

Design of online control and monitoring software for the CPPF system in the CMS Level-1 trigger upgrade

Li-Bo Cheng^{1,2} · Peng-Cheng Cao^{1,2} · Jing-Zhou Zhao² · Zhen-An Liu²

Received: 5 February 2018 / Revised: 21 March 2018 / Accepted: 29 March 2018 / Published online: 6 October 2018
© Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Chinese Nuclear Society, Science Press China and Springer Nature Singapore Pte Ltd. 2018

Abstract The concentration preprocessing and fan-out (CPPF) system is one of the electronic subsystems of the upgraded Compact Muon Solenoid (CMS) Level-1 trigger system. It includes, in hardware, eight specially designed CPPF cards, one CMS card called AMC13, one commercial Micro-TCA Carrier HUB (MCH) card, and a Micro-TCA shelf. Powerful online software is needed for the system, including providing reliable configuration and monitoring for the hardware, and a graphical interface for executing all actions and publishing monitoring messages. Further, to control and monitor the large amount of homogeneous hardware, the SoftWare Automating conTrol of Common Hardware (SWATCH) concept was proposed and developed. The SWATCH provides a generic structure and is flexible for customization. The structure includes a hardware access library based on the IPbus protocol, which assumes a virtual 32-bit address/32-bit data bus and builds a simple hardware access layer. Furthermore, the structure provides a graphical user interface, which is based on modern web technology and is accessible by web page. The CPPF controlling and monitoring online software was also customized from a common SWATCH cell, and provides a

finite state machine (FSM) for configuring the entire CPPF hardware, and five monitoring objects for periodically collecting monitoring data from five main functional modules in the CPPF hardware. This paper introduces the details of the CPPF SWATCH cell development.

Keywords CPPF · CMS · Level-1 trigger · SWATCH · Monitor · IPbus

1 Introduction

The Level-1 trigger (L1T) system at the compact Muon Solenoid (CMS) experiment includes two subsystems: a muon trigger system and a calorimeter trigger system. The muon trigger system receives data from muon detectors, including the drift tube (DT) in the barrel region, cathode strip chamber (CSC) in the endcap region, and resistant plate chamber (RPC) in both the barrel and endcap regions. The L1T system was upgraded to cope with higher collision rates caused by the increased collision energy and luminosity of the Large Hadron Collider (LHC) [1].

Figure 1 shows the structure of the muon upgrade L1T system, which is composed of a global trigger and three parts for the barrel, endcap, and overlap regions. The hits from the DT, CSC, and RPC are sent to corresponding regional track finders for muon track building. The built muon tracks in different regions are transmitted to the global muon trigger, where muon information is collected to make primitives. This muon system also has a layer for detector data concentration: concentration preprocessing and fan-out (CPPF) and TwinMux.

The CPPF system [2] is one of the upgraded L1T subsystems. It receives all RPC hits in the endcap region and

This work was supported by the National Natural Science Foundation of China (No. 11435013) and the National Key Program for S&T Research and Development(No. 2016YFA0400104)

✉ Zhen-An Liu
liuza@ihep.ac.cn

¹ School of Physical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

² State Key Laboratory of Particle Detection and Electronics, Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China

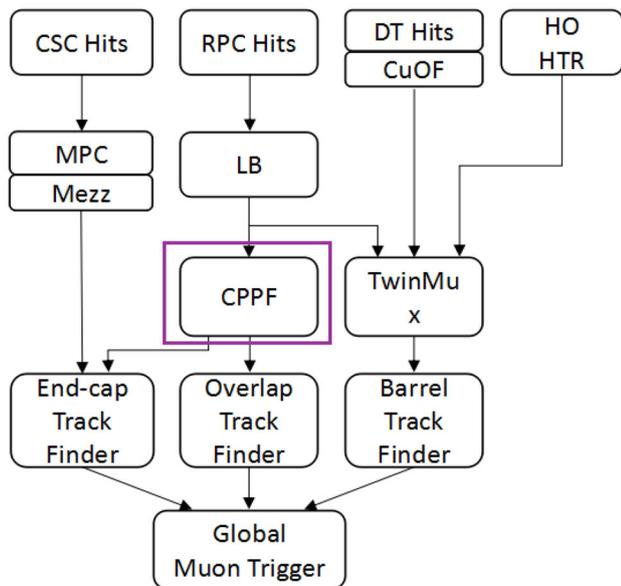


Fig. 1 Structure of the CMS Level-1 muon trigger system

all hits in the overlap region from the endcap, preprocesses them for cluster finding and angle conversion, and then concentrates the results and fans them out to the endcap muon trigger finder (EMTF) and overlap muon trigger finder (OMTF).

