CMS CR -2011/301

The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

14 November 2011 (v2, 16 November 2011)

Front-End Control and Monitoring System for the Resistive Plate Chambers at the CMS Experiment

Filip Thyssen for the CMS Collaboration

Abstract

The Compact Muon Solenoid (CMS) experiment at the CERN Large Hadron Collider (LHC) uses Resistive Plate Chambers (RPCs) as a trigger-dedicated muon detector. The RPC subdetector presently consists of 912 chambers, providing a ≈ 3 ns time resolution for the pseudorapidity region $|\eta| < 1.6$. Up to 18 on-detector Front-End Boards (FEBs) per RPC are responsible for signal processing. Each FEB is configured and monitored through its Inter-Integrated Circuit (I^2C) interface to ensure correct and safe detector operation. The resulting $\sim$1.2 k I^2C lines, serving 3 to 6 FEBs apiece, are accessible through Control Boards (CBs) in token rings.

An online control and monitoring system has recently been developed to automatically configure and continuously monitor FEB parameters. A fine-grained software model of hardware and detector components, organized in a dual hierarchical tree structure, facilitates command and data flow. A novel web-based user-interface allows control of and navigation through the system for efficient problem tracing.

This paper covers the front-end control and monitoring system for the RPC subdetector; the online software components that allow for flexible configuration and comprehensive monitoring are described.

Presented at *IEEE-nss-mic-rtsd2011: IEEE2011: Nuclear Science Symposium and Medical Imaging Conference*

Front-End Control and Monitoring System for the Resistive Plate Chambers at the CMS Experiment

F. Thyssen on behalf of the CMS Collaboration

***Abstract*—The Compact Muon Solenoid (CMS) experiment at the CERN Large Hadron Collider (LHC) uses Resistive Plate Chambers (RPCs) as a trigger-dedicated muon detector. The RPC subdetector presently consists of 912 chambers, providing a $\approx$ 3 ns time resolution for the pseudorapidity region $|\eta| < 1.6$.**

Up to 18 on-detector Front-End Boards (FEBs) per RPC are responsible for signal processing. Each FEB is configured and monitored through its Inter-Integrated Circuit (I^2C) interface to ensure correct and safe detector operation. The resulting $\sim$1.2 k I^2C lines, serving 3 to 6 FEBs apiece, are accessible through Control Boards (CBs) in token rings.

An online control and monitoring system has recently been developed to automatically configure and continuously monitor FEB parameters. A fine-grained software model of hardware and detector components, organized in a dual hierarchical tree structure, facilitates command and data flow. A novel web-based user-interface allows control of and navigation through the system for efficient problem tracing.

This paper covers the front-end control and monitoring system for the RPC subdetector; the online software components that allow for flexible configuration and comprehensive monitoring are described.

***Index Terms*—Control System, Software, CMS, Resistive Plate Chambers.**

I. INTRODUCTION

THE Compact Muon Solenoid (CMS) experiment [1] is one of the two general-purpose detectors observing proton-proton and heavy-ion collisions at the CERN Large Hadron Collider (LHC) [2]. Its compact geometry is built around a 6 m diameter superconducting solenoid with a 3.8 T magnetic field (Fig. 1). Inside the solenoid CMS is instrumented with a central tracker and hadronic and electromagnetic calorimeters; on the outside an iron return yoke is interleaved with gaseous detectors to trace muons. To make access to the subdetectors possible, the cylindrical barrel region is divided into five moveable wheels along the beam axis, and the endcap regions on both sides of the solenoid consist of three planar disks each.

