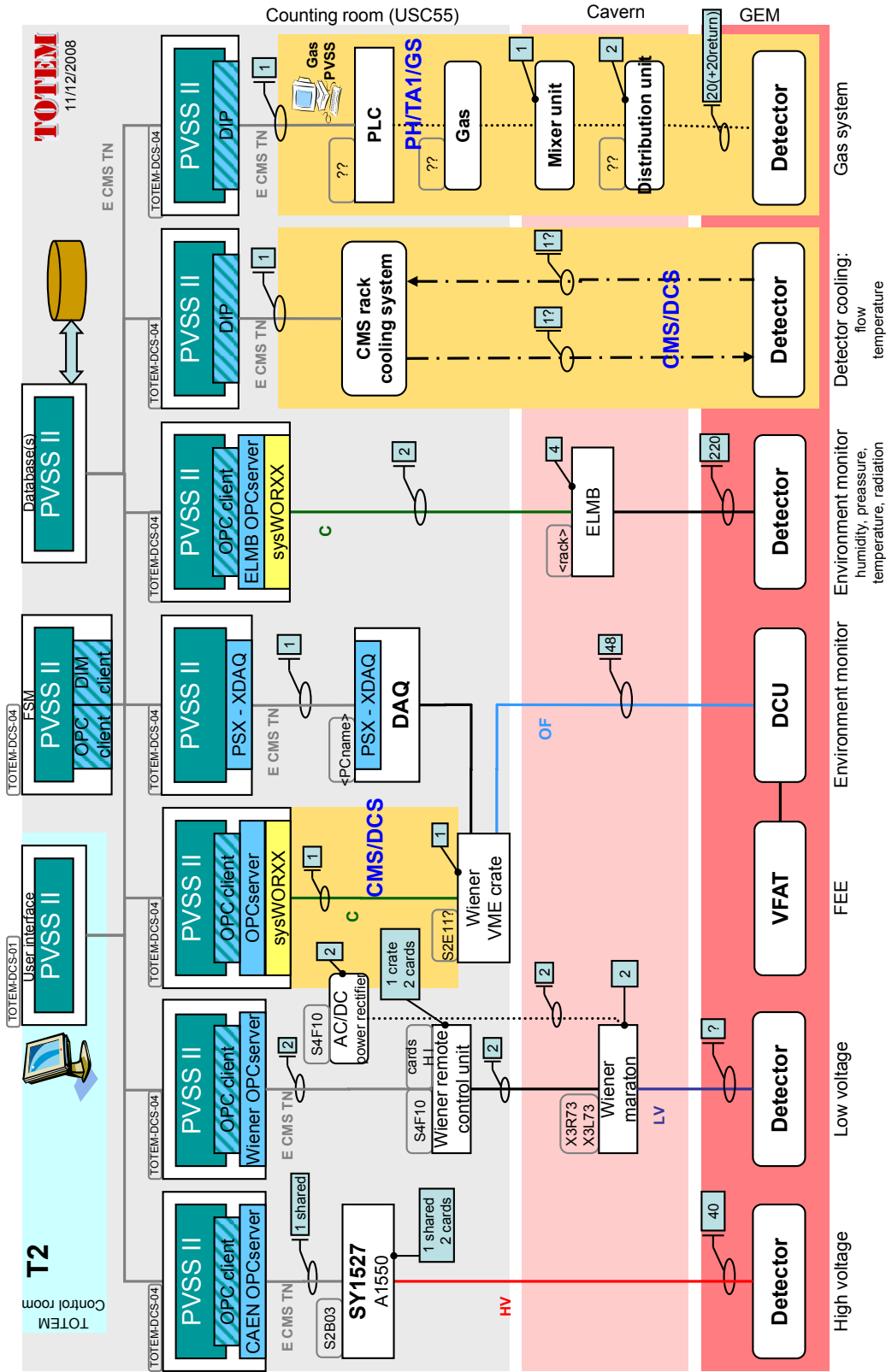
B.2 Roman Pots Silicon Detectors



B.3 T1



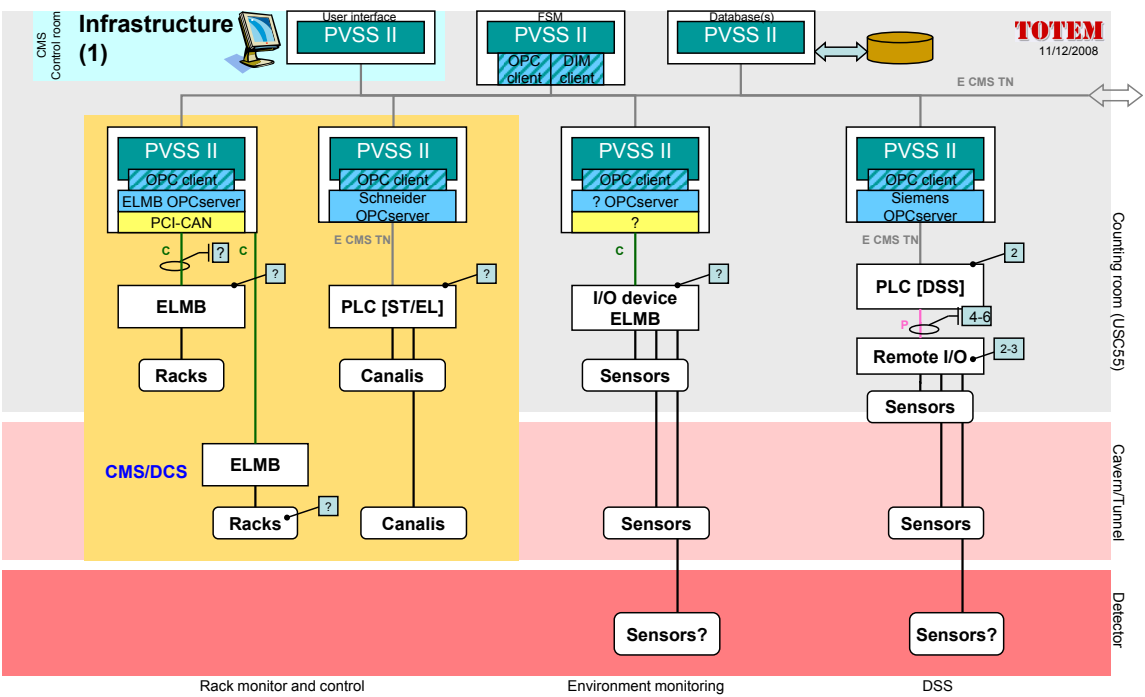
B.4 T2



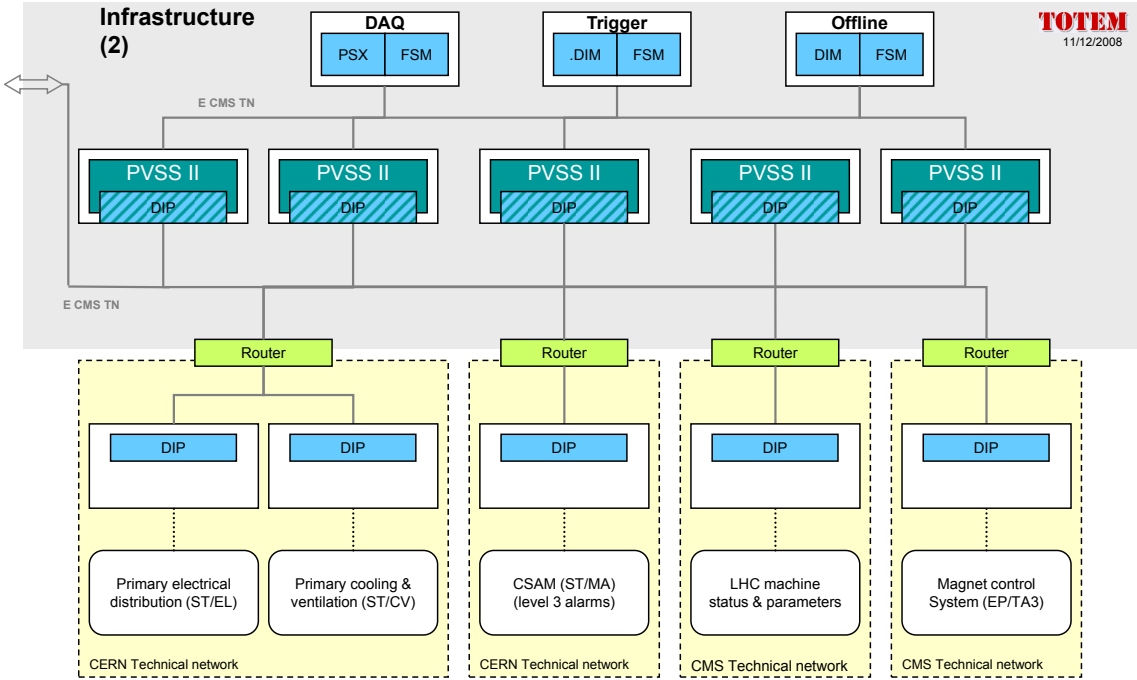
184



B.6 Counting room

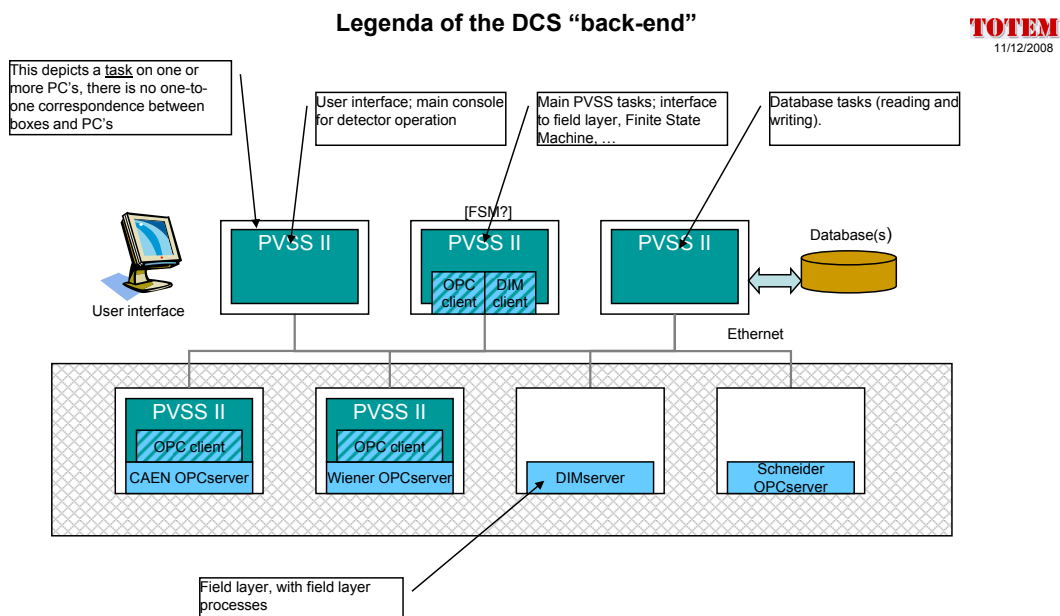
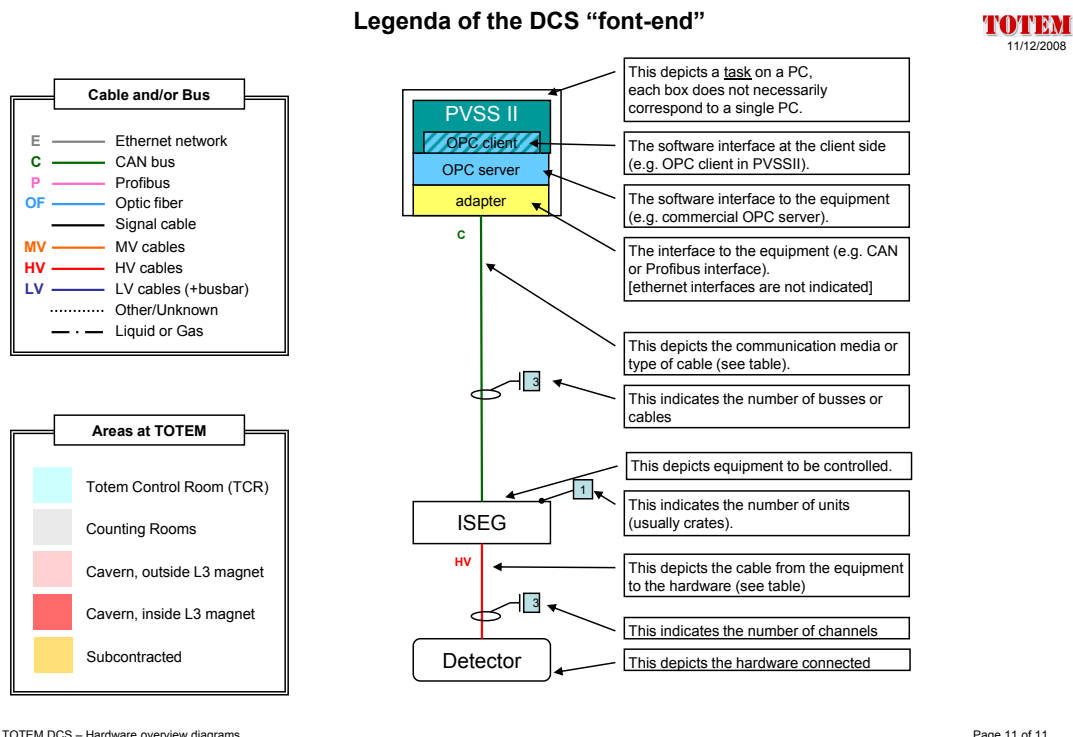


TOTEM DCS – Hardware overview diagrams



TOTEM DCS – Hardware overview diagrams

B.7 Legenda



Hardware components specifications

C.1 High voltage Caen power supplies



Figure C.1: CAEN crate 1527



Figure C.2: High Voltage module

C.1.1 Functional description of SY1527

The SY1527 allows to house, in the same mainframe, a wide range of boards with different functions, such as High/Low Voltage boards, generic I/O boards (temperature, pressure monitors, etc.).

For each slot there is a microprocessor that allows to set up and monitor the system parameters. All the operational parameters are stored in a nonvolatile read/write memory to be still available after Power-Off. The parameters can be controlled either via CAEN traditional built-in links (RS232, H.S. CAENET) or via Ethernet (TCP/IP).

C.1.2 Functional description of A1520P

The Model A1520P is a SY1527 System double-width board which houses 12 positive HV channels [CAE05]. Each output channel has an independent floating ground.

The output voltage can be programmed and monitored in the range [0,500] V with 1 mV steps. If the output voltage differs from the programmed value by more than 15 V, the channel is signalled to be either in OVERVOLTAGE or UNDERVOLTAGE condition. Moreover, for each channel, a voltage protection limit SVMAX can be fixed via software with 1 V resolution and the output voltage can not be programmed beyond this value. All output channels are provided with Remote Sensing Lines to compensate for the voltage drop over the cables. The HV RAMP-UP and RAMP-DOWN rates may be selected independently for each channel in the range [1,50] V/s in 1 V/s steps.

It is also possible to set an output current limit in the range [0,15] mA (25 nA steps). The output current is monitored with 25 nA resolution; if a channel tries to draw a current larger than its programmed limit it is signalled to be in OVERCURRENT condition.

The output current is permitted to equal the limit value only for programmed time interval and then is switched off at the selected RAMP-DOWN speed. The TRIP time (the maximum time an OVERCURRENT condition is allowed to last) can be programmed in 0.1 s steps. If a RESET, INTERLOCK or KILL command is issued to the board, all channels are switched off at the selected RAMP-DOWN speed.

C.2 Low voltage Wiener power supplies

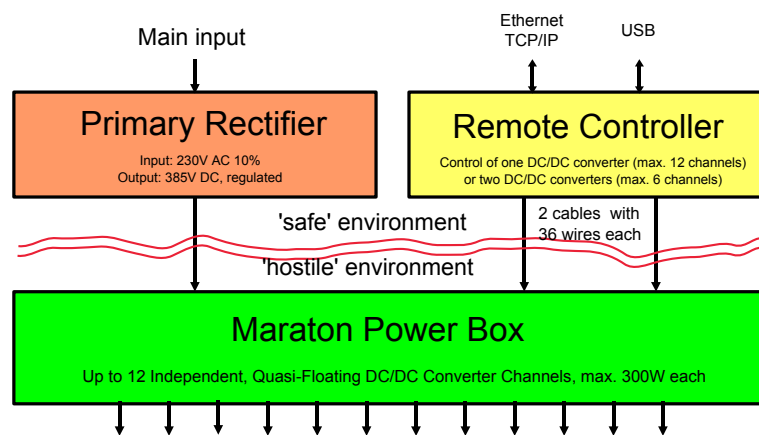


Figure C.3: Wiener Maraton System Overview (see [W-I06])

This Low voltage power generation system is splitted into 3 subsystem as shown in Figure C.3. With this layout is possible to operate in radiation areas, while providing some control functions.

C.2.1 Primary Rectifier

The Primary Rectifier and Remote Controller need to be located in an environment standard industrial conditions (Safe Environment).

This module converts the AC voltage (100 V or 230 V AC, 16 A) to a regulated DC voltage (nominal 385 V). There is no galvanic isolation.

C.2.2 Maraton Power Box



Figure C.4: Wiener Maraton Power Box

The Power Box is located near to the electronics which shall be supplied, and is capable to operate in a ‘Hostile Environment’ (strong magnetic field and / or radioactive radiation).

The Maraton Power Box uses the 385 VDC of the Primary Rectifier and generates up to 12 independent low voltage floating output voltages, that can be independently switched ON or OFF.

A global reset input allows to disable all outputs with just one signal (closed contact).

With a screwdriver potentiometer it is possible to adjust:

- Output Voltage at the Load (Sense Point)
- Maximum Output Voltage at the Terminals of the Power Box (OVP)
- Current Limit for each channel

For remote control and monitoring 6 signals are available for each channel: Two differential pairs for voltage and current monitoring, and one differential pair for a combined inhibit / status signal.

C.2.3 Maraton Remote Controller Module

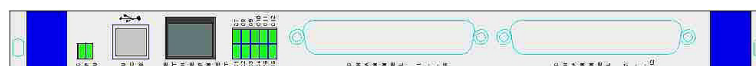


Figure C.5: Wiener Remote Controller

The remote controller for Maraton (RCM) is a 6U VME form-factor processor board. Only the +5V supply voltage of the VME backplane is used, there is no data connection to the VME bus.