To simplify the system maintenance, the upgraded Level-1 trigger system was constructed with a large number of electronic cards based on the uTCA modular standard, which is better for a general-purpose common control compared with the existing legacy system [3]. To develop common online software for them, the SoftWare Automating conTrol of Common Hardware (SWATCH) project was proposed and developed in CMS [4, 5]. The SWATCH is based on a C++ library and provides a generic structure for controlling and monitoring the hardware. The generic structure defines a reliable hardware access layer based on the IPbus protocol, including an application interface for writing and reading data to and from the hardware. In addition, the structure provides flexible interfaces of some common processes for configuration and objects for monitoring and can be connected with the CMS L1T central cell for global control. Furthermore, a graphical interface was developed for easy execution of the actions, and publishing monitoring messages. Specific subsystems can inherit the structure and customize the contents of the configuration and monitoring interface with its special requirements.

The online control and monitoring software for the CPPF system was customized from the general SWATCH cell. The software can provide necessary configurations for all the hardware and retrieve the running status of the hardware in real time. In addition, the software was

integrated to the CMS L1T central system for global control and management in April 2017. This paper describes this development and is organized as follows. In Sect. 2, we briefly introduce the functionalities of the CPPF system. In Sect. 3, we explain the details of the CMS SWATCH project. Integration of CPPF SWATCH into the central online system is described in Sect. 4. Test and Running is described in Sect. 5 and finally, a conclusion is provided in Sect. 6.

2 CPPF system description

According to the schedule of the CMS L1T system upgrade, the functionality of the CPPF system is responsible for aligning [6–10], concentrating RPC data (those from the overlap region in the endcap and all data from the endcap), and providing preprocessed information for muon track building. The preprocessing includes cluster finding from the received RPC hits, and converting the position information of the cluster to angular information. The hardware design of the CPPF system is based on the uTCA standard, which provides an embedded, scalable architecture, and offers the flexibility to build a robust system. There are two important components defined in the uTCA protocol, Micro-TCA Carrier HUB (MCH) and Advanced Mezzanine Card (AMC). The MCH is the main management module that enables and controls different components of the uTCA system, and the AMC is the main functional module that is implemented with telecommunication functions and allows the application to be scalable [11]. The hardware of the CPPF system includes one AMC13 card, one commercial MCH module, and eight CPPF boards [12] and is installed in one uTCA crate. Figure 2 shows the installed CPPF hardware in the CMS



Fig. 2 Installed CPPF hardware in USC55 (Color online)

Underground Service Cavern 55 (USC55), which is the CMS counting room.

The commercial MCH (MCHs) is used for crate management and provides an Ethernet connection with the crate, as well as data exchange between CPPF boards. The AMC13 is a customized MCH module by CERN, and sits in the second MCH slot of the crate. It provides trigger timing and control (TTC) signals, a feedback mechanism for the trigger throttling system (TTS) in case the data buffers become full, and a high-speed link to the data acquisition (DAQ) system [13]. The CPPF board is one kind of AMC in the uTCA system and is implemented with the main functions needed for RPC data transmission and processing. Each CPPF board is responsible for 90° of the RPC overlap region data transmission; eight CPPF boards are needed in total.

The functional diagram of a CPPF board is shown in Fig. 3. Every CPPF board has five main functional blocks [12]:

- *Optical input module* Receiving RPC hits from the RPC link board (LB) at 1.6 Gbps with 19 channels.
- *Algorithm module* Preprocessing received data with clusterization and an angle conversion algorithm.
- *Optical output module* Transmitting preprocessed data to the EMTF system and OMTF system at 10 Gbps with 12 channels.
- *Readout module* Recording received data and preprocessed data and uploading them to AMC13 by backplane connection, and finally to the DAQ system.
- *TTC module* Getting the TTC clock and processing TTC signal from AMC13 through the backplane connection.