Three different gaseous detector technologies are combined to trigger and reconstruct muons [3]. Drift Tube chambers (DTs) are arranged in coaxial layers in the barrel region to detect muons up to pseudorapidity $|\eta| < 1.2$. The endcap disks are equipped with Cathode Strip Chambers (CSCs) ($0.9 < |\eta| < 2.4$) to handle the higher rates and nonuniform magnetic field. Both regions are instrumented with Resistive Plate Chambers (RPCs) [4], and at present the RPC subdetector consists of 912 chambers covering $|\eta| < 1.6$. They are characterised by a time resolution $\approx$ 3 ns, crucial to assign muon candidates unambiguously to the corresponding 25 ns LHC Bunch Crossing (BX), and a spatial resolution of the order of 1 cm, fully adequate for the muon trigger system purposes.

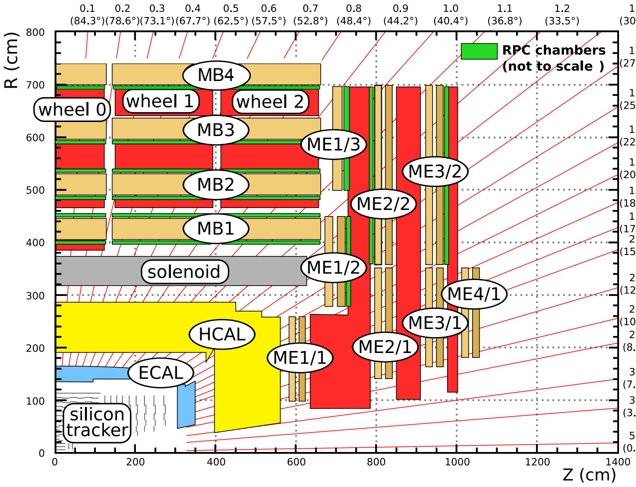


Fig. 1. Cross-section of a quadrant of the CMS experiment, showing the muon chambers in the barrel (MB) and endcap (ME) regions. The interaction point is located in the lower left corner, the beam pipe along the z axis.

The requirements on the RPC front-end electronics during CMS operation call for a fast, reliable and easy to use control system, and a comprehensive user interface for efficient problem tracing. To achieve these goals, hardware communication algorithms have been revised, new hardware descriptions were implemented and software applications and interfaces were added to the existing online software infrastructure.

A short description of the implementation of the RPCs in CMS is given in Section II. Section III gives an overview of the hardware architecture used to access to the front-end control system. Section IV describes the corresponding software architecture and Section V goes through the different application interfaces.

II. CMS RESISTIVE PLATE CHAMBERS

A. Detector

CMS uses double-gap RPCs, with each 2 mm gas gap composed of two parallel bakelite electrodes (Fig. 2). The gaps are placed one on top of another with common copper readout strips in between. The strips have a pitch ranging from 1.7 cm

---

Manuscript received November 15, 2011.

F. Thyssen is with the Ghent University, Ghent, Belgium. Corresponding author: Filip.Thyssen@cern.ch, Department of Physics and Astronomy, Proeftuinstraat 86, B-9000 Ghent, Belgium.

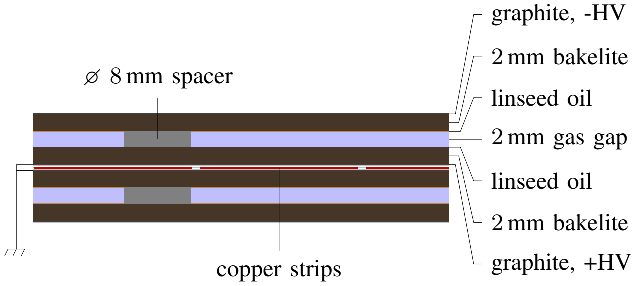


Fig. 2. Cross-section of a CMS Resistive Plate Chamber. The bottom gap mirrors the top one.

to 4.1 cm over a length of 47 cm to 125 cm, as dictated by the transverse momentum resolution needed for the muon trigger. They are operated in avalanche mode to safeguard the time resolution at high rates ($\sim 1\,\mathrm{kHz\,cm^{-2}}$).