After configuration of the RCM by a Windows XP computer connected to the USB port, the RCM provides access to many power supply parameters via Ethernet SNMP.

The following direct control functions are possible:

- Measurement of each Maraton output sense voltage
- Measurement of each Maraton output current
- Read the status of each Maraton channel
- Switch a Maraton channel ON or OFF

The on board microcontroller extends this functionality by comparing these values with additional limits, which can be modified via the network:

- Minimum sense voltage
- Maximum sense voltage
- Maximum current
- Maximum power

Each channel can be assigned to one output group. The reaction at any failure can be selected independently:

- Ignore the failure (not possible in case the power supply might get damaged)
- Switch the channel OFF
- Switch all channels with the same group number OFF
- Switch all channels of the Maraton Power Box OFF

C.3 Controller Area Network

C.3.1 Relation with OSI levels

The Open System Interconnection Model is a reference to how messages are transferred in a network. All the communication protocols including CAN try to follow this model. When a message is transmitted through a communication network, it is being processed through the seven layers of the OSI model. The top layer is the Software or Hardware Application which is trying to send the message through the network. The seven layers in the OSI model are shown in Table C.1.

Level name	Description
Application	Defines the Communication Nodes and type of service and security
Presentation	Conversion of Data format into human readable format
Session	Initiation and maintenance of a Communication session
Transport	Error checking and verification of data reception
Network	Routing of data to proper destination
Data Link	Network synchronization and creating data packets
Physical	Transfer of bit stream into the network

Table C.1: OSI Model

Many communication bus protocols do not use all the seven layers of this OSI Model. Since CAN is a closed network it does not need to have security and to present the data in a user interface. Also it does not need to maintain sessions and logins. Hence it uses only two Layers such as Physical and Data Link Layer, as shown in Figure C.6. The Physical Layer ensures the physical connection between the nodes in the network. While the Data Link layer contains Frames and information to identify the frames and errors.

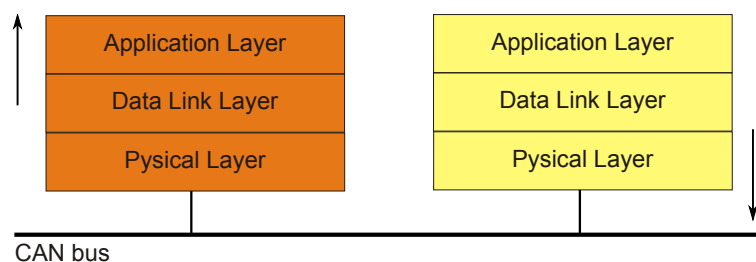


Figure C.6: Two Layers of CAN

C.3.2 CANopen

Although the usage of direct CAN for low level applications may be adequate, the ATLAS DCS (as initiators of the ELMB project) required a higher-level communication protocol on top of CAN. From the different higher-level communication protocols available, CERN has selected CANopen on the basis of flexibility and market acceptance. CANopen is defined by the CAN in Automation (CiA) organization and a detailed description of the protocol is provided in [GD97] and in [Var02].

This standard implements layer seven of the OSI communication model, the application layer. It also defines how the data-bytes of the CAN frames are used among the different nodes on the bus. The protocol also guarantees the interoperability of devices from different manufacturers standardizing:

1. The application layer, which defines common services and protocols for all devices.
2. The communication profile, which defines the communication between devices on the bus. The CANopen communication model implements both master-slave and slave-slave communications modules. In a CANopen network there must be at least one master application performing the boot-up process and maintaining the network in an operational state.
3. The so-called Device Profile Specifications (DSP), which specify the specific behavior for types of devices such that the inter-operability of devices from different producers and basic network operation are ensured.
4. Standardized description of the node functionality by means of the so-called Object Dictionary (OD). The OD is a standardized database, which holds an ordered collection of objects containing the description of the device and network behavior. The object dictionary is not stored in the CANopen node itself but defined in an Electronic Data Sheet (EDS). The OD is subdivided in three main areas: the Communication area, the Manufacturer Specific area and the Device Profile area.
5. Standardized communication objects. Two main types of mechanisms are implemented:
 - Process Data Objects (PDO), which are broadcasted and unconfirmed messages containing up to 8 data bytes. This mechanism is used for real-time transfers.
 - Service data objects (SDO), which are confirmed transfers of any length. Peer-to-peer communication is established between two nodes of the network by this mechanism. This type of objects are used for configuration of the nodes by direct accessing to their object dictionaries. PDOs are higher-priority messages than SDOs. This differentiation among real-time and configuration transfers and the arbitration collision mechanism of CAN, make this protocol specially suited for detector control since I/O operation affecting the status of the system are performed in real-time while lower-priority functions are always performed at low levels of bus occupancy.
6. Standardized Network Management (NMT) including system boot-up, node supervision and node identifier distribution. The CANopen master handles the transition of the slave between the possible states: initialization, pre-operational, operational, stopped. After the boot up sequence the device enters in pre-operational state. NMT messages are used to bring a slave node to operational or stopped states. Of particular relevance for LHC applications is the network supervision where the master node guards all remote slaves on the bus. There are two main mechanisms:
 - Node guarding, where the master produces guard messages that must be acknowledged by the slaves.
 - Heart-beat, where the slaves send messages to the bus at regular time interval that are consumed by the master. If the master does not receive a reply from a node within a certain time period the node is regarded as not functional and an error flag may be set.

7. Standardized systems services for synchronous operation of the network and handling of error messages from the nodes. Synchronous mode of operation is particularly interesting for detector control. The network master can query current values of the input channels by issuing a SYNC command to the bus. If the input PDO of the slave nodes have been configured to be operated synchronously, they are transmitted as response to a SYNC to the bus. This permits to obtain a snap-shot of the detector at regular time intervals. This is useful for time response analysis.

C.3.3 Sysworxx CAN to USB adapter

The Multiport CAN-to-USB 3004006 is an industrial USB-CAN interface with 16 CAN-channels, [SYS07]. The device is structured as 8 USB/CAN devices with 2 CAN-channels each. The logical devices are combined by 2 USB-hubs and connected to the PC via two USB ports (see Figure C.7).

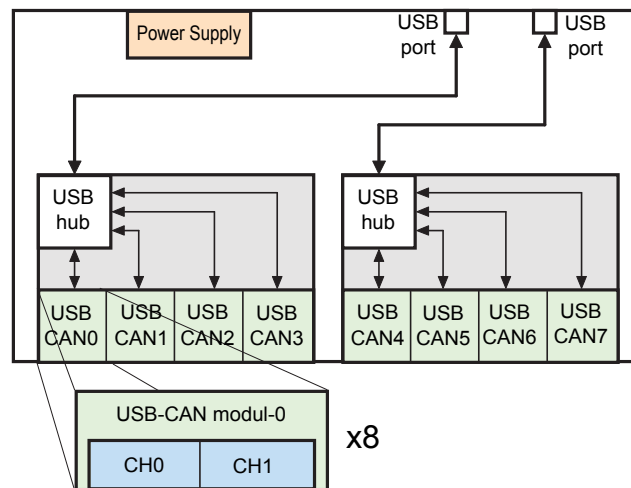


Figure C.7: Internal structure of the Multiport CAN-to-USB

A library to emulate the Kvaser+ interface is also provided. In this way the CAN-USB module is detected as a Kvaser PCI interface. A special API function set was implemented to support the extended functions of the Multiport CAN-to-USB, such as multiple CANchannels, baud rate configuration and acceptance mask filtering.

C.4 ELMB (Embedded Local Monitoring Boards)

C.4.1 Origins and objectives

The Embedded Local Monitor Board (ELMB) is a general-purpose I/O module for the monitoring and control of detector front-end equipment, [CT05]. This project is a collaboration between CERN, NIKHEF (The Netherlands) and PNPI (St. Petersburg, Russia). Initially they were developed for the ATLAS experiment, but later had been also used by the other experiments of LHC.

The general specifications of the design of the ELMB were the following:

- Radiation Tolerance to about 5 Gy and $3 \cdot 10^{10}$ neutrons cm^{-2}
- Operation in magnetic field up to 2 T

- Low power consumption, allowing for remote powering of the module via the bus CAN
- Low cost per I/O channel, due to the large number of nodes needed for the whole of the experiment (~ 5000)

The special feature of this device is that the firmware is stored in a read only memory. During the boot-up it is copied into a volatile memory for data acquisition. If radiation alters this volatile memory a reset (or power cycle) reloads the firmware from the ROM.

C.4.2 Software

The ELMB software is written in C and is divided into two main applications:

- The ELMBio, running on the master processor and handling the CANopen protocol, as well as the monitoring and control of the I/O channels. This package offers the flexibility required to support a wide range of applications. The program conforms to the Device Specification Profile (DSP) 301, where multiplexed PDO are used for the transmission of analogue input channels.
- The ELMBmon, which runs in the slave processor and provides the functionality needed for remote in system reprogrammability via the bus and a mechanism to prevent and correct against radiation effects.

The structure of the software allows for the operation of the ELMB as a finite state machine with four possible states:

- Initialization
- Stopped
- Preoperational, which is typically used to configure the node via the SDO protocol
- Operational

Transitions between these states are triggered by the appropriate NMT message. The monitoring of the I/O channels and the handling of PDO starts when the node enters in operational state. Since both the flash technology and EEPROM memories have been proven to be particularly robust to radiation effects, all constants are locally stored in these two types of memories. These parameters are reloaded after resetting of the node. All configuration parameters are stored in the Object Dictionary (OD) of the ELMB and can be accessed via SDO. In particular, the ADC is modelled, via software, by an object in the OD to allow for its configuration, in terms of type of measurement, gain, offsets, conversion rate, number of channels.

The real time data, thus the digital I/O signals and the analogue inputs are communicated via the PDO protocol. Digital outputs are transmitted asynchronously using the first Received PDO (RPDO1), whereas for the Digital Inputs both transmission types, synchronous and asynchronous are used. In this latter case, the first Transmitted PDO (TPDO1) containing two data bytes (PORTF and PORTA respectively), is used. The minimum time interval between two successive updates of the input channels, called debounce time, can also be set via software.

C.4.3 Hardware

Three different power regions identified, as shown in Figure C.9:

- the CAN controller part
- the digital part having two microprocessors
- the ADC (Analogue to Digital converter) region

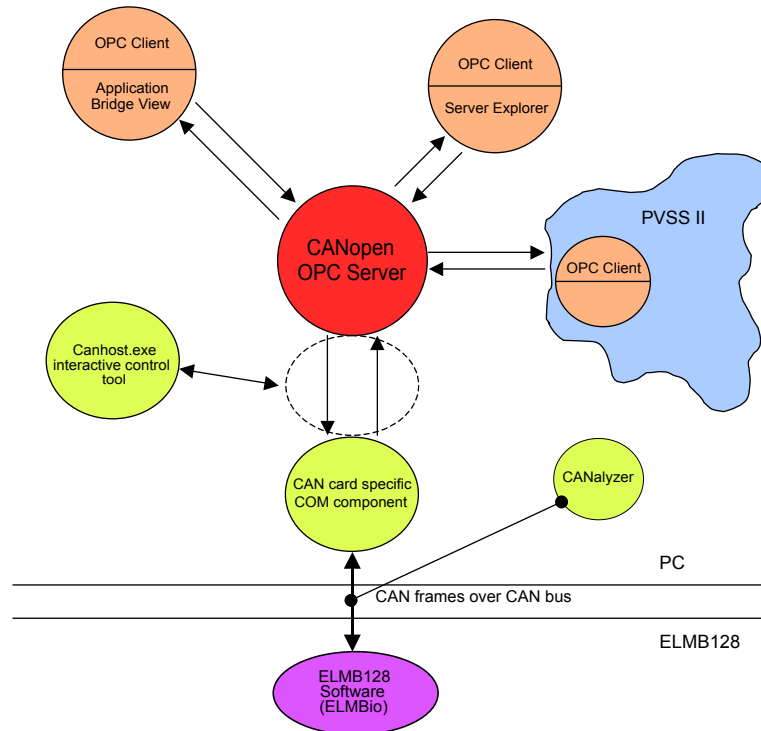


Figure C.8: Parts of the ELMB software

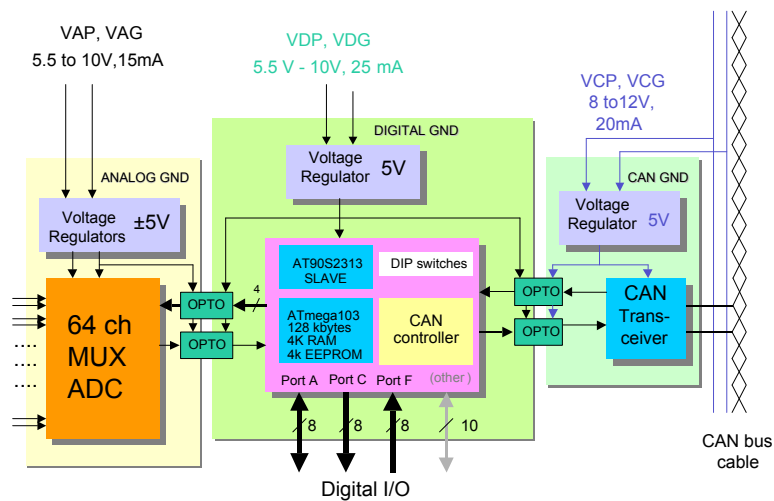


Figure C.9: Logic hardware design of the ELMB

AVR RISC architecture ATMEL Atmega128:

- 128 Kbytes of on chip flash memory
- 4 Kbytes of SRAM
- 4 Kbytes of EEPROM
- In-System Programming via CAN bus

Peripheral Features:

- Full CAN controller interface with PCA82C250
- 6 bit CAN identifier and 4 baud rates supported

- 3 wire SPI interface
- Real Time Counter with a separate 32 kHz crystal
- Timers
- 8 channel 10 bit ADC

I/O lines available:

- 6 external interrupt inputs
- *Port A*: 8 digital bi-directional I/O lines (can alternatively be used for external SRAM)
- *Port C*: 8 digital output lines (can alternatively be used for external SRAM)
- *Port D*: 5 digital bi-directional I/O lines
- *Port E*: 5 digital bi-directional I/O lines
- *Port F*: 8 digital input lines or 8 analog inputs for the ADC
- Strobe and enable lines for external SRAM

Optional Delta-sigma ADC CRYSTAL CS5523 with 64 channel multiplexer:

- 6 bipolar or unipolar input ranges from 25 mV to 5 V
- 100 pA input current on 25 mV, 55 mV and 100 mV
- 10 nA on 1 V, 2.5 V and 5 V ranges
- 8 conversion rates from 2 Hz to 100 Hz
- 64 channel multiplexer
- +5 V and -5 V on board power regulators

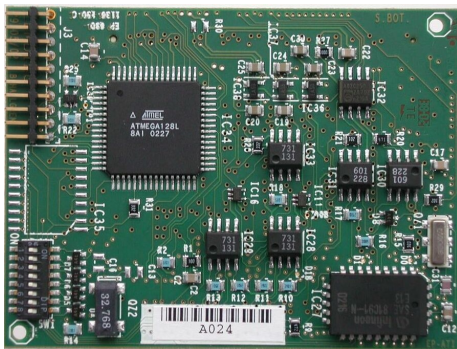


Figure C.10: Front side of the ELMB with the master and slave microcontrollers, CAN chip and DIP switches for node identification and setting of the baud rate.

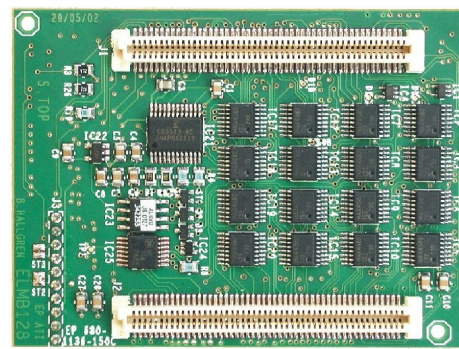


Figure C.11: Back side of the ELMB with the ADC, multiplexers and 100-pins SMD connectors

C.4.4 Motherboard

Although the ELMB can be directly embedded onto the front-end electronics of the detectors, it can also be used as an stand-alone I/O module by plugging it onto a multi-purpose motherboard. In the former case, the signals must be conditioned to the proper levels of the ADC by means of dedicated circuitry. In the latter, this adaptation can be done by using generic adapters, which are plugged onto the sockets of the motherboard.

On the front-side of the motherboard, the connectors for the ADC inputs, digital ports, SPI interface and power and bus connectors are located. The motherboard can be mounted on a DIN rail housing of size 80×180 mm. Each adapter serves groups of four channels. The ADC reference voltage (+2.5 V) and ground are also available on them.

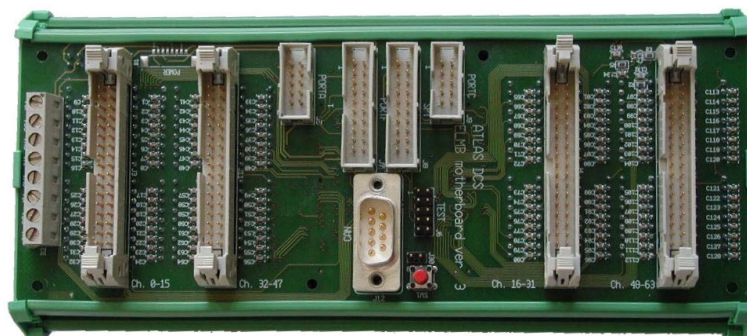


Figure C.12: The front side of the ELMB motherboard

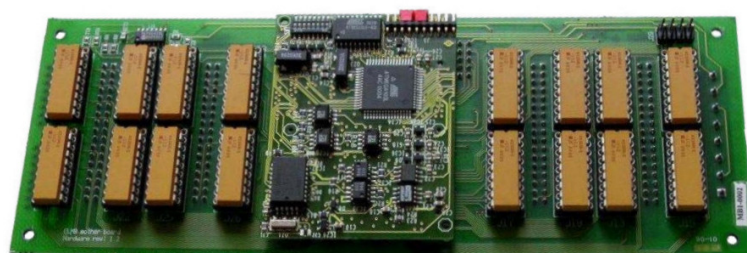


Figure C.13: The back side of the ELMB motherboard

C.4.5 Adapters

To connect the sensor to the ELMB, some electric adaptation is needed. For this purpose the resistor from the back of the ELMB (see Figure C.14) can replace to adapt to every different sensors. Examples of this are the Figure C.15 and Figure C.16.