Furthermore, the CPPF board was implemented with an Ethernet transmission module based on the UDP/IP protocol and IPbus protocol, in order to communicate with a PC server for hardware control and monitoring. IPbus is a

simple packet-based control protocol for modifying and reading memory-mapped resources in hardware [14]. It assumes the existence of a virtual bus with 32-bit address and 32-bit data and defines a handshake process to complement unreliable transmission of the UDP/IP protocol. In the CPPF board, all five functional modules are under the management of the IPbus master through an IPbus fabric block. In each module, the needed configuration and monitoring registers/memories are mapped with an address, which can be retrieved by the IPbus master through an address decoder.

The online software for the CPPF system hardware must provide configuration for all boards, and monitor the boards and report errors once errors appear during testing and running.

3 CMS SWATCH description

The SWATCH project was proposed and developed to control and monitor the hardware of the upgraded CMS Level-1 trigger system. Because the subsystems in the upgraded Level-1 trigger system have great commonality, SWATCH provides a generic structure for them by a set of abstract C++ classes. Special functionalities for each subsystem are implemented in classes that inherit from the generic C++ classes.

The communication between the hardware and the PC server is handled by the hardware access library (uHAL). It provides an end-user API for IPbus reading, writing, and read-modify-write (RMW) transactions, and a delayed dispatch mechanism to process multiple transaction requests. Furthermore, to solve the requirement of multiple-client access, ControlHub was formed, which implements the reliability mechanism defined in IPbus for the UDP packets and can simultaneously access one or more devices from multiple control applications [15]. As a result, there are two operating modes to access the hardware. One is the local-client mode in which the uHAL library communicates with the device directly over UDP/IP. The other one is the remote-client mode in which the uHAL library communicates with hardware exclusively via the ControlHub.

SWATCH also provides a graphical user interface (GUI), which is based on modern web technologies (ES6-JavaScript, SCSS, HTML5), and is accessible by web page. All the operations are available in the GUI, which means all the control and monitoring commands can be executed in the GUI. In addition, the GUI can be used to publish monitored status messages.

The hierarchy structure of CMS SWATCH is shown in Fig. 4. The hardware in each subsystem is managed by a corresponding subsystem cell, which is implemented with

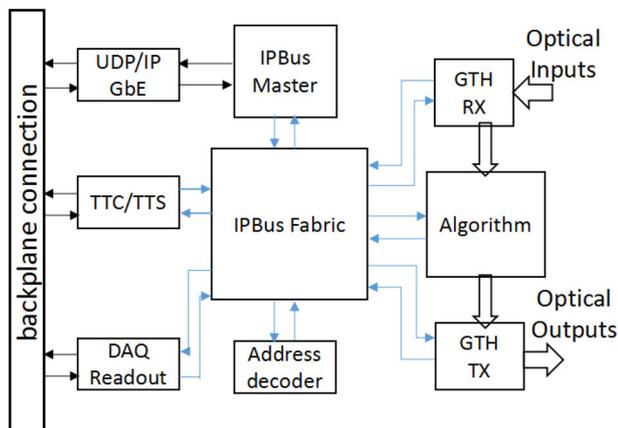


Fig. 3 Functional blocks of the CPPF board

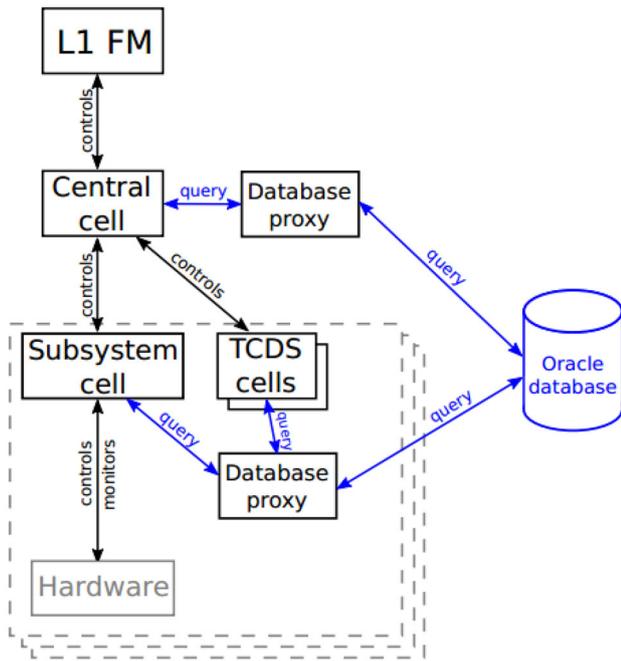


Fig. 4 CMS SWATCH structure, figure taken from Ref [17]. The CPPF SWATCH cell belongs to one of the subsystem cells for controlling and monitoring CPPF hardware

configurations for the hardware and monitoring for collecting status data from the hardware. The timing and control distribution system (TCDS) is a service system that distributes timing and control (synchronization) services to the electronics and receives back status information related to the readiness of the DAQ module to coordinate trigger rate [16]; it is managed by the TCDS cell. The configuration of all subsystems and TCDS systems for global running is coordinated by another cell, the central cell, which in turn is controlled by the Level-1 function manager (L1FM). The L1FM is the gateway from the top-level CMS run control. In addition, all the retrieval configuration parameters and monitoring data can be stored in a database.