B. Readout and Trigger

The signals induced on the strips when an ionising particle traverses an RPC are collected by up to 18 on-detector Front-End Boards (FEBs) per RPC. Depending on the region, the FEBs contain 2 or 4 ASICs handling 8 strips each, the so called FEB chips. To safeguard the time resolution, every FEB chip channel combines a threshold with a zero-crossing discriminator [5]. A monostable suppresses RPC afterpulses and defines the output pulse width, and a Low-Voltage Differential Signaling (LVDS) driver transmits the resulting pulses to Link Boards (LBs) located at the periphery of the CMS return yokes (balconies). The LBs synchronise and compress the data before sending them to the readout and trigger system in the CMS service cavern over ~ 100 m optical fibers [6].

C. Front-End Board Parameters

To ensure safe detector operation and optimise the detector performance, every FEB can be configured and monitored through an Inter-Integrated Circuit ($\mathrm{I^2C}$) interface. The main FEB parameters consist of the discriminator threshold, implemented as a Threshold Voltage (V_{Th}), and the output pulse width, defined by the Monostable Voltage (V_{Mon}). Both V_{Th} and V_{Mon} are common for the 8 channels of one FEB chip. Power-on default values have been preset before installation using PCB mounted potentiometers, and during the detector configuration phase they are replaced using commercial on-board DACs. ADCs give access to the actual values. To measure V_{Mon}, a voltage divider is needed to match the ADC range, so only half the resolution remains. For safe operation each FEB has at least one temperature sensor.

III. COMMUNICATION HARDWARE ARCHITECTURE

The hardware implementation of the front-end electronics control system outlined here is depicted in Fig. 3.

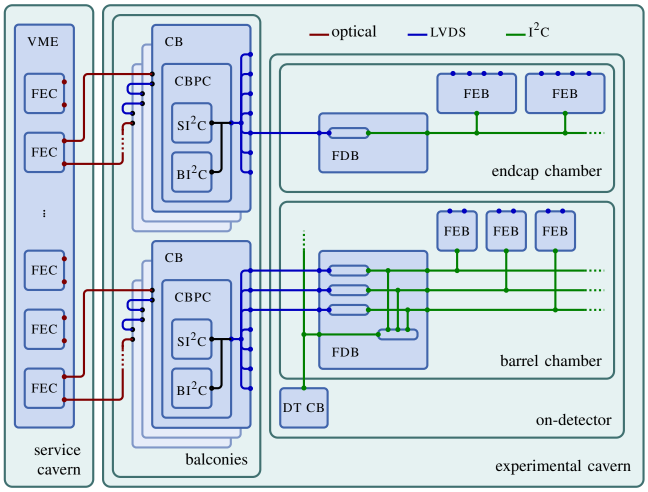


Fig. 3. Overview of the hardware architecture used for the CMS RPC front-end control system. The acronyms are explained in Section III.

A. Service Cavern to Balconies

For the RPC subdetector, communication with the electronics in the CMS experimental cavern is established using a set of Front-End Controllers (FECs) [7], installed in a VME crate in the service cavern. These FECs host 18 token rings, one for each side of every CMS return yoke holding RPCs (tower). Through optical links, they are connected to Control Boards (CBs) in the balconies. Copper interconnections between the CBs close the token rings. Inside the CBs, the communication over the token ring is handled by Communication and Control Unit (CCU) chips [8].

B. Experimental Cavern

From the periphery of the ⌀ 15 m wheels and disks, the CBs communicate with on-detector Distribution Boards (FDBs). These distribute the low-voltage supply to the FEBs and host the $\sim$1.2 k $\mathrm{I^2C}$ lines, each line serving three to six FEBs. For every $\mathrm{I^2C}$ line, a corresponding LVDS connection runs from the CB to the FDB, carrying the $\mathrm{I^2C}$ communication. This makes it possible to limit the logic in the FDB, and yet to cover the distance from balcony to RPC. The number of LVDS lines has been limited to three per $\mathrm{I^2C}$ line: the Serial Data Line (SDA) runs bidirectional over two lines, the Serial Clock Line (SCL) unidirectional from CB to FDB. This comes at the expense of losing the $\mathrm{I^2C}$-native clock stretching, which normally permits components to hold down the SCL line, effectively pausing communication.