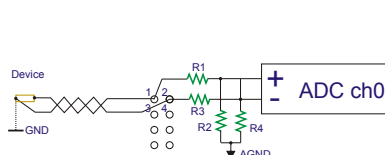


Figure C.14: Differential Attenuator

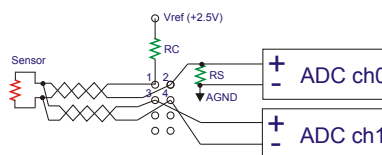


Figure C.15: 4-wires PT100

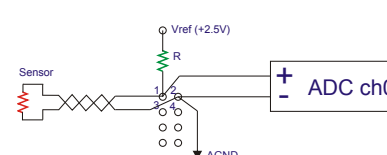


Figure C.16: 2-wires NTC or PT1000

C.5 PXI (PCI eXtensions for Instrumentation)

PXI (PCI eXtensions for Instrumentation) is a PC-based platform for measurement and automation systems. It was developed in 1997 and launched in 1998, as an open industry. PXI is governed by the PXI Systems Alliance (PXISA) [PXI98], a group of more than 65 companies chartered to promote the PXI standard, ensure interoperability, and maintain the PXI specification.

It combines the PCI electrical-bus features with the modular Eurocard packaging of CompactPCI, and then adds specialized synchronization buses and key software features.

Front end electronics

D.1 VFAT

The VFAT [AAB⁺07] is a trigger and tracking front-end ASIC, designed specifically for the readout of sensors in the TOTEM experiment at the LHC. The VFAT chip shown in Figure D.1 has been designed in quarter micron CMOS technology and measures 9.43 mm by 7.58 mm.

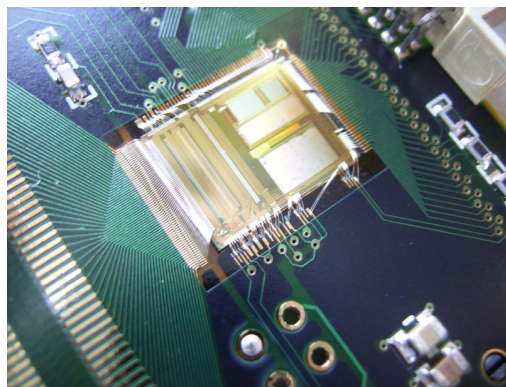


Figure D.1: Photograph of the VFAT chip

It has two main functions; the first one (Tracking) is to provide precise spatial hit information for a given triggered event. The second one (Trigger) is to provide programmable ‘fast OR’ information based on the region of the sensor hit. This can be used for the creation of a level-1 trigger.

Figure D.2 shows the block diagram for the signal path through the VFAT. It has 128 analog input channels which are equipped with a very low noise pre-amplifier and a 22 ns shaping stage plus comparator. A calibration unit allows delivery of controlled test pulses to any channel for calibration purposes. Signal discrimination on a programmable threshold provides binary hit information which passes through a synchronisation and monostable stage before being stored within SRAMs until a trigger is received. The monostable has a variable length from 1 to 8 clock periods. This has the effect

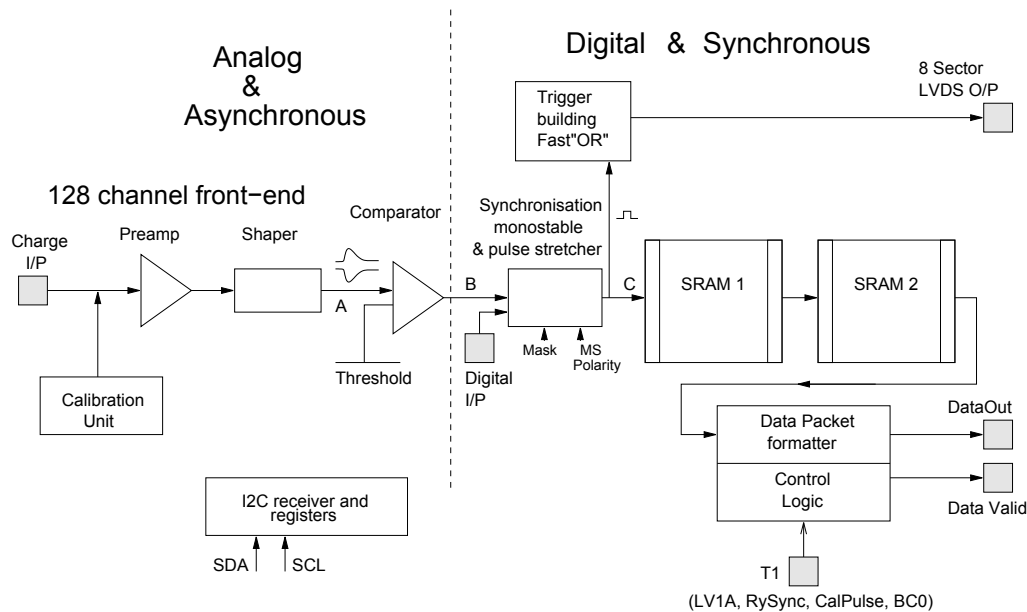


Figure D.2: Block diagram of the signal path through the VFAT

of recording the hit in more than one clock period (useful for gas detectors which have an uncertainty on the signal charge rise time). The SRAM storage capacity enables trigger latencies of up to $6.4 \mu\text{s}$ and the simultaneous storage of data for up to 128 triggered events. Dead time free operation with up to 100 kHz Poisson distributed trigger rates is ensured. Time and event tags are added to the triggered data which are then formatted and read from the chip in the form of digitized data packets at 40 Mbps. The data packet format is defined as in Figure D.3.

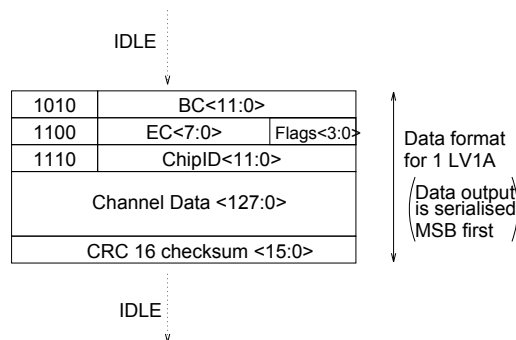


Figure D.3: VFAT data packet format

VFAT has many programmable functions controlled through an I²C interface. These include: internal biasing of analog blocks via 8 bit DACs, individual channel calibration via an internal test pulse with 8 bit programmable amplitude, calibration test pulse phase control, operate with positive or negative detector charge, 8 bit global threshold plus a 5 bit trim DAC threshold adjust on each channel, multiple possibilities for channel grouping for the ‘fast OR’ outputs, variable latency, various test modes plus an automatic self test of the digital memories.

For robustness against single event upsets (SEU), the digital parts of VFAT have been designed with hamming encoding for the SRAMs and triplication logic for the I²C interface and control logic. All analog circuitry employs layout techniques that reduce threshold voltage shifts under ionising radiation.

D.2 DCU

The Detector Control Unit (DCU) is an ASIC developed for the monitoring system for the CMS Tracker [MMM04]. Leakage currents in the Silicon detectors, power supply voltages of the readout electronics and local temperatures will be monitored in order to guarantee safe operating conditions. All these measurements can be performed by an A/D converter preceded by an analog multiplexer.

In TOTEM the DCU is integrated in the frontend electronics and readout through the DAQ system.

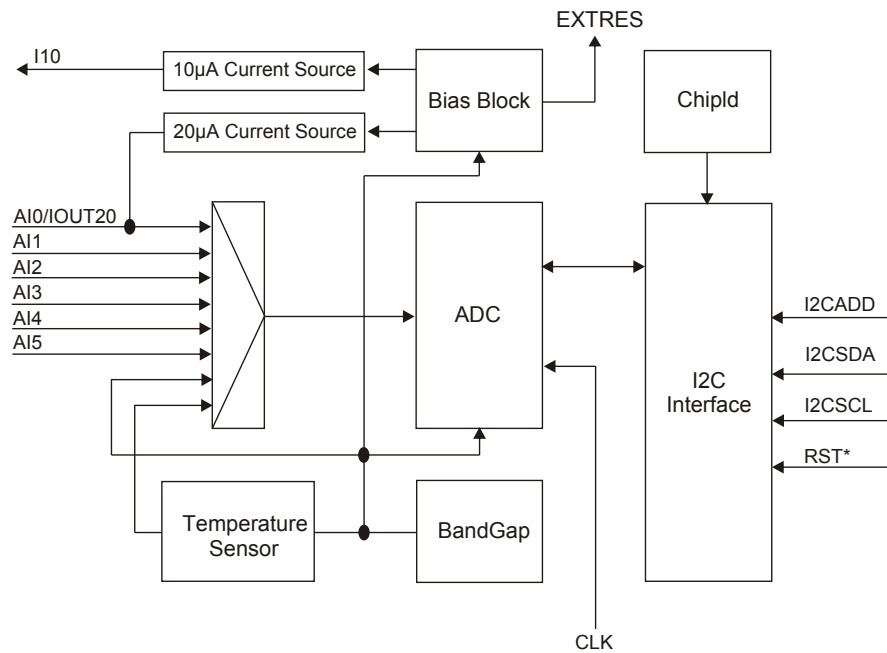


Figure D.4: The architecture of the DCUF is shown in the following block diagram

The DCU contains the following main blocks:

- a serial slave interface based on the standard I²C protocol
- a band-gap voltage reference
- an analogue multiplexer
- a 10 µA constant current source
- a 12-bit ADC
- one node controller (the CCU control itself is seen as a special channel capable for instance to report the status of the other CCU channels)
- an on-chip temperature sensor
- a 20 µA constant current source internally connected to one of the inputs of the analogue multiplexer
- a set of fuses that fixes a unique 24-bit Chip Identifier

The access to the internal registers of the DCU is available though an I²C interface. The user can select one of the ADC input channels, start an ADC acquisition and read the ADC output or the 24-bit chip identifier simply by accessing different I²C registers. A band-gap voltage reference gives to the ADC a stable reference voltage.

(This page is intentionally left blank)

Chapter

External systems

E.1 Detector safety system

E.1.1 Types of alarms

A ‘Level 1’ or ‘Level 2’ alarm indicates either an error condition or the occurrence of an undesired state of the detector or the other elements monitored by the DCS. ‘Level 1’ and ‘Level 2’, will be handled either by the Detector Safety System (DSS) or by the DCS. Most of these alarms will be handled by the DCS using the alarm-handling mechanism of PVSS II. An alarm is created when a parameter value leaves a definable value range and it is cleared when the value re-enters this range.

An alarm has several properties:

- source
- possibility/necessity of an acknowledgment by the operator as well as a specification of the delay of this acknowledgment
- severity level
- time of occurrence
- system state that led to the alarm
- additional details like a help text describing the alarm and a potential reaction or even cure

The highest level of safety, ‘Level 3’, is established for the most severe situations where a human life may be at threat. This critical level will be handled by the CERN Safety Alarm Monitoring (CSAM).

E.2 Relationship with other systems

For the LHC experiments there are three logically independent systems that act complementary to protect personnel and equipment. These ensure safety both in the surface buildings as well as in the underground areas; see [ABB⁺02b] and [LFMS03].

These systems are:

- **Detector Control System (DCS)**

The DCS is responsible for the overall monitoring and control of the detector. In normal circumstances, this should ensure that the experiment equipment is maintained in an operational state. This operational state should be, by definition, a safe state. In the event of a deviation from the desired condition, or of a fault being detected, the DCS would initiate corrective action to restore normal operation of the detector. This intervention could either be automatic or via user interaction. In such cases, the DCS would typically act on an individual piece of equipment, or a small set of equipment, in a very targeted manner. The DCS would try to avoid perturbing data taking whenever possible, even using equipment at the edge of its tolerance.

- **Detector Safety System (DSS)**

The DSS is responsible for safeguarding the experimental equipment. As such, it acts to prevent damage to the experimental equipment when a serious fault situation (e.g. too high temperature, water leak in a counting room) is detected, inside or outside of the detector, and will complement the functionality provided by the CSS and DCS. Whereas the DCS's prime responsibility is the supervision of the detector to maintain this in a correct operational state, the DSS is specially designed to protect the detector equipment. As such, the DSS will take responsibility for monitoring the environment of the detector with all the associated hazards and reacting to fault conditions as required in this specification. In comparison with the DCS, it will have a small number of inputs (of the order of a few hundred), but will incorporate all those required for its safety functions. It is expected that *DSS alarms* will occur only rarely. The DSS will typically perform rather coarse actions, e.g. switching off the power to a row of racks or to the whole counting room.

- **CERN Safety System (CSS)**

The CSS, which is comprised of the CERN Safety Alarm Monitoring System (CSAM) and the CERN Safety Equipment (CSE), is responsible for personal safety (Alarms-of-Level-3). As such, it is only indirectly concerned with equipment safety in as far as an equipment fault might lead to a situation endangering human life. In the case of a fault being detected by the CSS, the resulting action would typically be wide-ranging, such as cutting power to an area comprising several *DSS locations*, and the intervention of the CERN Fire Brigade.

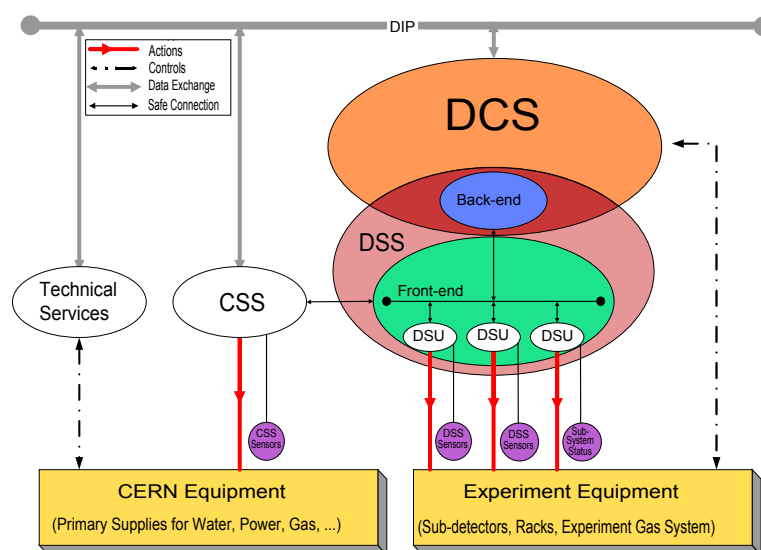


Figure E.1: Relationship between DCS and DSS

E.2.1 Architecture

At CERN, the DSS is common for all LHC experiments under the auspices of the Joint Controls Project (JCOP), is responsible for assuring the equipment protection for these experiments. Therefore, the DSS requires a high degree of both availability and reliability.

DSS frontend

The frontend is based on a redundant Siemens PLC, to which the safety-critical part of the DSS task is delegated. The PLC frontend is capable of running autonomously and of automatically taking predefined protective actions whenever required. It is supervised and configured by the CERN-chosen PVSS SCADA system via a Siemens OPC server.

The *DSS frontend* should accept inputs, not only from its own dedicated analogue and digital sensors, but also hardwired input signals directly from detectors, subsystems and the CSS. Some simple filtering on the signals, should be applied to avoid reacting on false signals. Also some basic processing on the input signals, such as AND, OR, limit checking,... is needed before triggering any action or alarm.

In the event of a fault being detected, an automatic action, *DSS action*, should be initiated immediately. The processing linking the input signals to output signals is known as the *Alarm/Action Matrix*. In some cases, the automated actions may be in the form of a sequence of individual actions. In such a sequence, it may be necessary to define a delay between subsequent actions, e.g. to ramp down voltages before switching off a set of racks. Nonetheless, for the majority of the cases the type of action taken will be rather coarse, e.g. switching of the power to a complete *DSS location*.

DSS backend

Its role is to provide a user interface, an interface to *External Systems*, a configuration mechanism and to perform data recording.

The *DSS backend* should be able to track variations and to store these, as well as issue warnings if values go outside predefined limits. The *DSS backend* should also display these parameters to the user, typically the SLIMOS, via dedicated displays. In addition to the basic displays, the *DSS backend* should provide tools for post-mortem and data analysis.

In the event of an alarm, the experiment control room operator should be notified. The *DSS backend* display should provide the user with all the possibilities necessary to identify and understand quickly the fault that has been detected. This can be achieved by automatically switching to a display showing the location of the detected fault, providing additional information related to that fault, as well as quick access to a specific help file for that fault situation. In the event of certain defined alarms arising, the *DSS backend* should be able to generate and send messages automatically to specified on-call experts, e.g. email, SMS, etc.

E.3 CRoss-platform DAQ Framework (XDAQ)

E.3.1 Introduction

XDAQ is a framework designed specifically by CMS for the development of distributed data acquisition systems in [CMS02]. It provides platform independent services, tools for local and remote inter-process communication, configuration and control, as well as technology independent data storage. To achieve these goals, the framework builds upon industrial standards, open protocols and libraries.

E.3.2 Executive framework

The distributed programming environment follows a layered middleware approach. The distributed processing infrastructure is made scalable by the ability to partition applications into smaller functional units that can be distributed over multiple processing units. In this scheme each computing node runs a copy of an executive that can be extended at run-time with binary plugin components.

The program exposes two types of interfaces:

- The *core interfaces*; that lie between the middleware and core plugin components, providing access to basic system functionalities and communication hardware. Core plugins manage basic system functions on behalf of the user applications, including network access, memory management and device access.
- The *application interfaces*; that provide access to the various functions offered by the core plugins and are placed between the middleware and the user application components as shown in Figure E.2.

Middleware services include information dispatching to applications, data transmission, exception handling facilities, access to configuration parameters, and location lookup (address resolution) of applications and services. Other system services include locking, synchronization, task execution and memory management. Applications communicate with each other through the services provided by the executive according to a peer-to-peer message-passing model. This allows each application to act both as a client and a server. The general programming model follows the event driven processing scheme where an event is an occurrence within the system. There is no need for a central place in which the incoming data have to be interpreted completely, and it is possible to add new functionality by defining new messages. It can be an incoming message, an interrupt, the completion of an instruction, like a direct memory access transfer, or an exception. Messages are sent asynchronously and trigger the activation of user-supplied callback procedures when they arrive at the receiver side.