The developed CPPF SWATCH cell belongs to one of the subsystem cells.

4 CPPF SWATCH cell implementation

The online software for the CPPF system should provide control and monitoring services for all CPPF hardware, and a GUI to support executing the commands visually. For the requirements of hardware configuration, it includes:

- The clock infrastructure in the CPPF must be initialized and configured correctly, and the AMC13 board must be properly initialized and configured.

- Optical input ports in CPPF boards must be properly configured, and received RPC data aligned with appropriate configured delays.
- Parameter values of data processing algorithms in CPPF boards must be properly set.
- DAQ-related functional modules must be configured correctly in both CPPF boards and the AMC13 board to record multiple bunch crossing (BX) clock window size data for both received RPC data and preprocessed data.

The monitoring objects mainly are the five functional blocks in the CPPF board; the software should collect monitoring data from the objects consistently, and then publish the status in the GUI with a message.

The CPPF online software is customized from the common SWATCH cell and is called the CPPF SWATCH cell.

The SWATCH cell provides interfaces for writing and reading to and from the hardware with the UDP/IP and IPbus protocol. The process is shown in Fig. 5. The PC server is implemented with a SWATCH cell and can transform read and write operations into corresponding IPbus transactions, which are contained within the payload of the UDP/IP Ethernet packet. Each transaction includes an address for the target, a write/read operation, and a data payload (only for the write operation). If the operation is inerrant, the hardware also answers the same format IPbus transaction to the PC server, and the answer transaction

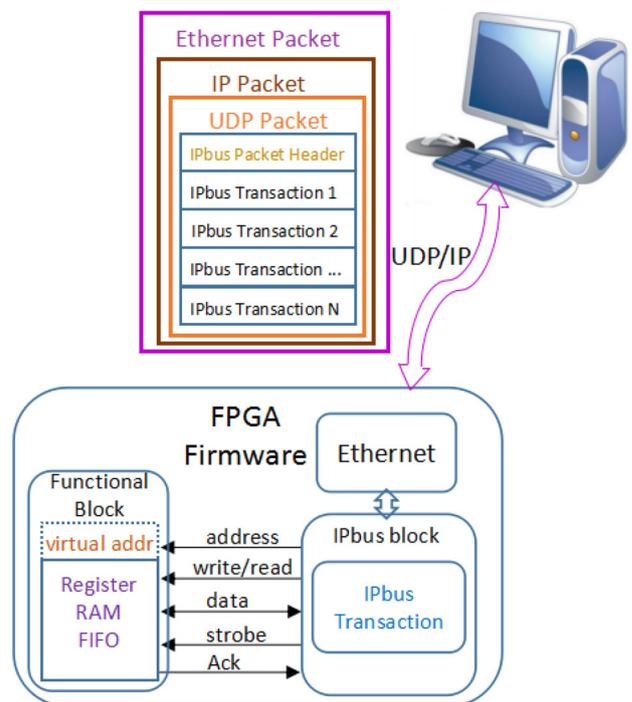


Fig. 5 Diagram of communication between the PC server and hardware FPGA by IPbus (Color online)

with read data for the read operation. Moreover, the hardware is also implemented with IPbus and UDP/IP-related firmware; it extracts the IPbus transaction from the Ethernet packet, and sends the transaction to the IPbus block. The IPbus block locks the memory target (such as the register, RAM, or FIFO) and performs the operation according to the received transaction. The memory target has a virtual address, which is predefined in firmware and is mapped with an IPbus address through the IPbus block. More details about how the PC server interconnects with the hardware by IPbus can be found in ref [18].

There are two fundamental actions defined in the SWATCH cell: **Command** and **Metric**, as shown in Fig. 6. The Command action is the basic building block for accessing the hardware and includes a single IPbus transaction for writing or reading.