C. $\mathrm{I^2C}$ Communication

A Programmable Controller inside every CB (CBPC) handles communication between the CCU chip and the LBs. It also takes care of the $\mathrm{I^2C}$ communication with the FDBs. To do so, both a Serial $\mathrm{I^2C}$ ($\mathrm{SI^2C}$) and a Block $\mathrm{I^2C}$ ($\mathrm{BI^2C}$) protocol have been implemented on the CBPC FPGA. The first allows sending and receiving single $\mathrm{I^2C}$ commands, whereas the latter receives a compact block of data describing a series of standard $\mathrm{I^2C}$ commands, extended with a pause to anticipate expected clock stretching.

In the barrel region, a redundant I^2C line is provided by the CMS DT subdetector. A multiplexer on the FDB connects three RPC I^2C lines to one I^2C line on an on-detector DT Control Board (DT CB).

IV. SOFTWARE ARCHITECTURE

The overall software architecture described in Sections IV-A and IV-B is summarised in Fig. 4.

A. *CMS RPC Online Software*

In the present RPC online software infrastructure, the control system for trigger and readout electronics consists of a set of C++ Cross Platform Data Acquisition (XDAQ) applications [9], embedded in the Trigger Supervisor (TS) framework [10]. RPC services (power and gas supply) are on the other hand managed by the RPC Detector Control System (DCS), implemented in PVSS (Prozessvisualisierungs- und Steuerungs-System) [11]. To ensure safe operation, DCS also has acces to various on-detector and environmental sensors. Both systems are supervised by the CMS Run Control and Monitor System (RCMS) [12] through the central TS and DCS respectively.

Being a crucial part of the readout and trigger chain, the front-end control has been integrated in the RPC trigger control system.

B. *Application Structure*

Within the RPC trigger control system, one Trigger Supervisor application (the RPC TS Cell) follows up CMS state transitions (Section V-B) and in turn manages all other RPC TS applications (workers). For every tower or FEC token ring, one worker (RPC Tower) takes care of the hardware communication. The communication with front-end electronics has been added to these workers to match the hardware architecture. FEBs and FDBs with communication line problems are moved to an artificial tower, the RPC DT Tower, which takes care of communication through the redundant I^2C lines. Communication between the TS applications is mediated by SOAP messages (Simple Object Access Protocol) over HTTP.

To keep track of front-end electronics data, states and reports (Section IV-D), they are collected in a central XDAQ application, the FEB Collector. To minimise overhead, these data are attached to SOAP messages in the Extended External Data Representation (EXDR) format, a binary data representation implemented in XDAQ with smaller encoding sizes and faster processing time than the standard SOAP XML representations. Using a XDAQ-PVSS SOAP Interface (PSX), values relevant for safe detector operation are forwarded to the RPC DCS and stored in a Conditions DataBase (CondDB).

All TS applications have access to a Configuration DataBase (ConfDB) by means of SOAP communication with a JAVA application that uses Hibernate for DB queries.