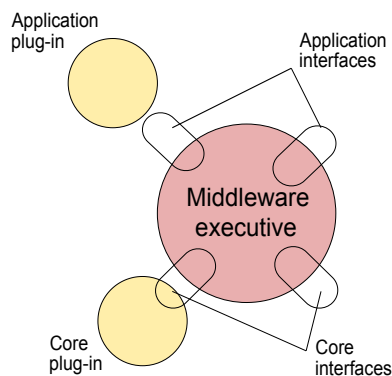


Figure E.2: Middleware interfaces

E.3.3 Application interfaces and state machines

One responsibility of the execution model is to keep track of the local states of all running applications. The overall behavior of an application in the scope of the executive framework can be modeled as a state machine that is defined for each individual application. In order to facilitate the development of such interfaces, a common set of states and transitions is included with all applications and executives. State changes can be initiated by control commands that are sent to the applications and executives.

The framework takes care of checking the consistency of the control commands. The success or failure of a state transition is reported to the initiator of the request. The current state of an application is a parameter that can be queried at any time from an external system. Dependencies between applications, which rely on state information are not foreseen and have to be managed by an external system (see Section E.4, *Run Control and Monitoring System*).

Both simple built-in data types as well as user-defined composite data structures are supported and the set of data types that are exported can be extended by the user. All functions defining the control interface of an application can be retrieved at run-time. This enables integration with external systems without the need to obtain a written specification of the application interface.

E.3.4 Protocols and data formats

The framework supports two data formats, one based on the I₂O specification and the other on XML.

I₂O messages are datagrams with a maximum size of 256 kB. For sizes larger than this maximum, the data have to be split and sent in a sequence of multiple frames. The framework provides mechanisms to perform this task. I₂O messages are primarily intended for the efficient exchange of binary information, e.g. data acquisition flow. No interpretation of such messages takes place in the executive. The messages are used directly by the applications to perform the required computation tasks. However, the content is platform-dependent and the application designer must obey all alignment and byte-ordering rules.

Despite its efficiency the I₂O scheme is not universal and lacks flexibility. A second type of communication has been chosen for tasks that require higher flexibility such as configuration, control and monitoring. This message-passing protocol, called Simple Object Access Protocol (SOAP) that relies on HTTP and encapsulates data using the eXtensible Markup Language (XML). The adoption of SOAP naturally leads to the use of Web Services that standardize the way in which applications export their interfaces to possible clients.

It is important to note that there is no limit to the data content type. Any text, binary or complex structure can be described in XML and any kind of protocol can be chosen to transmit these data.

E.4 Run Control and Monitoring System

E.4.1 Overview

The Run Control and Monitor System (RCMS) is the collection of hardware and software components responsible for controlling and monitoring the experiment during data taking [CMS02]. It provides physicists with a single point of entry to operate the experiment and to monitor detector status and data quality. The interface enables users to access and control the experiment from any part in the world providing a ‘virtual counting room’, where physicists and operators can perform all programmable actions on the system, effectively taking shifts from a distance.

In order to achieve its goals, the RCMS interoperates with the DCS, DAQ components and the trigger subsystem as shown, through the services provided by the distributed processing environment (XDAQ). For configuration, user administration and logging, the RCMS makes use of a database management system. The RCMS is the master controller of the DAQ when the experiment is taking data. It also instructs the DCS to act according to the specific needs of a data-taking session.

E.5 Architecture

The architecture of the DAQ system described in [CMS02] implies that there are roughly $O(100)$ objects that need to be controlled. The RCMS architecture is capable of scaling up to this order of magnitude by constructing hierarchies of distributed control applications. They consists of four types of elements, a Session Manager (SMR), a SubSystem Controller (SSC), a user interface and a set of services that are needed to support specific functions like security, logging and resource management.

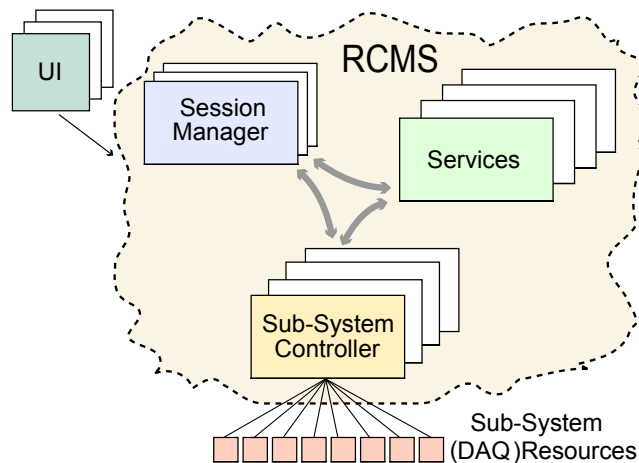


Figure E.3: Block diagram of the Run Control and Monitor System

The execution of a partition is defined as a ‘session’. A session is the allocation of all the hardware and software of a CMS partition needed to perform data-taking. Multiple sessions may coexist and operate concurrently. Each session is associated with a Session Manager (SMR) that coordinates all actions. The SMR accepts control commands and forwards them through a subsystem controller (SSC) to the components under its control. Single commands that are initiated by the user will be interpreted, expanded into sequences of commands where applicable, and routed to the proper subsystem resources by different RCMS components. Status information resulting from the actions corresponding to the commands submitted is expected asynchronously and is logged and analysed by the RCMS. The RCMS also handles all information concerning the internal status, malfunctions, errors and, when required, monitor data from the various DAQ subsystems.

The subsystems controlled by the RCMS, corresponding to the main elements of the DAQ are shown in Figure E.4.

The block diagram of the RCMS, showing the various services to support interaction with users and to manage subsystem resources, is shown in Figure E.3.

These services are:

- Security Service (SS). Provides login and user account management functions.
- Resource Service (RS). Provides access to the configuration database (ConfDB) that stores all the information about partitions and DAQ resources. The RS also handles the global ‘alive heart beat’ of the CMS DAQ.
- Information and Monitor Service (IMS). Collects messages and monitor data coming from DAQ resources or internal RCMS components and stores them in a database (LogDB). The IMS can distribute these messages to any external subscriber, with the possibility of filtering them according to a number of criteria, e.g. the message type, the level of severity and the source of the message. A similar mechanism is also used to distribute monitor data. The IMS also supports logbook facilities.

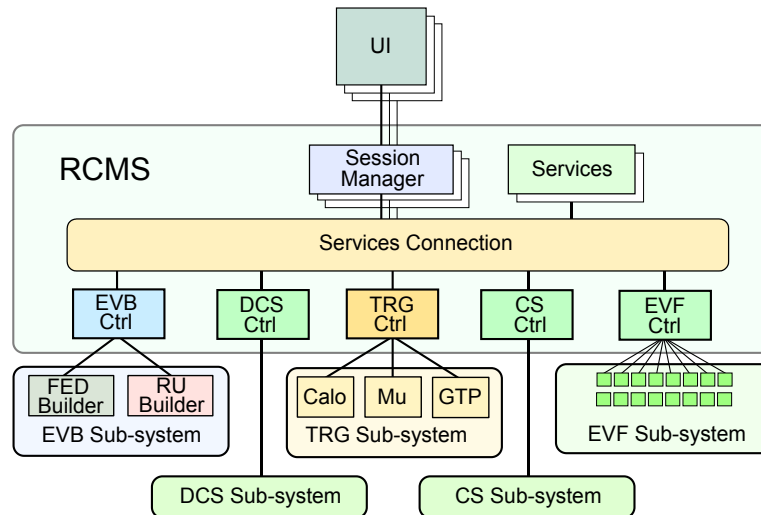


Figure E.4: Session Managers and SubSystems defined in the RCMS

- Job Control (JC). Starts, monitors and stops, when necessary, the software elements of the RCMS, including the data acquisition components.
- Problem Solver (PS). Uses information from the RS and IMS to identify malfunctions and attempts to provide automatic recovery procedures where applicable.

Subsystem	Resources defined
Event Builder (EVB)	FED Builder, RU Builder (RU, BU and EVM)
Event Filter (EVF)	Filter Units
Trigger (TRG)	Calorimeter Trigger, Muon Trigger, Global Trigger
Detector Control System (DCS)	Values accessible through the SCADA product
Computing Services (CS)	Storage, Monitor Services

Table E.1: Subsystems and their resources, as defined in the RCMS

E.5.1 RCMS and DAQ operation

The CMS data acquisition system can be operated in two exclusive ways:

- Partitioned mode, in which one or more DAQ partitions are used. All subsystems are partitioned according to the specific data-taking requirements, e.g. calibration runs, debugging of a specific detector. In this mode, multiple DAQ sessions can run concurrently. A special case of this mode is a ‘physics run’ in which all operational detectors are used in a single partition. A physics run includes all five DAQ elements, namely the trigger, the event builder, the event filter subsystems, the detector control system and computing services.
- Stand-alone mode, in which the detector partitions work independently using an individual trigger and individual data acquisition system. FED supervisor processors will be equipped with XDAQ software both to control and setup the FEDs and to acquire data when they are working in stand-alone mode. In this case BU and FU functionalities (including event storage capability) can be embedded in the FED supervisor.

State definitions and system synchronization

All RCMS subsystems can handle basic commands like ‘halt’, ‘configure’, ‘enable’, ‘disable’, ‘suspend’ and ‘resume’ that correspond to the state transitions defined for the XDAQ executive. The responses to the execution of these commands are reported to the IMS. All subsystems implement a state machine similar to the one in Figure E.5.

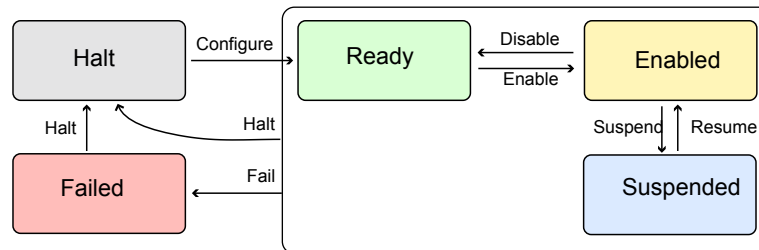


Figure E.5: State diagram of a generic DAQ component

Run definition

A data run is defined as the interval between the starting and stopping of data taking. Traditionally, a new run was started either because of major changes in run conditions or because of DAQ operations. Run conditions include such information as calibration constants and trigger tables, whereas DAQ resets and tape changes are examples of DAQ operations.

Sessions and partitions

The detectors, such as the muon detectors, tracker and calorimeters, represent ‘natural’ partition boundaries within the DAQ system. These partitions are reflected in those DAQ subsystems that have to deal directly with the detectors and their electronics like DCS and the event builder. These physical partitions impose constraints on the definition of the logical partitions that will be stored in a database. The same is true for the trigger where the partitions are imposed by the TTC distribution topology. However, this is not the case for the event filter where partitions are freely configurable, as they are only influenced by the computing power required by the session to which they belong.

The DCS, from the point of view of the RCMS is an external system with its own independent partition management. However, during data-taking, the RCMS instructs the DCS to set up and monitor partitions corresponding to the detector elements needed for the data-taking run in question. The RCMS also has access to all partition information in the DCS.



Additional resources

F.1 bigbrother.cern.ch

This is the dedicated server for the project. It serves the webpage of iDoc, acts as the subversion repository as a DIM name server. Future services can be configured if needed.

Because of the criticalness of the files stored, daily backups are done internally to another hard disk in the same machine, and also the main backup tool of CERN makes a copy of the files.

The authentication mechanism is linked to the main Active Directory of CERN for avoiding the maintenance of user accounts.

F.2 SubVersion

Subversion is a centralized system for sharing information. At its core is a repository, which is a central store of data. The repository stores information in the form of a filesystem tree. Any number of clients connect to the repository, and then read or write to these files. By writing data, a client makes the information available to others; by reading data, the client receives information from others. Figure F.1 illustrates this.

What makes the Subversion repository special is that it remembers every change ever written to it: every change to every file, and even changes to the directory tree itself, such as the addition, deletion, and rearrangement of files and directories.

When a client reads data from the repository, it normally sees only the latest version of the filesystem tree. But the client also has the ability to view previous states of the filesystem. For example, a client can ask historical questions like, ‘What did this directory contain last Wednesday?’ or ‘Who was the last person to change this file, and what changes did he make?’ These are the sorts of questions that are at the heart of any version control system: systems that are designed to track changes to data over time.

The typical work cycle by a user in a client machine looks like this:

- Update the working copy in the local machine.

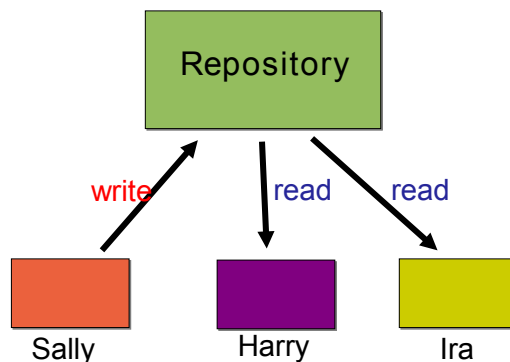


Figure F.1: A typical client/server system

- Make changes
- Examine the changes
- Possibly undo the changes
- Resolve Conflicts (merge others' changes)
- Commit the changes

Subversion also has some support for different releases of a product. It is possible to have a main development folder in the repository, and every time that there is a new release, copy this folder into another one. But the history of the file is maintained with the copy.

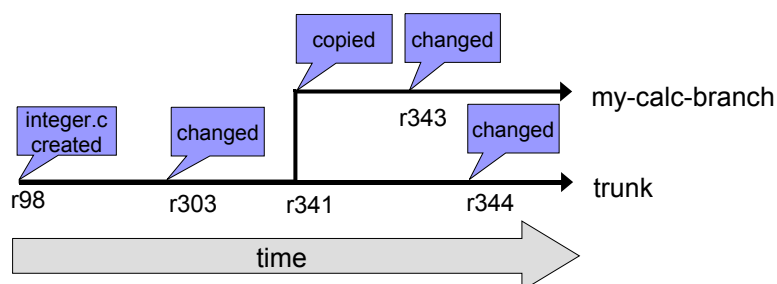


Figure F.2: Branches in Subversion

Subversion is used in the Big.Brother project for all the generated documentation and developments:

- iDoc: the maintenance of this application is not an objective of the TOTEM DCS, but for a better usability some tweaks in the code are needed over the time. The source code is managed in a subversion repository, but this repository is also mounted as a local filesystem in the web server. In this setup any change done committed by subversion will be immediately reflected in the webpage.
- Documentation: all the documents produced are also managed by subversion.
- PVSS: all the project of PVSII. This is a quite difficult part to maintain due the constantly change of the log files and file-databases used by the projects. This is mainly solved ignoring some of the directories when committing and updating.

F.3 iDoc

iDOC is dedicated web application written in PHP that represents a directory tree on the file system as a structured web document. Authors create content by populating the source directory tree, respecting the rules described here. A web page corresponds to a source directory.

All other files, except for specific control files, placed in each source directory are considered to be leaf nodes of the web document and linked from the directory's web page. These files may also be thought as an attachments. All subdirectories correspond to child web pages.

The control file is a plain text file in a similar syntax to the wiki.

The result can be observed in the main web of the project: <http://www.cern.ch/bb/>, while a dedicated page for this utility is in <http://www.cern.ch/idoc/> and in Figure F.3.

Figure F.3: Big.Brother website

F.4 TWiki

TWiki is a variant of a Wiki, a kind of software collaboration tools.

Users can edit the webpages freely after authentication. This allows the updating of documentation very quickly. It is usually used at CERN for the following tasks:

- designing and documenting software projects
- developing a knowledge base and FAQ system
- scheduling events by using calendar features

- operating an internal message board
- tracking issues, bugs and features
- managing documents
- archiving software
- writing and storing minutes of meetings

But is possible to check that in the Big.Brother project most of its key features are superseded by more proper tools.

The line between TWiki and iDoc+SVN is not clear. Twiki does not needs a local working copy (can be edited directly form the web). But for maintaining lots of attached files and nested levels, iDOC is more suitable.

F.5 HyperNews

HyperNews it is a cross between the WWW and usual newsgroups. Readers can browse through the messages written by other people and reply to those messages. A forum (also called a base article) holds a tree of these messages, displayed as an indented outline that shows how the messages are related.

Users can become members of HyperNews or subscribe to a forum in order to get email whenever a message is posted, so they do not have to check if anything new has been added. This is the major difference when comparing to the web based forums. Every web forum was a mail address associated. A recipient can then send a reply email back to HyperNews, rather than finding a browser to write a reply, and HyperNews then places the message in the appropriate forum.

So the dilemma of using forum or mailing lists is up to the user configuration. In the server level both as coupled as a single one service. It is accessible from <http://hn.cern.ch>

F.6 JIRA

JIRA is a commercial product developed by Atlassian. The objectives are to be a bug tracking, issue tracking, and project management application developed to make this process easier for your team. JIRA has been designed with a focus on task achievement, is instantly usable and is flexible to work with.