The Metric action provides a standard interface for collecting monitoring data from the hardware, and persistently reporting error and warning status messages. It includes three parts:

- *Read Command* for reading monitoring data from the hardware.
- *Time Counter* for periodically operating the read command.
- *Status Message* from the comparison between the read monitoring data and predefined conditions. For example, in the CPPF optical input module, the “not in table” error means the received data is not in the table of 8B/10B code and is used for evaluating the input module function. This error bit is counted periodically, and can be read by a metric in the CPPF SWATCH cell. By comparing the read data with predefined conditions, which defines the “good status” as an error count of “0,” ‘warning status’ as an error count ranging in (1, 10), and ‘error status’ as an error

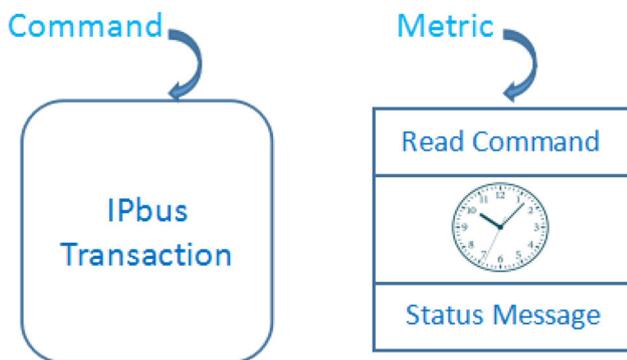


Fig. 6 Command and metric actions in the CPPF SWATCH cell. Each command includes a single IPbus request transaction. Each Metric includes a read command, a time counter, and a status message from the comparison result

count surpassing 10, the metric can produce the corresponding status message.

4.1 Controlling module development

Three actions are defined for the individual SWATCH control board:

- *Command* One-shot stateless action, e.g., reset the MMC.
- *Command sequence* Multiple commands chained and executed in succession.
- *FSM* Finite state machine, which defines the possible states of the subsystem; each transition between two states is typically a single command or command sequence.

Furthermore, the parameters used by the commands are provided through the **Gatekeeper**, which is a generic interface providing uniform access to the SWATCH application both from files and database.

The controlling module of the CPPF SWATCH is based mainly on a standard FSM defined in the SWATCH, as shown in Fig. 7. The states in the FSM include: *Halted (initial state) state*, *Engaged state*, *Synchronized state*, *Configured state*, *Aligned state*, *Running state*, and *Paused state*.

Commands and command sequences are executed for the transitions between the FSM states. The transitions are:

- Engage (Halted to Engaged): Start the configuration.

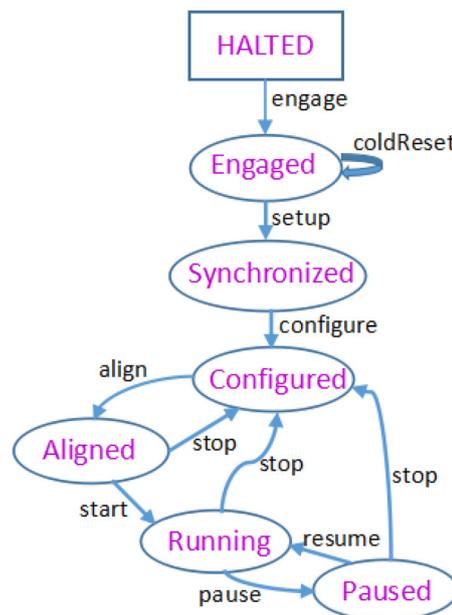


Fig. 7 Configuring FSM in the CPPF SWATCH cell

- Cold reset (Engaged to Engaged): Reboot AMC13 and CPPF boards with a different firmware image.
- Setup (Engaged to Synchronized): Reset AMC13 and CPPF boards, configure clocks and TTC blocks, and initialize optical inputs and outputs.
- Configure (Synchronized to Configured): Set the proper value for algorithm parameters, and configure the DAQ module.
- Align (Configured to Aligned): Configure optical input ports (GTH RX) and align input data.
- Start (Aligned to Running): Start running AMC13 and CPPF boards.

The configuration data for the parameters of each command can be taken from a single top-level XML file, or from a database. The XML file refers to one or more configure module files that can be split according to different functional modules in hardware. As a result, the split configuration files are much more reusable. The configuration data can also be stored in an online database, and can be retrieved by the SWATCH cell when needed during running and testing. In general, the XML file configuration mode is used for testing, whereas the database configuration mode is used for testing and running.