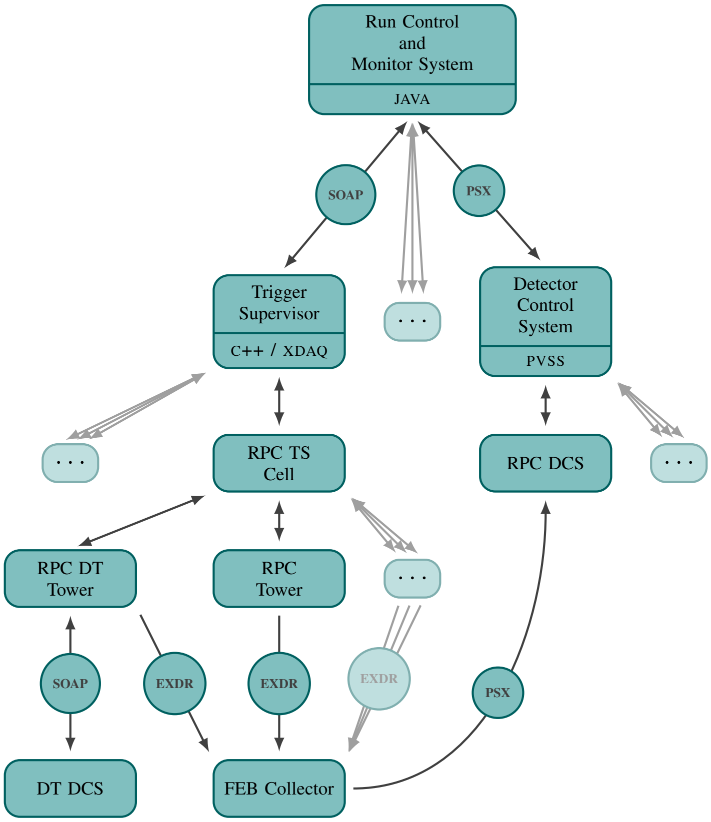


Fig. 4. Overview of the software architecture relevant for the CMS RPC front-end control system. The acronyms are explained in Section IV.

C. *Hardware Description*

The diverse RPC geometries dictate different hardware implementations of FEBs and FDBs throughout the detector. Breaking hardware and detector components down to their functional elements (hardware items), the structure is brought back to a fine grained tree, seamlessly incorporating these hardware differences. At run time the Configuration DataBase provides the actual hardware layout and the hardware item tree is built accordingly. Where applicable, the hardware items implement a common configuration and monitoring interface for coherent command flow down the tree.

To manage the three available I^2C interfaces, a common base describes I^2C communication. Commands and responses are separated, as required by both BI^2C and the lines provided by DT: for the first commands are buffered before sending them to the CB, for the latter a SOAP client-service interface with the DT DCS is needed not to interfere with DT hardware communications. The interface selection is described in the loaded hardware layout, and the front-end electronics implementations are blind to their differences.

D. *Logging Mechanism*

Given the vast number of hardware components, a compact logging mechanism with predefined log messages was implemented. In this system, a log message is defined by its category and ID, with a named alias for ease of use. Both a hardware state and event description are associated, as such reducing

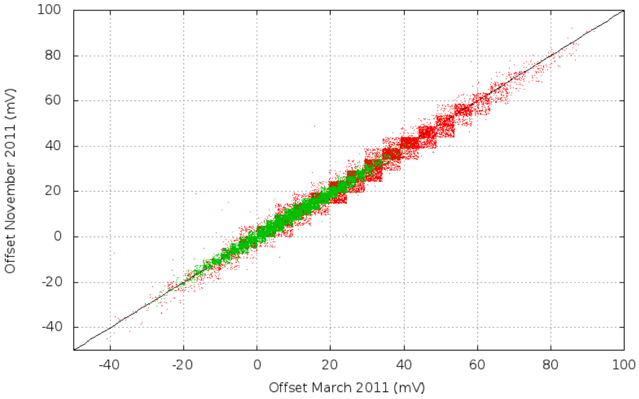


Fig. 5. Stability of the measured offsets for V_{Th} (green) and V_{Mon} (red).

hardware state updates and event reporting in code to instances of `log("event_name")` and `unlog("event_name")`. This algorithm enables compact in-memory storage of log messages and resulting states per hardware item. More detailed messages are reported using log4cplus for offline problem tracing by experts.

E. Configuration Data

To allow for optimisation of the detector performance in different scenarios, individual FEB parameter settings are stored in ConfDB, associated to configuration keys. The configuration keys are linked to global CMS configurations, and can be selected manually for commissioning purposes.