Some of its features are:

- Manage bugs, features, tasks, improvements or any issue
- A clean and powerful user interface that is easy to understand
- Map your business processes to custom workflows
- Track attachments, changes, components and versions
- Full text searching and powerful filtering
- Customisable dashboards and real-time statistics
- Enterprise permissioning and security
- Easily extended to and integrated with other systems (including email, RSS, Excel, XML and source control)
- Highly configurable notification options
- Web service enabled for programmatic control (SOAP, XML-RPC and REST interfaces)

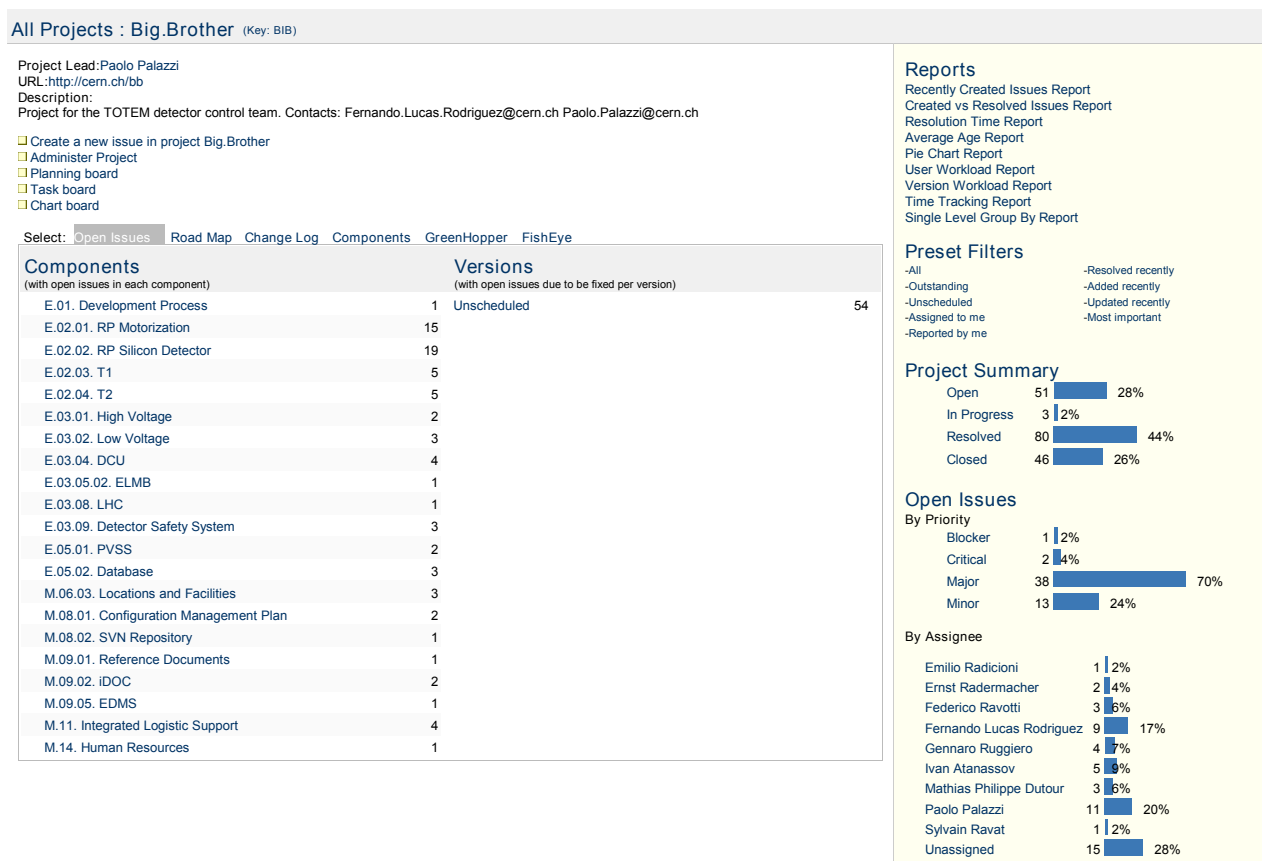


Figure F.4: JIRA interface

F.7 Photo Gallery

A photo gallery was habilitated for easy referral to all the cables and other pieces of hardware being installed and its current status. Over the time this gallery became TOTEM wide and used by the outreach members for promotion of the experiment. It is accessible from <http://www.cern.ch/totem-gallery/>

F.8 eLog

ELog is an opensource utility to log all the event that had taken place during shift operation. The TOTEM installation is accessible from <https://bigbrother.cern.ch/elog/>

F.9 CMS remote access

CMS Central DCS provides two ways to access/interact with the production system:

- A remote desktop to a Windows Terminal Server and exporting dynamically to that machine the PVSS panels running in the DCS computing nodes.
- A web interface as shown in Figure F.5. This interface show and control all the PVSS managers, is able to monitor and kill running processes in each node and shows the PVSS logs. Showing the current installed versions of PVSS components and upgrading some of them was been implemented following a TOTEM DCS request.



[Common](#)
[ECAL](#)
[HCAL](#)
[RPC](#)
[CSC](#)
[Tracker](#)
[Trigger](#)
[Services](#)

UTC: 23:43:33 1-Dec-08 Home Help
local: 00:43:33 2-Dec-08 WelcomeLoginLogout Elogbook
page loaded:00:43:07 2-Dec-08

- CMS-CENT-DCS-01
- CMS-ALIG-DCS-01
- CMS-CENT-DCS-02
- CMS-CENT-DCS-03
- CMS-CENT-DCS-04
- CMS-CENT-DCS-10
- CMS-CENT-DCS-11
- CMS-CENT-DCS-12
- CMS-CENT-DCS-13
- CMS-CENT-DCS-19
- CMS-CSC-DCS-09
- CMS-DT-DCS-04
- CMS-ECAL-DCS-01
- CMS-HCAL-DCS-02
- CMS-PIX-DCS-01
- CMS-RC-DCS-01
- CMS-RC-DCS-02
- CMS-RC-DCS-03
- CMS-RPC-DCS-01
- CMS-SCR-DCS-01
- CMS-SCR-DCS-02
- CMS-TRG-DCS-01
- CMS-TRK-DCS-01
- TOTEM-DCS-01
 - TOTEM-DCS-02
 - TOTEM-DCS-03
 - TOTEM-DCS-04
 - TOTEM-DCS-05

TOTEM-DCS-01

[Pmon](#)
[Processes](#)
[Log](#)
[Configuration](#)

PVSS project management

System name	System number	Inst. tool version	Management	Last checked
totem_dcs_01	245	3.4.1	Central	00:42 02-DEC-2008

Installed components

Component name	Component version	Overwrite files	Force required	Restart on install	Installation date	Installation result
CMSfwAlertSystem	1.1.0	NO	NO	NO	06-NOV-08	OK
CMSfwPerformance	0.0.1	NO	NO	NO	16-JUL-08	OK
CMSfw_CAENOPCCConfigurator	1.0.4	NO	NO	NO	06-NOV-08	OK
fwAccessControl	3.2.1	NO	NO	NO	01-JUL-08	OK
fwAlarmHandling	3.2.0	NO	NO	NO	06-NOV-08	OK
fwAnalogDigital	3.2.0	NO	NO	NO	16-JUL-08	OK
fwCaen	3.2.0	NO	NO	NO	06-NOV-08	OK
fwConfigs	3.1.1	NO	NO	NO	01-JUL-08	OK
fwConfigurationDB	3.3.51	NO	NO	NO	25-JUL-08	OK
fwCore	3.2.0	NO	NO	NO	01-JUL-08	OK
fwDIM	16.02.0	NO	NO	NO	06-NOV-08	OK
fwDIP	3.2.1	NO	NO	NO	06-NOV-08	OK
fwDevice	3.2.0	NO	NO	NO	01-JUL-08	OK
fwDeviceEditorNavigator	3.2.0	NO	NO	NO	01-JUL-08	OK
fwElmb	3.5.3	NO	NO	NO	01-DEC-08	OK
fwFSM	26.05.0	NO	NO	YES	06-NOV-08	OK
fwFSMConfDB	3.3.5	NO	NO	YES	01-JUL-08	OK
fwGeneral	3.2.0	NO	NO	NO	01-JUL-08	OK
fwLogErrorHandlerCMSmod	0.0.1	NO	NO	NO	16-JUL-08	OK
fwNode	3.2.0	NO	NO	NO	01-JUL-08	OK
fwRDBArchiving	1.1	NO	NO	NO	01-JUL-08	OK
fwTreeView	3.2.0	NO	NO	NO	01-JUL-08	OK
fwTrending	3.1.0	NO	NO	NO	01-JUL-08	OK
fwWiener	3.2.0	NO	NO	NO	06-NOV-08	OK
totCeHv	1.0.1	YES	NO	NO	19-NOV-08	OK
totRadmon	1.0.1	YES	NO	NO	19-NOV-08	OK
totServices	1.0.5	YES	NO	NO	19-NOV-08	OK
totUserInterface	3.0.6	YES	NO	NO	19-NOV-08	OK
unDistributedControl	3.2.1	NO	NO	NO	01-JUL-08	OK

System name	Number	Connected
cms_cent_dcs_01	1	No
totem_dcs_02	246	No
totem_dcs_03	247	No
totem_dcs_04	248	No
totem_dcs_05	249	No

Figure F.5: CMS online web interface



Radiation monitoring library

Here is included the PVSS library used for the initial RADMON commissioning. Notice that all the storage is done using arrays and not PVSS datapoints and the storage using text files and not the PVSS archiving mechanism.

However it includes a fully generic readout sequence with sensors timing, DAC state checks and auto-recovery capabilities.

```
// Control script written by F. Lucas Rodriguez
// Updated On 01-12-08

3  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Configuration of Parameters
//      -> This has to evolve into a radmon configuration DP type
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

8  const int switchDisable = 0;
// int switchEnable = 4095;
int switchEnable = 1023; // Current needed to turn on each of the FOUR DAC channels

13 const int pin1Disable = 0;
const int pin2Disable = 0;
const int radfet1Disable = 0;
const int radfet2Disable = 0;

18 const int defaultPin1Enable = 186;
const int defaultPin2Enable = 186;
const int defaultRadfet1Enable = 25;
const int defaultRadfet2Enable = 35;

23 const float NtcBeta = 3530.;

//int pin1Enable = pin1Disable ;
//int pin2Enable = pin2Disable ;
//int radfet1Enable = radfet1Disable;
28 //int radfet2Enable = radfet2Disable;
int pin1Enable = defaultPin1Enable ;
int pin2Enable = defaultPin2Enable ;
int radfet1Enable = defaultRadfet1Enable;
int radfet2Enable = defaultRadfet2Enable;

33 unsigned delayNtc = 0;
unsigned delayPin = 500;
unsigned delayRadfet = 1000;

38 const int maxReiterationsSwitch = 20;
const int maxReiterationsCurrent = 5;
const double currentConversionFactor = 0.01; // 100ohm RL resistors

// Delay changes as function of the ELMB ADC Rate (Hz)
43 //const int sdoDelay = 120; // 30hz
//const int sdoDelay = 225; // 15hz
const int sdoDelay = 315; // 8hz
//const int sdoDelay = 600; // 4hz

48 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Consistent IDs for array elements
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

const int sequenceNtc = 1;
53 const int sequenceSwitch = 2;
const int sequencePin1 = 3;
const int sequencePin2 = 4;
const int sequenceRadfet1 = 5;
const int sequenceRadfet2 = 6;
58 const int sequenceCount = 6;
const int posFlag = 1;
const int posTimeStamp = 2;
const int posElmb = 3;
const int posDac = 4;
63 const int posIsc = 5;
const int posDevice = 6;
const int posAo = 7;
const int posAisdoValue = 8;
const int posAisdoCurrent = 9;
```

```

68  const int posEnable           = 10;
    const int posDisable        = 11;
    const int posDelay          = 12;
    const int posValue          = 13;
    const int posCurrent        = 14;
73  const int posExpectedCurrent = 15;
    const int posExpectedDivergency = 16;
    const int posElapsed        = 17;
    const int posCount          = 17;

78  const string colorOk          = "green";
    const string colorError      = "red";
    const string colorUninitialized = "grey";
    const string colorMeasured   = "cyan";

83  //////////////////////////////////////
    // In fuction of the DAC number and the ISC number it defines the starting
    // values for Ao and AiSdo that go on the variable "baseao" and "baseaisdo"
    // these are outputs because of "&". In the second part, the last channel is also
    // identified for all cases in order to have the 4 channels for the switch array.
88  // "targetElmb" is never used apart if channel in the ELMB is broken. However,
    // it carries the information of which ELMB is in use.
    //////////////////////////////////////

    bool totRadmon_calculateDaclscFromChannel(string targetElmb, int &DAC, int
        &ISC, dyn_int &AoDac, int &AoChannel, int &AiCurrent)
93  {
    int AiBase = 0;

    // PATCH PANEL--ELMB CONNECTIVITY
    switch (channel)
98  {
        case 0:
        case 1:
        case 2:
        case 3:
103    case 4:
        case 5:
            DAC=1;
            ISC=3;
            AiBase=0;
108    break;

        case 16:
        case 17:
        case 18:
113    case 19:
        case 20:
        case 21:
            DAC=2;
            ISC=3;
118    AiBase=16;
            break;

        case 32:
        case 33:
123    case 34:
        case 35:
        case 36:
        case 37:
            DAC=1;
            ISC=1;
            AiBase=32;
            break;

128
        case 40:
133    case 41:

```

```

    case 42:
    case 43:
    case 44:
    case 45:
138     DAC=1;
        ISC=2;
        AiBase=40;
        break;

143     case 48:
    case 49:
    case 50:
    case 51:
    case 52:
148     case 53:
        DAC=2;
        ISC=1;
        AiBase=48;
        break;

153     case 56:
    case 57:
    case 58:
    case 59:
158     case 60:
    case 61:
        DAC=2;
        ISC=2;
        AiBase=56;
163     break;

    default:
        return false;
}

168 // PP-DAC--ELMB LOGIC
    int AoDacBase = (DAC-1)*16;
    AoDac = makeDynInt(AoDacBase+12,AoDacBase+13,AoDacBase+14,AoDacBase+15);
    AoChannel = (AoDacBase+(ISC-1)*4)+(channel-AiBase);
173     AiCurrent = AiBase+2;

    return true;
}

178 bool totRadmon_calculateAoCurrentFromDacIsclsc(string targetElmb, int DAC, int ISC, dyn_int
    &AoDac, int &AoChannel, int &AiCurrent)
{
    int AiBase;

    switch (DAC)
183     {
        case 1:
            switch (ISC)
            {
                case 1:
188                 AiBase = 48;
                    break;
                case 2:
                    AiBase = 56;
                    break;
193                 case 3:
                    AiBase = 16;
                    break;
                default:
                    return false;
198             }
    }
```

```

        break;
    case 2:
        switch (ISC)
        {
203         case 1:
            AiBase = 32;
            break;
            case 2:
208             AiBase = 40;
            break;
            case 3:
                AiBase = 0;
                break;
            default:
213                 return false;
        }
        break;
    default:
        return false;
218 }

// PP-DAC- -ELMB LOGIC
int AoDacBase = (DAC-1)*16;
AoDac = makeDynInt(AoDacBase+12,AoDacBase+13,AoDacBase+14,AoDacBase+15);
223 AoChannel = (AoDacBase+(ISC-1)*4);
AiCurrent = AiBase+2;

return true;
}

228 ////////////////////////////////////////////////////
// Here it calculate the readout sequences
// Checking the DAC and ISC parameters if channel in the elmb is broken
// tweaking can be done for a Pach Pannel that is using the 'recovered' channels
233 // on the 4th ISC connector
bool calculateSequences(string targetElmb, int DAC, int ISC, dyn_dyn_anytype &sequences)
{
    // Calculation
    int aisdoCurrent;

238     if (totRadmon_calculateFromDacIsc(targetElmb, DAC, ISC, aisdoCurrent) == false)
    {return false;}

    int ChannelPin1    = aisdoCurrent -2;
243     int ChannelRadfet1 = aisdoCurrent -1;
    int ChannelRI      = aisdoCurrent;
    int ChannelNtc     = aisdoCurrent+1;
    int ChannelPin2    = aisdoCurrent+2;
    int ChannelRadfet2 = aisdoCurrent+3;
248

    dyn_anytype Switch = makeDynAnytype();
    dyn_anytype Ntc     = makeDynAnytype();
    dyn_anytype Pin1    = makeDynAnytype();
    dyn_anytype Pin2    = makeDynAnytype();
253     dyn_anytype Radfet1 = makeDynAnytype();
    dyn_anytype Radfet2 = makeDynAnytype();

    mixed uninitialized;
    // the variable type "mixed" is used to manage exceptions in the fW
258     // contrary to "anytype" the variable type "mixed" gets each time a new type: if OK is
    // a float
    // if NOT OK is an array that contains warning and errors.

    // here it generates an array for each sensor of lenght defined by "posCount" and data
    // of "mixed" type
    // in the present version there are 17 positions

```

```

263 // position of differet data are defined at the beginning
    for (int i=1;i<=posCount;i++)
    {
        dynAppend(Switch, uninitialized);
        dynAppend(Ntc, uninitialized);
268     dynAppend(Pin1, uninitialized);
        dynAppend(Pin2, uninitialized);
        dynAppend(Radfet2, uninitialized);
        dynAppend(Radfet1, uninitialized);
    }
273 // the "Flag" entry interacts with the user interface.
    // "false" indicate that the script is busy and that the UI doesn't have to be updated

    Ntc[posFlag] = false;
    Ntc[posElmb] = targetElmb;
278     Ntc[posDac] = DAC;
    Ntc[posIsc] = ISC;
    Ntc[posDevice] = "Ntc";
    Ntc[posAo] = false;
    Ntc[posAisdoValue] = ChannelNtc;
283     Ntc[posAisdoCurrent] = aisdoCurrent; //false;
    Ntc[posEnable] = false;
    Ntc[posDisable] = false;
    Ntc[posDelay] = delayNtc;
    Ntc[posExpectedCurrent] = 2;
288     Ntc[posExpectedDivergency] = 3.0;

    Switch[posFlag] = false;
    Switch[posElmb] = targetElmb;
    Switch[posDac] = DAC;
293     Switch[posIsc] = ISC;
    Switch[posDevice] = "Switch";
    Switch[posAo] = totRadmon_calculateSwitchAo(targetElmb, DAC);
    Switch[posAisdoValue] = false;
    Switch[posAisdoCurrent] = false;
298     Switch[posEnable] = switchEnable;
    Switch[posDisable] = switchDisable;
    Switch[posDelay] = false;
    Switch[posValue] = false;
    Switch[posCurrent] = false;
303     Switch[posExpectedCurrent] = false;
    Switch[posElapsed] = false;
    Switch[posExpectedDivergency] = false;

    Pin1[posFlag] = false;
308     Pin1[posElmb] = targetElmb;
    Pin1[posDac] = DAC;
    Pin1[posIsc] = ISC;
    Pin1[posDevice] = "Pin1";
    Pin1[posAo] = baseao +0;
313     Pin1[posAisdoValue] = ChannelPin1;
    Pin1[posAisdoCurrent] = aisdoCurrent;
    Pin1[posEnable] = pin1Enable;
    Pin1[posDisable] = pin1Disable;
    Pin1[posDelay] = delayPin;
318     Pin1[posExpectedCurrent] = 1000;
    Pin1[posExpectedDivergency] = 0.5;

    Pin2[posFlag] = false;
    Pin2[posElmb] = targetElmb;
323     Pin2[posDac] = DAC;
    Pin2[posIsc] = ISC;
    Pin2[posDevice] = "Pin2";
    Pin2[posAo] = baseao +2;
    Pin2[posAisdoValue] = ChannelPin2;