4.2 Monitoring module development

Similarly, two generic interfaces are defined in SWATCH for monitoring individual boards:

- *Metric* Individual piece of monitoring data, read from hardware, also reports the error and warning conditions in real time.
- *Monitorable object* Multi-level trees of monitoring information that are organized by metrics.

Values stored in metrics can be automatically reported to an external service and then stored in a database through a dedicated process.

The CPPF SWATCH cell must provide the current status of monitorable objects related to CPPF board function modules in real time, and report warnings or errors to the GUI interface.

There are five functional modules in the CPPF board: the TTC module, readout module, optical input port module, optical output port module, and algorithm module. Thus, five monitorable corresponding objects are needed:

- TTC object: Bunch counter, L1A counter, and orbit counter monitoring; bunch clock (40 MHz) locked status monitoring; BC0 (the first BX) locked status monitoring; single-bit error counter monitoring; double-bit error counter monitoring; and other functions.

- Readout object: TTS signal state monitoring, CPPF DAQ core ready status monitoring, event counter monitoring, and other functions.
- Input port object: GTH locked status monitoring, CRC error for received data status monitoring, and other functions.
- Output port object: GTH TX operating status during running monitoring and other functions.
- Algorithm object: Status of the key probe for algorithm monitoring.

Figure 8 is part of the monitoring graphical interface. It is obvious that the CPPF SWATCH cell includes monitoring for one AMC13 card and eight CPPF cards. Every CPPF card includes five monitorable objects: input object, output object, algorithm object, and readout object.

5 CPPF SWATCH cell test and running

Specific tests were developed for testing hardware functions in the CMS Level-1 trigger upgrade project [19]. One of them was the TTC test, which has a set of predefined conditions to see whether the configured TTC module in hardware is correct or not. Another test is the DAQ test, which can capture the recorded data by the readout module in hardware; the test prints expected readout data in the end if the readout module is well configured. CPPF hardware passed all the tests after being configured by the CPPF SWATCH cell, thus verifying that the performance of the cell is expected and correct.

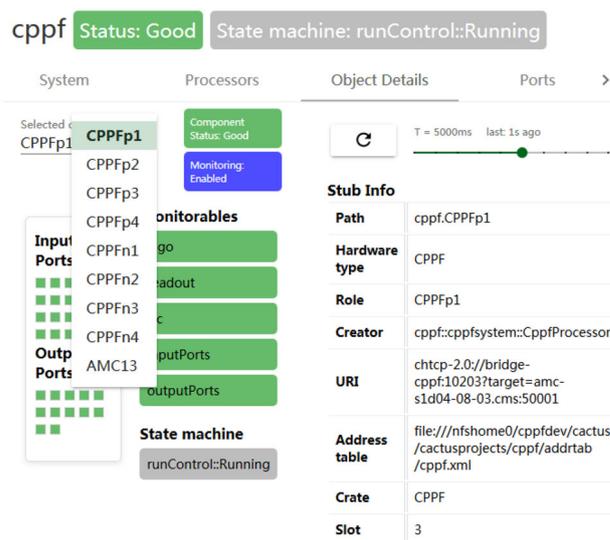


Fig. 8 CPPF SWATCH cell monitoring graphical interface (Color online)

After being well tested, the CPPF SWATCH cell was integrated into the central cell in April 2017. There are two running modes for the CPPF SWATCH cell: global running mode and local running mode. In the global running mode, the CPPF cell is connected with the central cell, and the FSM in the CPPF cell can be controlled by the central cell to configure the hardware in the CPPF system. Meanwhile, the central cell can collect monitoring information observed by the CPPF cell, and report the status in the GUI. However, the CPPF cell must be disconnected from the central cell first if it crashes or before entering into local running mode to perform local tests.

The integrated CMS Level-1 trigger system is accessible by a web application called Level-1 page (L1page), as shown in Fig. 9. The application implements access and control for the trigger processes and provides a GUI with all of the L1T monitoring information, including the running status and warnings and errors from all L1T subsystems. In addition, it provides a brief visual diagram of the system structure, including the interconnections among L1T subsystems and with corresponding detectors, which makes it much easier for non-experts (e.g., shifters) to identify the source of problems when they occur. In addition, it includes links to the relevant explanatory documentation and available experts, and information regarding the running (e.g., collision and cosmic rays).