Reading back the ADC values for the DAC parameters V_{Th} and V_{Mon}, stable offsets have been found between written and measured values (Fig. 5), so also these are stored and monitored.

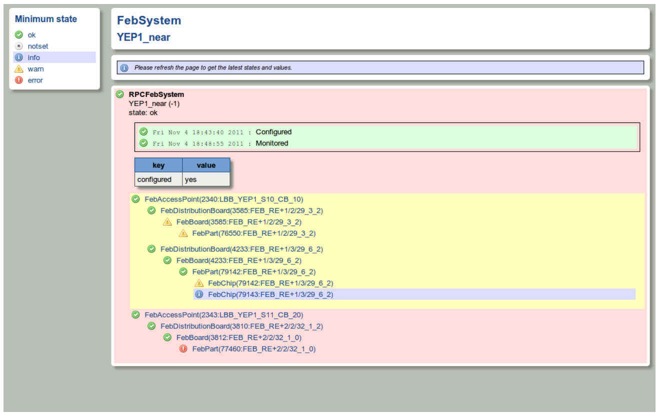
A separate Graphical User Interface (GUI) has been built to interact with the ConfDB to facilitate problem-solving parameter adjustments.

F. Performance

Parallelisation of front-end electronics communication to the level of the FEC token rings has brought its communication speed up to the physical limit, and the integration in the trigger and readout control system allows synchronisation with other activities on the towers. Storage of the V_{Th} and V_{Mon} offsets in ConfDB keeps from having to measure them during time-critical operations and allows separation of true offsets and possible DAC problems. These factors, combined with an optimisation of the communication algorithm, brought the time needed for the configuration of the full system down by more than a factor of five compared to the previous implementation.

V. APPLICATION INTERFACES

In addition to the standard SOAP interfaces as used by RCMS and the TS applications, XDAQ implements web-interfaces for user interaction with the software. The flexibility of the latter has been exploited to give access to the front-end electronics in several ways: control panels were added to the TS applications, and access to hardware and detector trees has been implemented in the RPC Tower applications and the FEB Collector respectively.



FebSystem
YEP1_near

Please refresh the page to get the latest states and values.

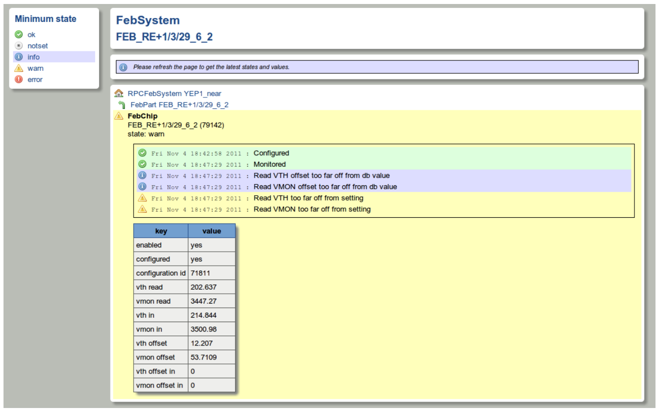
RPCFebSystem
YEP1_near (-1)
state: ok

```
Fri Nov 4 18:43:40 2011 : Configured
Fri Nov 4 18:48:55 2011 : Monitored
```

key	value
configured	yes

- FebAccessPoint(2340:LBB_YEP1_S10_CB_10)
  - FebDistributionBoard(3585:FEB_RE+1/2/29_3_2)
    - FebBoard(3585:FEB_RE+1/2/29_3_2)
      - FebPart(76550:FEB_RE+1/2/29_3_2)
  - FebDistributionBoard(4233:FEB_RE+1/3/29_6_2)
    - FebBoard(4233:FEB_RE+1/3/29_6_2)
      - FebPart(79142:FEB_RE+1/3/29_6_2)
        - FebChip(79142:FEB_RE+1/3/29_6_2)
        - FebChip(79143:FEB_RE+1/3/29_6_2)
- FebAccessPoint(2343:LBB_YEP1_S11_CB_20)
  - FebDistributionBoard(3810:FEB_RE+2/2/32_1_2)
    - FebBoard(3812:FEB_RE+2/2/32_1_0)
      - FebPart(77460:FEB_RE+2/2/32_1_0)

Fig. 6. Hardware tree of a full tower. The menu on the left allows filtering the tree by state. The icons of hardware items correspond to their state, the background colors to the worst state further down the tree.