```



```

328  Pin2[posAisdoCurrent      ] = aisdoCurrent;
    Pin2[posEnable           ] = pin2Enable ;
    Pin2[posDisable          ] = pin2Disable;
    Pin2[posDelay             ] = delayPin;
    Pin2[posExpectedCurrent   ] = 1000;
333  Pin2[posExpectedDivergency] = 0.5;

    Radfet1[posFlag           ] = false;
    Radfet1[posElmb           ] = targetElmb;
    Radfet1[posDac             ] = DAC;
338  Radfet1[posIsc            ] = ISC;
    Radfet1[posDevice          ] = "Radfet1";
    Radfet1[posAo              ] = baseao +1;
    Radfet1[posAisdoValue      ] = ChannelRadfet1;
    Radfet1[posAisdoCurrent     ] = aisdoCurrent;
343  Radfet1[posEnable         ] = radfet1Enable ;
    Radfet1[posDisable         ] = radfet1Disable;
    Radfet1[posDelay           ] = delayRadfet;
    Radfet1[posExpectedCurrent  ] = 140;
    Radfet1[posExpectedDivergency] = 0.5;
348

    Radfet2[posFlag           ] = false;
    Radfet2[posElmb           ] = targetElmb;
    Radfet2[posDac             ] = DAC;
    Radfet2[posIsc            ] = ISC;
353  Radfet2[posDevice          ] = "Radfet2";
    Radfet2[posAo              ] = baseao +3;
    Radfet2[posAisdoValue      ] = ChannelRadfet2;
    Radfet2[posAisdoCurrent     ] = aisdoCurrent;
    Radfet2[posEnable          ] = radfet2Enable ;
358  Radfet2[posDisable         ] = radfet2Disable;
    Radfet2[posDelay           ] = delayRadfet;
    Radfet2[posExpectedCurrent  ] = 170;
    Radfet2[posExpectedDivergency] = 0.5;

363  // All the above arrays are integrated in the array "sequences".
    // The prvious 6 arrays are here ordered in a sequence.

    sequences = makeDynAnytype();
    for (int i=1;i<=sequenceCount;i++)
368  {dynAppend(sequences, false);}
    sequences[sequenceSwitch ] = Switch;
    sequences[sequenceNtc     ] = Ntc;
    sequences[sequencePin1    ] = Pin1;
    sequences[sequencePin2    ] = Pin2;
373  sequences[sequenceRadfet1] = Radfet1;
    sequences[sequenceRadfet2] = Radfet2;

    return true; // returns all "true" in output of the function
}
378 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    // Basic readout function
    ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void readISC(dyn_dyn_anytype &sequences, bool power)
383 {
    if (power)
    {
        // if "power" is "true" means --> DAC was enable (it shouldn't): then it runs "
        // setDAC" (see below) to try to disable it
        setDAC(sequences, false);
388    }
    // after "setDAC" run with "false" condition --> now it is possible to read NTC
    readNtc (sequences[sequenceNtc ],power);

    if (power) // if DAC is disable

```

```
393 {
    // DAC enable
    setDAC(sequences, true);
}
// now it is possible to read the sensors
398 readPin (sequences[sequencePin1 ], power);
readPin (sequences[sequencePin2 ], power);
readRadfet(sequences[sequenceRadfet1], power);
readRadfet(sequences[sequenceRadfet2], power);

403 if (power)
{
    // DAC disable
    setDAC(sequences, false);
}
408 }

// easy readout of a NTC
void readNtc(dyn_anytype &sequence, bool power)
{
413 readSequence(sequence, power);
sequence[posValue ] = 1./297.+1./NtcBeta*(log((sequence[posValue ]/2.5)/10000.));
sequence[posValue ] = 1./sequence[posValue ] - 273.15;
}

418 // easy readout of a PIN
void readPin(dyn_anytype &sequence, bool power)
{
    readSequence(sequence, power);
    sequence[posValue ] = sequence[posValue ]/1000000*11;
423 }

// easy readout of a RadFET
void readRadfet(dyn_anytype &sequence, bool power)
{
428 readSequence(sequence, power);
sequence[posValue ] = sequence[posValue ]/1000000*11;
}

void setDAC(dyn_dyn_anytype &sequences, bool action)
433 {
    string targetElmb = sequences[sequenceSwitch][posElmb];
    int ao = sequences[sequenceSwitch][posAo]; //it identifies the correct "ao" for the
        first switch channel
    int switchValue;

438 if (action)
    {
        // if "action" is true record time and "switchValue" is set to ENABLE
        sequences[sequenceSwitch][posTimeStamp] = getCurrentTime();
        switchValue = sequences[sequenceSwitch][posEnable ];
443 }
    else
    {
        // if "action" is false the "switchValue" is set to DISABLE
        switchValue = sequences[sequenceSwitch][posDisable];
448 }

    // iterate several times (up to "maxReiterationsSwitch") to set switches OFF using the
    proper function "setDACSwitches"
    for (int limit=0; limit<maxReiterationsSwitch; limit++)
    {
453 setDACSwitches(targetElmb, ao, switchValue);
        // ---> eventually here try to force also other DAC channels to close <---
        string dp;
```

```

double value,current;
dyn_string dsExceptionInfo;
458 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//      dp=sequences[sequenceNtc][posElmb]+"/AI/aisdo_"+sequences[sequenceNtc][
//      posAisdoValue ];
//      fwElmb_elementSQ(dp+".rawValue", 2, value, dsExceptionInfo);
//      if (dynlen(dsExceptionInfo) > 0)
463 //      {
//          DebugN(dp);
//          DebugN(dsExceptionInfo);
//      }
//      else
468 //      {
//          dpGet(dp+".value", value);
//      }
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

473 dp=sequences[sequenceNtc][posElmb]+"/AI/aisdo_"+sequences[sequenceNtc][
//      posAisdoCurrent]; // identify the proper RL
fwElmb_elementSQ(dp+".rawValue", 2, current, dsExceptionInfo); // RL current is
//      readout. The answer is wait for 2ms
if (dynlen(dsExceptionInfo) > 0)
{
478     DebugN(dp);
    DebugN(dsExceptionInfo);
}
else
{
483     dpGet(dp+".value", current);
    current = current*currentConversionFactor;
}

if (current==0)
{
488     // current flowing in RL = 0 --> OK
    // DebugN("Device "+dp+" seems disconnected; aborting switch verification");
    break;
}
else
493 {
    if (current>50)
    // current lowing in RL > 50 uA (since the "rawValue" is in uV)
    // here it tries to set OFF also the DAC channels of the current ISCs
    // ----> try to switch OFF all DAC channels !!!! <----
498 {
        DebugN("switch critical "+dp+" ; switchValue "+switchValue+" ; current "+current
            +" ; "+limit);
        setSwitch(sequences[sequencePin1][posElmb]+"/AO/ao_"+sequences[sequencePin1
            ][posAo]+".value", sequences[sequencePin1][posDisable]);
        setSwitch(sequences[sequencePin2][posElmb]+"/AO/ao_"+sequences[sequencePin2
            ][posAo]+".value", sequences[sequencePin2][posDisable]);
        setSwitch(sequences[sequenceRadfet1][posElmb]+"/AO/ao_"+sequences[
            sequenceRadfet1][posAo]+".value", sequences[sequenceRadfet1][posDisable]);
503 setSwitch(sequences[sequenceRadfet2][posElmb]+"/AO/ao_"+sequences[
            sequenceRadfet2][posAo]+".value", sequences[sequenceRadfet2][posDisable]);
        delay(5);
        if (limit<maxReiterationsSwitch -2)
        {
508             limit = maxReiterationsSwitch -2;
        }
        //setDACSwitches(targetElmb,ao,switchDisable);
        //delay(1);
        //setDACSwitches(targetElmb,ao,switchValue);
    }
}

```

```

513     //setDACSwitches(targetElmb,ao,switchDisable);
    }
    else
    {
        // if ( checkMeasurementDivergency(current,sequences[sequenceNtc][
        //      posExpectedCurrent],sequences[sequenceNtc][posExpectedDivergency]) )
        // {
518         break;
        // }
        // else
        // {
        //     DebugN("switch not responding "+dp+" ; switchValue "+switchValue+" ; current
        //            "+current+" ; "+limit);
523         //     delay(1);
        // }
    }
}
528 }

void setDACSwitches(string targetElmb, int ao, int value)
{
    int gate0 = ao-3;
533    int gate1 = ao-2;
    int gate2 = ao-1;
    int gate3 = ao-0;
    setSwitch(targetElmb+fwDevice_HIERARCHY_SEPARATOR+"AO"+fwDevice_HIERARCHY_SEPARATOR+"
        ao_"+gate0+".value", value);
    setSwitch(targetElmb+fwDevice_HIERARCHY_SEPARATOR+"AO"+fwDevice_HIERARCHY_SEPARATOR+"
        ao_"+gate1+".value", value);
538    setSwitch(targetElmb+fwDevice_HIERARCHY_SEPARATOR+"AO"+fwDevice_HIERARCHY_SEPARATOR+"
        ao_"+gate2+".value", value);
    setSwitch(targetElmb+fwDevice_HIERARCHY_SEPARATOR+"AO"+fwDevice_HIERARCHY_SEPARATOR+"
        ao_"+gate3+".value", value);
}

void setSwitch(string dp, int value)
543 {
    // This is the function that sets the dp for all ao of the DAC card
    dpSetWait(dp, value);
}

548 void readSequence(dyn_anytype &sequence, bool power)
{
    //DebugN(sequence);

    string dp;
553    time_t start,tend, tdiff;
    unsigned delayTotal, elapsedTotal;
    unsigned delaySeconds, delayMilliSeconds;

    sequence[posFlag] = true ;
558    sequence[posTimeStamp] = getCurrentTime();

    if (sequence[posAo]!= false)
    {
        // Enable Switch (the DAC channel corresponding at each sensor)
563        dp = sequence[posElmb]+"AO/ao_"+sequence[posAo];
        setSwitch(dp+".value", sequence[posEnable]); // here it takes the value proper of
            each sensor in the above table
    }

    tstart = getCurrentTime();
568    delayTotal = sequence[posDelay];
    fixDelay(delayTotal); // calculate the real delay (see below)

```

```

delaySeconds      = floor(delayTotal/1000);
delayMilliseconds = delayTotal -(delaySeconds*1000);
delay(delaySeconds, delayMilliseconds);

573 for (int limit=0; limit<maxReiterationsCurrent; limit++) // it retries several times
{
    doSequenceMeasurement(sequence); // it calls the "real" sequence (see below)
    if (power==false)
578     { break; }
    if (sequence[posCurrent]==0)
    { break; }
    if ( checkMeasurementDivergency(sequence[posCurrent], sequence[posExpectedCurrent],
        sequence[posExpectedDivergency]) )
    { break; }
583     DebugN("Measurement quality wrong; retrying "+sequence[posDevice]);
    //     DebugN(sequence[posCurrent]);
    //     DebugN(currentConversionFactor);
    //     DebugN(sequence[posCurrent]*currentConversionFactor);
    //     DebugN(sequence[posExpectedCurrent]*0.5);
588 //     DebugN(sequence[posExpectedCurrent]*1.5);
}

    tend = getCurrentTime();
    tdiff = tend-tstart;
593 elapsedTotal = second(tdiff)*1000+milliSecond(tdiff);
    sequence[posElapsed] = elapsedTotal;

    if (sequence[posAo]!= false)
    {
598         dp = sequence[posElmb]+"/AO/ao_"+sequence[posAo]
            setSwitch(dp+".value", sequence[posDisable]);
    }

    sequence[posFlag] = false ;
603 }

bool checkMeasurementDivergency(double current, double expected, double divergency)
{
    if ( (current>expected*(1-divergency)) && (current<expected*(1+divergency)) )
608     { return true; }
    else
    { return false; }
}

613 // calculate the delay to be applied considering the SDO commands bus delay
void fixDelay(unsigned &totalmili)
{
    double correction = 2*sdoDelay;
    double real = totalmili-correction;
618     if (real > 0)
    {
        totalmili = real;
    }
623     else
    {
        totalmili = 0;
    }
}

628 void doSequenceMeasurement(dyn_anytype &sequence)
{
    string dp;
    double value, current;
633     dyn_string dsExceptionInfo;

```

```

if (sequence[posAisdoValue] != false)
{
    // Read Value
638 dp=sequence[posElmb]+"/AI/aisdo_"+sequence[posAisdoValue];
    fwElmb_elementSQ(dp+".rawValue", 2, value, dsExceptionInfo);
    if (dynlen(dsExceptionInfo) > 0)
    {
        DebugN(dp);
643     DebugN(dsExceptionInfo);
        sequence[posValue] = dsExceptionInfo;
    }
    else
    {
648     dpGet(dp+".value", value);
        sequence[posValue] = value;
    }
}

653 if (sequence[posAisdoCurrent] != false)
{
    // Read Current
    dp=sequence[posElmb]+"/AI/aisdo_"+sequence[posAisdoCurrent];
    fwElmb_elementSQ(dp+".rawValue", 2, current, dsExceptionInfo);
658     if (dynlen(dsExceptionInfo) > 0)
    {
        DebugN(dp);
        DebugN(dsExceptionInfo);
        sequence[posCurrent] = dsExceptionInfo;
663     }
    else
    {
        dpGet(dp+".value", current);
        sequence[posCurrent] = current*currentConversionFactor;
668     }
}

void debugSequences(dyn_dyn_anytype sequences)
673 {
    if (debug)
    {
        DebugN("    Switch    ===> "+sequences[sequenceSwitch]);
        DebugN("    Ntc      ===> "+sequences[sequenceNtc]);
678     DebugN("    Pin1     ===> "+sequences[sequencePin1]);
        DebugN("    Pin2     ===> "+sequences[sequencePin2]);
        DebugN("    Radfet1  ===> "+sequences[sequenceRadfet1]);
        DebugN("    Radfet2  ===> "+sequences[sequenceRadfet2]);
    }
683 }

```

Part

References

(This page is intentionally left blank)

Publications

- [AAA⁺07] M. Albrow, G. Antchev, M. Arneodo, V. Avati, P. Bartalini, V. Berardi, U. Bottigli, M. Bozzo, E. Brucken, V. Burtovoy, A. Buzzo, M. Calicchio, F. Capurro, M. Catanesi, P. Catastini, M. Ciocci, R. Croft, K. Datsko, M. Deile, J. de Favereau de Jeneret, D. de Jesus Damiao, E. Robutti, A. de Roeck, D. D'Enterria, E. de Wolf, K. Eggert, R. Engel, S. Erhan, F. Ferro, W. García-Fuertes, W. Geist, M. Grothe, J. Guillaud, J. Heino, A. Hees, T. Hilden, J. Kalliopuska, J. Kaspar, P. Katsas, V. Kim, V. Klyukhin, V. Kunderát, K. Kurvinen, A. Kuznetsov, S. Lami, J. Lamsa, G. Latino, R. Lauhakangas, E. Lippmaa, J. Lippmaa, Y. Liu, A. Loginov, M. Lokajícek, M. Lo Vetere, F. Lucas Rodríguez, M. Macrí, T. Mäki, M. Meucci, S. Minutoli, J. Mnich, I. Moussienko, M. Murray, H. Niewiadomski, E. Noschis, G. Notarnicola, S. Ochesanu, K. Österberg, E. Oliveri, F. Oljemark, R. Orava, M. Oriunno, M. Ottela, S. Oryn, P. Palazzi, A. Panagiotou, R. Paoletti, V. Popov, V. Petrov, T. Pierzchala, K. Piotrkowski, E. Radermacher, E. Radicioni, G. Rella, S. Reucroft, L. Ropelewski, X. Rouby, G. Ruggiero, A. Rummel, M. Ruspa, R. Ryutin, H. Saarikko, G. Sanguinetti, A. Santoro, A. Santroni, E. Sarkisyan-Grinbaum, L. Sarycheva, F. Schilling, P. Schlein, A. Scribano, G. Sette, W. Snoeys, G. Snow, A. Sobol, A. Solano, F. Spinella, P. Squillacioti, J. Swain, A. Sznajder, M. Tasevsky, C. Taylor, F. Torp, A. Trummal, N. Turini, M. van der Donckt, P. van Mechelen, N. van Remortel, A. Vilela-Pereira, J. Whitmore, and D. Zaborov, 'Prospects for diffractive and forward physics at the LHC,' CERN/LHCC 2006-039/G-124, 2007.
- [AAA⁺08] G. Anelli, G. Antchev, P. Aspell, V. Avati, M. Bagliesi, V. Berardi, M. Berretti, V. Boccone, U. Bottigli, M. Bozzo, E. Brücken, A. Buzzo, F. Cafagna, M. Calicchio, F. Capurro, M. Catanesi, P. Catastini, R. Cecchi, S. Cerchi, R. Cereseto, M. Ciocci, S. Cuneo, C. da Vià, E. David, M. Deile, E. Dimovasili, M. Doubrava, K. Eggert, V. Eremín, F. Ferro, A. Foussat, M. Galuska, F. Garcia, F. Gherarducci, S. Giani, V. Greco, J. Hasi, F. Haug, J. Heino, T. Hilden, P. Jarron, C. Joram, J. Kalliopuska, J. Kaplon, J. Kaspar, V. Kunderát, K. Kurvinen, J. Lacroix, S. Lami, G. Latino, R. Lauhakangas, E. Lippmaa, M. Lokajícek, M. Lo Vetere, F. Lucas Rodríguez, D. Macina, M. Macrí, C. Magazzù, G. Magazzù, A. Magri, G. Maire, A. Manco, M. Meucci, S. Minutoli, A. Morelli, P. Musico, M. Negri, H. Niewiadomski, E. Noschis, G. Notarnicola, E. Oliveri, F. Oljemark, R. Orava, M. Oriunno, A. Perrot, K. Österberg, R. Paoletti, E. Pedreschi, J. Petäjäjärvi, P. Pollovio, M. Quinto, E. Radermacher, E. Radicioni, S. Rangod, F. Ravotti, G. Rella, E. Robutti, L. Ropelewski, G. Ruggiero, A. Rummel, H. Saarikko, G. Sanguinetti, A. Santroni, A. Scribano, G. Sette, W. Snoeys, F. Spinella, P. Squillacioti, A. Ster, C. Taylor, A. Tazzioli, D. Torazza, A. Trovato, A. Trummal, N. Turini, V. Vacek, N. van Remortel, V. Vins, S. Watts, J. Whitmore, and J. Wu, 'The TOTEM experiment at the CERN Large Hadron Collider,' JINST, 2008.
- [AL08] I. Atanassov and F. Lucas Rodríguez, 'Finite state machines hierarchy,' EDMS 896904 rev. 0.1, 2008.

Publications

- [ALPR08] I. Atanassov, F. Lucas Rodríguez, P. Palazzi, and F. Ravotti, ‘TOTEM DCS phasing and planning,’ EDMS 896893 rev. 0.1, 2008.
- [BBB⁺08] M. Berretti, V. Boccone, U. Bottigli, M. Bozzo, E. Brücken, A. Buzzo, F. Cafagna, M. Calicchio, F. Capurro, M. Catanesi, P. Catastini, R. Cecchi, S. Cerchi, R. Cereseto, M. Ciocci, S. Cuneo, C. da Vià, E. David, M. Deile, E. Dimovasili, M. Doubrava, K. Eggert, V. Eremin, F. Ferro, A. Foussat, M. Galuška, F. Garcia, F. Gherarducci, S. Giani, V. Greco, J. Hasi, F. Haug, J. Heino, T. Hilden, P. Jarron, C. Joram, J. Kalliopuska, J. Kaplon, J. Kaspar, V. Kunderát, K. Kurvinen, J. Lacroix, S. Lami, G. Latino, R. Lauhakangas, E. Lippmaa, M. Lokajícek, M. Lo Vetere, F. Lucas Rodríguez, D. Macina, M. Macrí, C. Magazzù, G. Magazzù, A. Magri, G. Maire, A. Manco, M. Meucci, S. Minutoli, A. Morelli, P. Musico, M. Negri, H. Niewiadomski, E. Noschis, G. Notarnicola, E. Oliveri, F. Oljemark, R. Orava, M. Oriunno, A. Perrot, K. Österberg, R. Paoletti, E. Pedreschi, J. Petäjäjärvi, P. Pollovio, M. Quinto, E. Radermacher, E. Radicioni, S. Rangod, F. Ravotti, G. Rella, E. Robutti, L. Ropelewski, G. Ruggiero, A. Rummel, H. Saarikko, G. Sanguinetti, A. Santroni, A. Scribano, G. Sette, W. Snoeys, F. Spinella, P. Squillacioti, A. Ster, C. Taylor, A. Tazzioli, D. Torazza, A. Trovato, A. Trummal, N. Turini, V. Vacek, N. V. Remortel, V. Vins, S. Watts, J. Whitmore, and J. Wu, ‘Diffraction at TOTEM,’ HERA LHC workshop, 2008.
- [DJL⁺08] M. Deile, M. Jonker, F. Lucas Rodríguez, G. Maire, and E. Radermacher, ‘Movement control of the TOTEM roman pots,’ EDMS 873014 rev. 0.6, 2008.
- [DL08] E. Dimovasili and F. Lucas Rodríguez, ‘Pinout tables for the TOTEM roman pots,’ EDMS 945519 rev. 0.2, 2008.
- [DLR08] E. Dimovasili, F. Lucas Rodríguez, and F. Ravotti, ‘TOTEM on-line radiation monitoring system,’ EDMS 873014 rev. 0.6, 2008.
- [Gro08] I. P. W. Group, *IEEE P1394 - Standard for a High Performance Serial Bus*. IEEE Standards Association, 2008.
- [GTR⁺08] S. Giani, M. Tuhkanen, E. Radermacher, F. Lucas Rodríguez, G. Ruggiero, F. Ravotti, E. R. M. Deile, V. Avati, E. Dimovasili, W. Snoeys, and S. Sadilov, ‘TOTEM data architectural design,’ <http://www.cern.ch/test-DMTotem/>, February 2008.
- [LM08] F. Lucas Rodríguez and J. Morant, ‘TOTEM DCS ELMB rack proposal,’ EDMS 982044 rev 2.0, December 2008.
- [Luc07] F. Lucas Rodríguez, ‘TOTEM DCS document catalog,’ EDMS 896895 rev. 0.1, 2007.
- [Luc08a] F. Lucas Rodríguez, ‘Computing resources,’ EDMS 896907 rev. 0.1, 2008.
- [Luc08b] F. Lucas Rodríguez, ‘Operational model of the FESA gateway for the TOTEM roman pots,’ EDMS 889422 rev. 0.2, 2008.
- [Luc08c] F. Lucas Rodríguez, ‘TOTEM DCS hardware overview diagrams,’ EDMS 868711 rev. 0.7, 2008.
- [Luc08d] F. Lucas Rodríguez, ‘TOTEM DCS sensors summary table,’ EDMS 896905 rev. 0.1, March 2008.
- [Luc08e] F. Lucas Rodríguez, ‘TOTEM DCS summary tables,’ EDMS 896903 rev. 0.1, 2008.
- [Luc08f] F. Lucas Rodríguez, ‘TOTEM general: Functional and technical requirements,’ EDMS 896896 rev. 0.1, 2008.

- [ORL06] M. Oriunno, E. Radermacher, and F. Lucas Rodríguez, ‘Operation of the TOTEM roman pots,’ EDMS 863466 rev. 1.0, 2006.
- [PLDR08] M. Philippe Dutour, F. Lucas Rodríguez, M. Deile, and E. Radermacher, ‘TOTEM roman pots control system use cases specification,’ EDMS 896907 rev. 0.1, 2008.

(This page is intentionally left blank)

Bibliography

- [AAB⁺03] R. Assmann, O. Aberle, I. S. Baishev, L. Bruno, M. Brügger, E. Chiaveri, B. Dehning, A. Ferrari, B. Goddard, J. J. M. Jiménez, V. Kain, D. Kaltchev, M. Lamont, F. Ruggiero, R. Schmidt, P. Sievers, J. Uythoven, V. Vlachoudis, L. Vos, and J. Wenninger, ‘Designing and building a collimation system for the high-intensity LHC beam,’ LHC-PROJECT-REPORT-640, June 2003.
- [AAB⁺07] P. Aspell, G. Antchev, D. Barney, S. Reynaud, W. Snoeys, and P. Vichoudis, ‘VFAT2: A front-end system on chip providing fast trigger information, digitized data storage and formatting for the charge sensitive readout of multi-channel silicon and gas particle detectors,’ in *Proceedings of TWEPP-07, Topical Workshop on Electronics for Particle Physics*, Prague, Czech Republic, September 2007.
- [AAG⁺05] O. Aberle, R. Assmann, B. Goddard, V. Kain, M. Jonker, M. Lamont, R. Losito, A. Masi, R. Schmidt, C.-H. Sicard, and M. Sobczak, ‘The controls architecture for the LHC collimation system,’ in *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems 2005*, Geneva, Switzerland, October 2005.
- [ABB⁺02a] R. Assmann, I. Baishev, M. Brugger, L. Bruno, H. Burkhardt, G. Burtin, B. Dehning, C. Fischer, B. Goddard, E. Gschwendtner, M. Hayes, J. Jeanneret, R. Jung, V. Kain, D. Kaltchev, M. Lamont, R. Schmidt, E. Vossenberg, E. Weisse, and J. Wenninger, ‘Requirements for the LHC collimation system,’ LHC-PROJECT-REPORT-599, 2002.
- [ABB⁺02b] A. Augustinus, G. Benincasa, H. Burckhart, B. Flockhart, P. Gavillet, R. Nunes, S. Philippin, J. Pothier, W. Salter, E. Sbrissa, C. Schäfer, S. Schmeling, L. Scibile, and W. Tejessy, ‘A detector safety system for the LHC experiments: Functional requirements document,’ CERN-JCOP-2002-012, 2002.
- [ABd⁺04] M. Alfonsia, G. Bencivennia, P. de Simone, F. Murtasa, M. P. Lenera, W. Bonivento, A. Cardinib, C. Deplanob, D. Pincib, D. Rasinob, and B. Saittab, ‘High-rate particle triggering with triple-gem detector,’ *Nuclear Instruments and Methods in Physics Research Section A*, vol. 518, pp. 106--112, 2004.
- [AGVW02] R. Assmann, B. Goddard, E. Vossenberg, and E. Weisse, ‘The consequences of abnormal beam dump actions on the LHC collimation system,’ LHC-PROJECT-NOTE-293, 2002.
- [ALI08] ALICE Collaboration, ‘ALICE website,’ <http://www.cern.ch/alice/>, 2008.
- [Ass09] B. Association, ‘Building Automation and Control Network,’ <http://www.bacnetassociation.org>, February 2009.

Bibliography

- [ASZZ02] R. Assmann, F. Schmidt, F. Zimmermann, and M. Zorzano, ‘Equilibrium beam distribution and halo in the Lhc,’ in *Proceedings of the European Particle Accelerator Conference 2002*, Paris, France, 2002.
- [ATL97] ATLAS Collaboration, ‘ATLAS muon spectrometer: Technical Design Report, chapter 6,’ CERN/LHCC/97-22, May 1997.
- [ATL08] ATLAS Collaboration, ‘ATLAS website,’ <http://www.cern.ch/atlas/>, 2008.
- [Aug06] A. Augustinus, ‘ALICE DCS user requirements document,’ <http://alicedcs.web.cern.ch/AliceDCS/URD/>, July 2006.
- [Bai07] S. Baird, ‘Accelerators for pedestrians,’ CERN-AB-NOTE-2007-014, February 2007.
- [Bau08] C. Bault, ‘Procédure d’installation des pots romains,’ EDMS 901060 rev 2.0, 2008.
- [BBB⁺04] D. Beck, K. Blaum, H. Brand, Herfurth, and S. Schwarz, ‘A new control system for ISOLTRAP,’ *Nuclear Instruments and Methods in Physics Research Section A*, pp. 567--579, July 2004.
- [BCC⁺03] V. Baggiolini, F. Calderini, F. Chevrier, S. Jensen, K. Kostro, N. Trofimov, and J. Andersson, ‘Controls middleware project,’ <http://www.cern.ch/proj-cmw/>, 2003.
- [BCR⁺04] V. Berardi, M. G. Catanesi, E. Radicioni, R. Herzog, R. Rudischer, E. Wobst, M. Deile, K. Eggert, F. Haug, P. Jarron, D. Macina, H. Niewiadomski, E. Noschis, M. Oriunno, A. Perrot, G. Ruggiero, W. Snoeys, A. Verdier, V. Boccone, M. Bozzo, A. Buzzo, F. Capurro, S. Cuneo, F. Ferro, M. Macri, S. Minutoli, A. Morelli, P. Musico, M. Negri, A. Santroni, G. Sette, A. Sobol, V. Avati, E. Goussev, M. Järvinen, J. Kalliopuska, K. Kurvinen, R. Lauhakangas, F. Oljemark, R. Orava, K. Österberg, V. Palmieri, H. Saarikko, A. Toppinen, V. Kundrať, M. Lokajíček, C. Da Vià, J. Hasi, A. Kok, and S. Watts, ‘TOTEM: Technical Design Report,’ CERN-LHCC-2004-002, January 2004.
- [BH07] R. Barillère and S. Haider, ‘LHC gas control systems: A common approach for the control of the Lhc experiments gas systems,’ CERN-JCOP-2002-14, September 2007.
- [BRJ05] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language: user guide*. Addison-Wesley, May 2005.
- [BS04] M. Beharrell and W. Salter, ‘LHC data interchange protocol (DIP),’ EDMS 457113 rev 2.0, June 2004.
- [BSW02] C. Bourelly, J. Soffer, and T. Wu, ‘Impact-picture phenomenology for $p\pm p$, $k\pm p$ and $pp, p\bar{p}$ elastic scattering at high energies,’ *The European Physical Journal C*, vol. 28, pp. 97--105, 2002.
- [Bur08] P. Burkimsher, ‘Joint PVSS and JCOP framework course,’ <http://itcobe.web.cern.ch/itcobe/Services/Pvss/Training/welcome.html>, January 2008.
- [CAE05] CAEN S.P.A., ‘1520P technical information manual,’ April 2005.
- [Car05] E. Carrone, ‘Design, construction and commissioning of the thermal screen control system for the cms tracker detector at CERN,’ Ph.D. dissertation, Università degli studi di Bari, January 2005.

- [CEG⁺02] J. Cudell, V. Ezhela, P. Gauron, K. Kang, Y. Kuyanov, S. B. Lugovsky, E. Martynov, B. Nicolescu, E. Razuvaev, and N. Tkachenko, 'Benchmarks for the forward observables at RHIC, the tevatron-run II, and the LHC,' *Physical Review Letters*, pp. 201 801--201 805, 2002.
- [CER06] CERN TS-CE, 'LHC civil engineering,' <http://www.cern.ch/ts-dep/groups/ce/ce.htm>, 2006.
- [CER08a] CERN, 'A global endeavour,' <http://www.cern.ch/public/en/About/Global-en.html>, 2008.
- [CER08b] CERN, 'History highlights,' <http://www.cern.ch/public/en/About/History-en.html>, 2008.
- [CER08c] CERN, 'LHC: the guide,' CERN-BROCHURE-2008-001, January 2008.
- [CER08d] CERN Press Release, 'Lhc to restart in 2009,' <http://press.web.cern.ch/press/PressReleases/Releases2008/PR17.08E.html>, December 2008.
- [CLvL05] M. Case, M. Liendla, and F. van Lingenb, 'XML based detector description language,' CMS-NOTE-2005/000, April 2005.
- [CMS97] CMS Collaboration, 'CMS: The muon project,' CERN-LHCC 97-32, 1997.
- [CMS02] CMS Collaboration, 'The TriDAS project: Technical Design Report volume 2: Data acquisition and high-level trigger,' CERN-LHCC-2002-026, December 2002.
- [CMS05] CMS Trigger/DAQ Group, 'Integration of run control and detector control systems,' CMS-IN-2005-015, 2005.
- [CMS08] CMS Collaboration, 'CMS website,' <http://cms.cern.ch>, 2008.
- [CT05] J. Cook and G. Thomas, 'ELMB128 documentation: Everything you wanted to know about the ELMB128 but were afraid to ask,' EDMS 684947 rev 4.3, February 2005.
- [CW07] G. Corti and D. Wiedner, 'LHCb radiation monitors for detectors and on detector electronics,' EDMS 860046, 2007.
- [DBBL07] R. Davis, A. Burns, R. Bril, and J. Lukkien, 'Controller area network (CAN) schedulability analysis: Refuted, revisited and revised,' *Real-Time Systems*, pp. 239--272, April 2007.
- [Dei08] M. Deile, 'TOTEM beam interlocks logic,' Private communication, August 2008.
- [Din08] J. A. Dinis Neves, 'Summer student report: Quality testing of ELMB-DAC production,' September 2008.
- [Dou04] B. Douglas, *Real Time UML: Advances in the UML for the real-time systems*. Addison-Wesley, February 2004.
- [DTB97] K. Dutton, S. Thompson, and B. Barraclough, *The art of control engineering*. Prentice Hall, 1997.
- [Dut08] M. Dutour, 'Interface control document,' EDMS 970055 rev 1.0, September 2008.
- [ECS04a] ECSS, 'The European Cooperation for Space Standardization,' <http://www.ecss.nl>, 2004.
- [ECS04b] ECSS, 'Space engineering: Control engineering,' ECSS-E-60A, September 2004.

Bibliography

- [ETM08] ETM A.G., ‘Prozessvisualisierungs und steuerungssystem (PVSS),’ <http://www.etm.at>, 2008.
- [Gas04] C. Gaspar, ‘JCOP framework: Hierarchical controls,’ CERN-JCOP-2004-001, February 2004.
- [GD97] G. Gruhler and B. Dreier, ‘CANopen implementation guidelines,’ <http://www.can-cia.org>, 1997.
- [GDC01] C. Gaspar, M. Donszelmann, and P. Charpentier, ‘Dim, a portable, light weight package for information publishing, data transfer and inter-process communication,’ *Computer Physics Communications*, pp. 102--109, 2001.
- [GGV06] F. Glege, R. Gómez-Reino Garrido, and J. Varela, ‘CMS DCS integration guidelines rev. 3.0,’ http://cmsdoc.cern.ch/cms/TRIDAS/DCS/central_dcs/guidelines, October 2006.
- [Gle08] F. Glege, ‘CMS racks control,’ Private communication, January 2008.
- [Gra09] M. Graphic, ‘Volcano network architect,’ <http://www.mentor.com>, January 2009.
- [Hel08a] Helmholtz Centre for Heavy Ion Research, ‘Control System framework,’ <http://sourceforge.net/projects/cs-framework/>, 2008.
- [Hel08b] Helmholtz Centre for Heavy Ion Research, ‘GSI at a glance,’ http://www.gsi.de/portrait/ueberblick_e.html, 2008.
- [HSRG07] A. Holmes-Siedle, F. Ravotti, and M. Glaser, ‘The dosimetric performance of RADFETs in radiation test beams,’ in *IEEE Radiation Effects Data Workshop 23-27*, Honolulu, Hawaii, July 2007.
- [Huh06] M. Huhtinen, ‘TOTEM collaboration meetings 31/1/2006 (T2) and 7/3/2006 (T1),’ <http://indico.cern.ch/categoryDisplay.py?categId=31183>, 2006.
- [ILP03] M. Islam, R. Luddy, and A. Produkin, ‘Elastic scattering at LHC and nucleon structure,’ *Modern Physics Letters A*, vol. 18, pp. 743--752, 2003.
- [IS85] G. Ingelman and P. Schlein, ‘Jet structure in high mass diffractive scattering,’ *Physics Letters B*, vol. 152, pp. 256--260, 1985.
- [JCO07] JCOP Framework Team, ‘Joint controls project (JCOP) framework subproject: guidelines and conventions,’ CERN-JCOP-2000-008, July 2007.
- [JLOT96] J. Jeanneret, D. Leroy, L. Oberli, and T. Trenkler, ‘Quench levels and transient beam losses in lhc magnets,’ LHC-PROJECT-REPORT-44, 1996.
- [Joh03] C. Johnson, *Process control instrumentation technology*. Prentice Hall, 2003.
- [KAd⁺03] K. Kostro, J. Andersson, F. di Maio, S. Jensen, and N. Trofimov, ‘The controls middleware (CMW) at CERN: status and usage,’ in *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems 2003*, Gyeongju, Korea, October 2003.
- [KBB⁺97] W. Kienzle, M. Bozzo, M. Buénerd, Y. Muraki, J. Bourotte, M. Haguénauer, G. Sanguinetti, G. Matthiae, A. Faus-Golfe, and J. Velasco, ‘The TOTEM collaboration: Letter of Intent,’ CERN-LHCC-97-049, August 1997.