FebSystem
FEB_RE+1/3/29_6_2

Please refresh the page to get the latest states and values.

RPCFebSystem YEP1_near
FebPart FEB_RE+1/3/29_6_2
FebChip
FEB_RE+1/3/29_6_2 (79142)
state: warn

```
Fri Nov 4 18:42:58 2011 : Configured
Fri Nov 4 18:47:29 2011 : Monitored
Fri Nov 4 18:47:29 2011 : Read VTH offset too far off from db value
Fri Nov 4 18:47:29 2011 : Read VMON offset too far off from db value
Fri Nov 4 18:47:29 2011 : Read VTH too far off from setting
Fri Nov 4 18:47:29 2011 : Read VMON too far off from setting
```

key	value
enabled	yes
configured	yes
configuration id	71811
vth read	202.637
vmon read	3447.27
vth in	214.844
vmon in	3500.98
vth offset	12.207
vmon offset	53.7109
vth offset in	0
vmon offset in	0

Fig. 7. Page of a hardware item further down the tree. The menu on the left allows filtering the tree by state. Since this hardware item is marked as printable, its parameters are shown. The time stamps and descriptions of the log messages are listed.

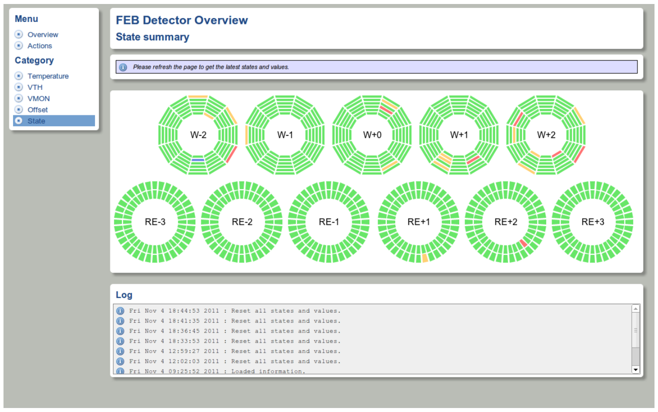
A. Configuration and Monitoring

For manual configuration or monitoring of the FEB system, a panel has been added both to the RPC Trigger Supervisor Cell and to the individual RPC Towers. After selection of a configuration key, each tower or the full system can be configured and monitored. Unlike the following user interfaces, this panel is embedded in the TS framework and uses Asynchronous JavaScript and XML (AJAX) for fast updates.

A log4cplus appender selects the log messages relevant for the user and displays them in the panel, and summaries are sent to the RPC Trigger Supervisor Cell's configuration panel.

B. RCMS State Transitions

To handle the complex CMS experiment's control system, a finite set of states has been defined, common for all subsystems. While the actions of the RPC TS applications for state transitions are elaborate, for the front-end electronics they have been kept to a minimum.



Please refresh the page to get the latest states and values.

```
Fri Nov 4 18:44:53 2011 : Reset all states and values.
Fri Nov 4 18:41:35 2011 : Reset all states and values.
Fri Nov 4 18:36:45 2011 : Reset all states and values.
Fri Nov 4 18:33:53 2011 : Reset all states and values.
Fri Nov 4 12:59:27 2011 : Reset all states and values.
Fri Nov 4 12:02:03 2011 : Reset all states and values.
Fri Nov 4 09:25:52 2011 : Loaded information.
```

Fig. 8. Detector overview as shown on the FEB Collector application. Each colored unit represents the state of an RPC chamber on a CMS wheel (W) or endcap disk (RE), and can be clicked to browse down its tree. The menu on the left allows viewing the states per log category.