- [KCD⁺06] G. Kramberger, V. Cindro, I. Dolenc, I. Mandic, and M. Mikuz, ‘Design and functional specification of ATLAS radiation monitor,’ <http://www-f9.ijs.si/~mandic/RADMON/docs/RADMON-Concept-V2.0.pdf>, 2006.
- [KOP⁺99] W. Kienzle, M. Oriunno, A. Perrot, S. Weisz, M. Bozzo, A. Buzzo, M. Macri, A. Santroni, G. Settle, M. Buénerd, F. Malek, Y. Muraki, K. Kasahara, G. Sanguinetti, G. Matthiae, P. Privitera, V. Verzi, A. Faus-Golfe, J. Velasco, and S. Torii, ‘The TOTEM collaboration: Technical Proposal,’ CERN-LHCC-99-007, March 1999.
- [Kou02] J.-P. Koutchouk, ‘Measurement of the beam position in the LHC main rings,’ EDMS 327557 rev. 2.0, 2002.
- [KR05] J. Knobloch and L. Robertson, ‘LHC computing grid: Technical Design Report,’ LHCC-2005-024, June 2005.
- [KRS⁺05] V. Kain, J. Ramillon, R. Schmidt, K. Vorderwinkler, and J. Wenninger, ‘Material damage test with 450 GeV LHC-type beam,’ Knoxville, Tennessee, May 2005.
- [KTH⁺02] J. Kuijt, P. Timmer, B. Hallgren, P. de Groen, D. Tascon Lopez, S. Schouten, and H. Boterenbrood, ‘ELMB-DAC user manual rev. 1.0,’ <http://www.nikhef.nl/pub/departments/ct/po/html/ELMB/DAC10.pdf>, May 2002.
- [Lau06] R. Lauckner, ‘LHC operational mode,’ LHC-OP-ES-0004, March 2006.
- [LFMS03] S. Lüders, B. Flockhart, G. Morpurgo, and S. Schmeling, ‘The CERN detector safety system for the LHC experiments,’ in *Computing in High Energy and Nuclear Physics*, La Jolla, California, March 2003.
- [LHC00] LHC Study Group, ‘LHC Design Report volume 1,’ EDMS 115442 rev 1.0, November 2000.
- [LHC01] LHCb Collaboration, ‘LHCb Technical Design Report,’ CERN-LHCC-2001-010, 2001.
- [LHC08] LHCb Collaboration, ‘LHCb website,’ <http://www.cern.ch/lhcb/>, 2008.
- [LM01] R. Ley and D. Manglunki, ‘CERN accelerators,’ <http://www.cern.ch/public/en/About/History-en.html>, 2001.
- [Lop06] N. Lopez, ‘Printed circuit board layout design files for hute_mon04,’ EDMS 760347 rev. 2.0, 2006.
- [LW06a] S. Land and J. Walz, *Practical support for CMMI-SW software project documentation*. Wiley-Interscience, 2006.
- [LW06b] S. Land and J. Walz, *Practical support for ISO 9001 software project documentation*. Wiley-Interscience, 2006.
- [Man08] I. Mandic, ‘Radmon readout prototype for atlas experiment,’ Private communication, JSI Ljubljana, October 2008.
- [MCD⁺06] N. Mokhov, P. Czarapata, A. Drozhdin, D. Still, and R. Samulyak, ‘Beam-induced damage to the TEVATRON components and what has been done about it,’ in *Proceedings of HB2006*, Tsukuba, Japan, May 2006.
- [MCG⁺07] I. Mandic, V. Cindro, A. Gorisek, G. Kramberger, and M. Mikuz, ‘Integrating radiation monitoring system for the ATLAS detector at the LHC,’ *IEEE Transactions on Nuclear Science*, pp. 1143--1150, 2007.

Bibliography

- [MLD09] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC Unified Architecture*. Springer Verlag, 2009.
- [MMM04] G. Magazzu, A. Marchioro, and P. Moreira, ‘The detector control unit: an ASIC for the monitoring of the CMS silicon tracker,’ *IEEE Transaction on Nuclear Science*, pp. 1333--1336, 2004.
- [MRKS03] N. Mokhov, I. Rakhno, J. Kerby, and J. Strait, ‘Protecting LHC IP1/IP5 components against radiation resulting from colliding beam interactions,’ LHC-PROJECT-REPORT-633, March 2003.
- [MSW06] D. Macina, W. H. Smith, and J. Wenninger, ‘LHC experiments beam interlocking,’ EDMS 653932 rev 1.0, June 2006.
- [Mye99] D. Myers, ‘The LHC experiments joint controls project: JCOP,’ in *Proceedings of International Conference on Accelerator and Large Experimental Physics Control Systems 1999*, Trieste, Italy, October 1999.
- [Nie08] H. Niewiadomski, ‘Reconstruction of protons in the TOTEM roman pots detectors at the LHC,’ Ph.D. dissertation, University of Manchester, September 2008.
- [Nos06] E. Noschis, ‘Planar edgeless detectors for the TOTEM experiment at the LHC,’ Ph.D. dissertation, University of Helsinki, June 2006.
- [OPC96] OPC Foundation, ‘OLE for process control,’ <http://www.opcfoundation.org>, August 1996.
- [Org03a] Organisation for Economic Co-operation and Development, *Frascati Manual*. OECD Publishing, 2003.
- [Org03b] I. S. Organization, ‘Iso 16484,’ October 2003.
- [Pal06] P. Palazzi, ‘TOTEM DCS project management plan,’ EDMS 828889 rev. 0.3, May 2006.
- [Par96] D. Paret, *Le bus CAN*. DUNOD, 1996.
- [PS06] B. Puccio and R. Schmidt, ‘The beam interlock system for the LHC,’ EDMS 567256 rev 0.2, June 2006.
- [PXI98] PXI Systems Alliance, ‘PXI website,’ <http://www.pxisa.org>, 1998.
- [Rav06] F. Ravotti, ‘Development and characterization of radiation monitoring sensors for the high energy physics experiments of the CERN LHC accelerator,’ Ph.D. dissertation, Université Montpellier II, November 2006.
- [RGM05] F. Ravotti, M. Glaser, and M. Moll, ‘Sensor catalogue: Data compilation of solid-state sensors for radiation monitoring,’ EDMS 590497 rev 1.0, May 2005.
- [RGR⁺07] F. Ravotti, M. Glaser, A. B. Rosenfeld, M. Lerch, A. Holmes-Siedle, and G. Sarabayrouse, ‘Radiation monitoring in mixed environments at CERN: from the IRRAD6 facility to the LHC experiments,’ *IEEE Transactions on Nuclear Science*, pp. 1170--1177, 2007.
- [RM08] S. Redaelli and A. Masi, ‘Application software for the LHC collimators and movable elements,’ LHC-TCT-ES-0001 rev 0.2, November 2008.
- [Rug03] G. Ruggiero, ‘Signal generation in highly irradiated silicon microstrip detectors for the ATLAS experiment,’ Ph.D. dissertation, University of Glasgow, May 2003.

- [Rug08] G. Ruggiero, 'Product breakdown structure and naming scheme of the roman pot system,' EDMS 906715 rev. 0.2, 2008.
- [RZ06] F. Ruggiero and F. Zimmermann, 'Possible scenarios for an LHC upgrade,' in *Proceedings of LHC-LUMI-05*, CARE-HHH-APD LHC-LUMI-05 workshop, August 2006.
- [Sch05] S. Schmeling, 'Common tools for large experiment controls: A common approach for deployment, maintenance and support,' in *Proceedings of the 14th IEEE-NPSS Real Time Conference*, Stockholm, Sweden, 2005.
- [SCP01] SCPI Consortium, 'SCPI website,' <http://www.scpiconsortium.org>, 2001.
- [SIE07] SIEMENS, 'SITOP power flexi 6ep13532ba00,' 2007.
- [Sil06] Silicon Microstructures Inc., 'Surface mount and dip pressure sensors low-cost packaged die sm5430/sm5470,' 2006.
- [Sof02] Software Engineering Institute, 'Capability Maturity Model Integrated,' <http://www.sei.cmu.edu/cmmi/>, 2002.
- [Sor08] Sorina Popescu, 'Magnet field measurements in the forward region,' CMS week December, 2008.
- [Str03] J. Strait, 'Towards a new LHC interaction region design for a luminosity upgrade,' in *Proceedings of PAC2003*, Portland, Oregon, May 2003.
- [STVW05] M. Schwerin, A. Tsiros, P. Verdini, and M. Weber, 'The humidity sensors for the CMS tracker,' CMS-NOTE-2005/000, 2005.
- [SYS06] SYS TEC electronic GmbH, 'USB-CAN interface,' <http://www.systec-electronic.com>, 2006.
- [SYS07] SYS TEC electronic GmbH, 'Systems manual for USB-CAN modules GW-001, GW-002, 3004006, 320400x, 340400x,' <http://www.systec-electronic.com>, June 2007.
- [Szy02] C. Szyperski, *Component Software: Beyond Object-Oriented Programming*. 2nd ed. Addison-Wesley Professional, 2002.
- [TL06a] TS-LEA, 'Demande d'installation de câbles RP,' 2006.
- [TL06b] TS-LEA, 'Demande d'installation de câbles T1 et T2,' 2006.
- [TOT08] TOTEM Collaboration, 'TOTEM website,' <http://www.cern.ch/totem/>, 2008.
- [Tse05] E. Tsesmelis, 'Experiment-machine interface issues and signal exchange,' in *Proceedings of the LHC project workshop XIV; CERN-AB-2005-014*, Chamonix, France, January 2005.
- [Tse06] E. Tsesmelis, 'Data and signals to be exchanged between the LHC machine and experiments,' EDMS 701510 rev 4.0, February 2006.
- [Var02] F. Varela Rodríguez, 'The detector control system of the ATLAS experiment at CERN: An application to the calibration of the modules of the tile hadron calorimeter,' Ph.D. dissertation, Universidad de Santiago de Compostela, March 2002.
- [VVG08] V. Vacek, V. Vins, M. Galuska, and M. Doubrava, 'Calibration of 4 low range pressure sensors for TOTEM roman pots,' Private communication, 2008.

Bibliography

- [W-106] W-Ie-Ne-R GmbH, ‘Maraton power supply system: Technical manual,’ September 2006.
- [Zel07] S. Zelepukin, ‘CMS ECAL power supplies for the ELMB,’ Private communication, CERN PH/UCM, September 2007.

Glossary

AD :	Antiproton Decelerator; a CERN accelerator
ADC :	Analog/Digital Converter
ALICE :	A Large Ion Collider Experiment; a LHC experiment
API :	Application Programming Interface
ASN.1 :	Abstract Syntax Notation One
ATLAS :	A Toroidal LHC ApparatuS; a LHC experiment
aTTS :	asynchronous Trigger Throttling System
BIS :	Beam Interlock System
BLM :	Beam Loss Monitor
BNF :	Backus-Naur Form
BPM :	Beam Position Monitor
CAN :	Controller Area Network
CANopen :	High level communication protocol for CAN
CASTOR :	Centauro And STRange Object Research in nucleus nucleus collisions; a dectector of CMS
CASTOR :	CERN Advanced STORage manager
CBS :	Cost Breakdown Structure
CCA :	Collimators Control Application
CCC :	CERN Control Centre
CERN :	European Organization for Nuclear Research
CiA :	CAN in Automation
CIBF :	Controls Interlocks Beam Fibre-link
CIBU :	Control Interlock Beam User Interfaces
CM :	Configuration Management
CMMI :	Capability Maturity Model Integration
CMP :	Configuration Management Plan
CMS :	Compact Muon Solenoid; a LHC experiment
CMW :	Controls MiddleWare; an AB technology
CNES :	Centre National d'Etudes Spatiales
CNGS :	Cern Neutrinos to Gran Sasso
COM :	Component Object Model
CSAM :	CERN Safety Alarm Monitoring System
CSC :	Cathode Strip Chambers
CSE :	CERN Safety Equipment
CSS :	CERN Safety System
CSS :	Collimators Supervisory System
CTR :	Current Terminating Ring
CTS :	Current Terminating Structure
CU :	FSM Control Unit

Glossary

DAC :	Digital/Analog Converter
DAQ :	Data AcQuisition system
DCOM :	Distributed COM
DCS :	Detector Control System
DCUF :	Detector Control Unit version Final
DDL :	Detector Description Language
DESY :	Deutsches Elektronen-Synchrotron
DIM :	Distributed Information Management system
DIP :	Data Interchange Protocol
DOM :	Document Object Model
DSP :	CANopen Device Profile Specifications
DSS :	Detector Safety System
DSU :	Detector Safety Unit
DU :	FSM Device Unit
ECS :	Experiment Control System
ECSS :	European Collaboration on Space Sciences
EDS :	Electronic Data Sheet
EIA :	Electronic Industries Alliance
ELMB :	Embedded Local Monitor Board
ELMB :	Embedded Local Monitoring Board
ELMB-DAC :	ELMB Digital-to-Analog Converter
ELSD :	EdgeLess Silicon Detector
ESA :	European Space Agency
EVB :	EVEnt Builder
FBS :	Function Breakdown Structure
FEE :	Front End Electronics
FESA :	Font-End Software Architecture; an LHC technology
FSM :	Finite State Machine
GÉANT :	A multi-gigabit pan-European data communications network for research and education use
GCS :	Gas Control System
GEM :	Gas Electron Multipliers
GMT :	General Machine Timing
GPIB :	General Purpose Interface Bus
GSI :	Helmholtzzentrum für Schwerionenforschung –Helmholtz Centre for Heavy Ion Research
HEP :	High Energy Physics
HERA :	An accelerator located at DESY in Hamburg
HF :	Hadronic Forward
HMI :	Human Machine Interface
HV :	High Voltage
I ² C:	Inter-Integrated Circuit
ICD :	Interface Control Document
IEEE :	Institute of Electrical and Electronics Engineers
IETF :	Internet Engineering Task Force
IMPI :	Intelligent Platform Management Interface

IMS :	Information and Monitor Service
INCOSE :	International Council on Systems Engineering
IP :	Interaction Point
IR :	Interaction Region
ISAG :	Industrial SCADA Application Group
ISC :	Integrated Sensor Carrier
ISO :	International Organization for Standardization
ISOLDE :	Isotope Separator OnLine DEvice; a CERN accelerator
ISR :	Intersecting Storage Rings
ITCO :	Information Technology for Controls; a CERN group
JCOP :	Joints COntrols Project; a CERN project
LCG :	LHC Computing Grid
LEADE :	LHC Experiment Accelerator Data Exchange Working Group
LEAF :	LHC Experimental Areas Forum
LECC :	LHC Electronics Coordinating Committee
LEIR :	Low Energy Ion Ring; a CERN accelerator
LEMIC :	LHC Experiment/Machine Interface Committee
LEP :	Large Electron-Positron collider
LHC :	Large Hadron Collider; the new CERN accelerator
LHCb :	Large Hadron Collider beauty; a LHC experiment
LHCC :	LHC experiments Committee
LHCf :	Large Hadron Collider forward; a LHC experiment
LINAC :	LINear ACcelerator; a CERN accelerator
LSA :	LHC Software Architecture
LU :	FSM Logic Unit
LV :	Low Voltage
LVDT :	Linear Variable Differential Transformer; a position sensor
LXI :	LAN extensions for Instrumentation
MIBs :	Management Information Basis
NMT :	CANopen Network Management
NREN :	National REsearch Network
OBS :	Organization Breakdown Structure
OD :	CANopen Object Dictionary
OID :	SNMP Object IDentifiers
OLE :	Object Linking and Embedding
OPC :	OLE for Process Control
OPC-UA :	OPC Unified Architecture
OSI :	Open System Interconnection
PBS :	Product Breakdown Structure
PCB :	Printed Circuit Board
PCI :	Peripheral Component Interconnection
PDO :	CANopen Process Data Objects
PLC :	Programmable Logic Controller
PM :	Project management
PMP :	Project Management Plan
PP :	Patch Pannel

Glossary

PS :	Proton Synchrotron; a CERN accelerator
PSB :	Proton Synchrotron Booster; a CERN accelerator
PSO :	People, System, Organization
PVSS :	ProzessVisualisierungs und SteuerungsSystem –Process visualization and control system
PXI :	PCI eXtensions for Instrumentation
PXISA :	PXI Systems Alliance
QA :	Quality Assurance
QCD :	Quantum ChromoDynamics
QD :	Defocusing Quadrupole
QF :	Focusing Quadrupole
RAC :	Real Application Clusters
RADMON :	RADiation MONitors
RBS :	Risk Breakdown Structure
RCM :	Wiener Maraton Remote Controller Module
RCMS :	Run Control and Monitor System
RF :	RadioFrequency
RFC :	IETF Request for Comments
RP :	Roman Pots; a TOTEM experiment detector
RP-PP :	Roman Pot Patch Panel
RPCS :	Roman Pots Control System; motorization
RpMe :	Roman Pots Mechanics; part of a TOTEM experiment detector
RpSi :	Roman Pots Silicon Detector; part of a TOTEM experiment detector
SCADA :	Supervisory Control And Data Acquisition
SCMP :	Software Configuration Management Plan
SDK :	Software Development Kit
SDO :	CANopen Service Data Objects
SEI :	Software Engineering Institute
SLIMOS :	Shift Leader In Matters Of Safety
SLP :	Safe LHC Parameter
SMI :	State Management Interface
SML :	State Manager Language
SNMP :	Simple Network Management Protocol
SOAP :	Simple Object Access Protocol
SPS :	Super Proton Synchrotron; a CERN accelerator
SRD :	Software Requirements Document
SUSY :	SUper SYmmetry
SVN :	SubVersioN
T1 :	Telescope 1; a TOTEM experiment detector
T2 :	Telescope 2; a TOTEM experiment detector
TEVATRON :	Tevatron is a circular particle accelerator at the Fermi National Accelerator Laboratory
TN :	Technical Network
TOTEM :	TOTAL cross section; elastic scattering and diffraction dissociation ExperiMent; a LHC experiment
UI :	User Interface
UML :	Unified Modeling Language
URD :	User Requirements Document

USB :	Universal Serial Bus
VFAT :	Very Forward ATLAS TOTEM chip
VME :	Versa Module Europa
W3C :	World Wide Web Consortium
WBS :	Work Breakdown Structure
WG :	Working Group
XDAQ :	Cross-platform DAQ framework
XML :	eXtensible Markup Language

(This page is intentionally left blank)