FEB configurations are stable between CMS runs, so configuration ($\approx 137\,\mathrm{s}$) and a first monitoring cycle ($\approx 254\,\mathrm{s}$) of the system are executed immediately after powering on the electronics. This reduces the state transition actions to loading the configurations into memory for monitoring reference, and initiating automatic monitoring of the FEB parameters during CMS runs. Using this combination, the software continuously attempts to detect problems.

C. Hardware and Detector Tree Browsers

The front-end hardware item trees described above are browsable through a web-interface within the CMS network (Fig. 6 and 7). Access to the logged events and monitored data smooths online problem tracing.

Since all data and log message IDs are forwarded to the FEB Collector, a second tree can be built. For the FEB Collector, it was chosen to organise the hardware items in a tree representing the detector logic rather than the plain hardware logic. A Scalable Vector Graphics (SVG) representation of the detector states gives the user a pictorial overview of the complete RPC subdetector (Fig. 8). It allows browsing down the tree, enabling fast problem spotting and investigation.

VI. CONCLUSION

Throughout the 2011 LHC p-p collision period the here described system has been proven to be a reliable and useful tool for the RPC subdetector. Easy access to relevant information for both shifter and expert enabled prompt response in case of problems. Compared to the previous software, configuration speed has increased by more than a factor of five, and monitoring has been automatised without interfering with the existing hardware communication.

REFERENCES

[1] The CMS Collaboration, "The CMS experiment at the CERN LHC," *J. Instrum.*, vol. 3, p. S08004, 2008, doi:10.1088/1748-0221/3/08/S08004.
[2] L. Evans and P. Bryant, "LHC Machine," *J. Instrum.*, vol. 3, p. S08001, 2008, doi:10.1088/1748-0221/3/08/S08001.
[3] The CMS Collaboration, *The CMS muon project: Technical Design Report*. Geneva: CERN, 1997, CERN-LHCC-97-032; CMS-TDR-003.
[4] R. Santonico and R. Cardarelli, "Development of resistive plate counters," *Nucl. Instrum. Methods*, vol. 187, p. 377, 1981, doi:10.1016/0029-554X(81)90363-3.
[5] M. Abbrescia *et al.*, "New Developments on Front End Electronics for the CMS Resistive Plate Chambers," *Nucl. Instrum. Meth. A*, vol. 456, p. 143, 2000.
[6] K. Banzuzi *et al.*, "Link board for CMS RPC: Production issues," *Nucl. Instrum. Meth. A*, vol. 587, p. 110, 2008.
[7] F. Drouhin *et al.*, "The CERN CMS Tracker Control System," in *Nuclear Science Symposium Conference Record, 2004 IEEE*, vol. 2, Oct 2004, p. 1196.
[8] C. Paillard, C. Ljuslin, and A. Marchioro, "The CCU25: a network oriented communication and control unit integrated circuit in a 0.25 μm CMOS technology," in *8th Workshop on Electronics for LHC Experiments*, Colmar, France, 2002, p. 174.
[9] V. Brigljevic *et al.*, "Using XDAQ in Application Scenarios of the CMS Experiment," in *Proceedings of CHEP 03*, Mar 2003, p. MOGT008.
[10] I. Magrans de Abril *et al.*, "Conceptual design of the CMS Trigger Supervisor," *IEEE Transactions on Nuclear Science*, vol. 53, no. 2, p. 474, Apr 2006.
[11] G. Polese *et al.*, "The detector control systems for the CMS Resistive Plate Chamber," *J. Phys.: Conf. Ser.*, vol. 219, no. 2, p. 022019, Apr 2010.
[12] M. Bellato *et al.*, "Run Control and Monitor System for the CMS Experiment," *eprint arXiv:cs/0306110*, 2003.