The CPPF, as marked with a red box in Fig. 9, is also in the interface's list. The central cell can automatically control the CPPF SWATCH cell to configure the hardware and collect monitoring data; operators can also click it and enter the CPPF SWATCH cell and see the detailed status of the hardware of the CPPF system. The CPPF SWATCH cell has worked well since being integrated and is highly useful for system testing and running.

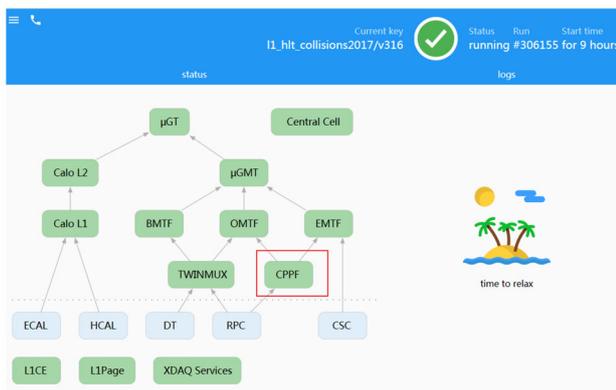


Fig. 9 CMS Level-1 trigger system L1page (Color online)

6 Conclusions

We developed an online software based on SWATCH to control and monitor CPPF system hardware in the upgraded CMS Level-1 trigger system. The software uses a standard defined FSM in the SWATCH platform for commands to configure the CPPF hardware; the command parameters can be taken from the configuration data in an XML file or database. Furthermore, the software provides standard defined monitoring objects including input ports, output ports, algorithm module, readout interface, and TTC interface, for monitoring and reporting their status in real time. The software also provides a GUI and is accessible by web page. Being connected with the CMS central cell, it can be controlled during CMS global running. The software was deployed in April 2017 and has run well since then.

Acknowledgements The authors are thankful to Prof. Darin Acosta, Prof. Costa Funtas, the former Level-1 trigger managers, Prof. Alexander Tapper, Prof. Emmanuel Francois Perez, the present managers, and Drs. Karol Bunkowski, Andrea Troissi, Andrew Brinkerhoff, et al., for their kind discussions and comments.

References

1. CMS collaboration. Technical proposal for the upgrade of the CMS detector through 2020 (Technical Proposal, CERN) (2011)
2. Z.A. Liu, *Design and Construction of CPPF System in CMS L1 Trigger Phase I Upgrade*. (Indico webpage) (2013), <http://indico.ihep.ac.cn/event/7102/session/9/contribution/92/material/slides/0.pdf>
3. A. Tapper, The CMS Level-1 Trigger for LHC Run II. (CERN document, No. CMS-CR-2016-303.) (2016), <http://cds.cern.ch/record/2238553>
4. J. Brooke, K. Bunkowski, I. Cali, SWATCH: common control SW for the uTCA-based upgraded CMS L1 Trigger. *J. Phys. Conf. Ser.* **664**, 082012 (2015). <https://doi.org/10.1088/1742-6596/664/8/082012>
5. G. Codispoti, Common software for controlling and monitoring the upgraded CMS Level-1 trigger, in *2017. At International Conference on Technology and Instrumentation in Particle Physics 2017*, Beijing, China, 21–26 May (2017)
6. D. Sun, Z. Liu, J. Zhao, Belle2Link: a global data readout and transmission for Belle II experiment at KEK. *Phys. Procedia* **37**, 1933–1939 (2012). <https://doi.org/10.1016/j.phpro.2012.01.036>
7. J. Zhao, Z.A. Liu, H. Xu, et al., A general xTCA compliant and FPGA based data processing building blocks for trigger and data acquisition system, in *2014 19th IEEE-NPSS Real Time Conference, RT 2014—Conference Records* (2015). <https://doi.org/10.1109/RTC.2014.7097528>
8. H. Lin, Z.A. Liu, O. Martineau-Huynh et al., A prototype of self-triggering front-end unit for radio detection of ultra high energy neutrinos, in *IEEE Nuclear Science Symposium Conference Record* (2013). <https://doi.org/10.1109/NSSMIC.2013.6829558>
9. M.N. Wagner, S. Fleischer, M. Galuska, et al., Prototype for the trigger-less data acquisition of the PANDA experiment 2014 IEEE real time conference, in *2014 19th IEEE-NPSS Real Time*

- Conference, RT 2014—Conference Records* (2015). <https://doi.org/10.1109/RTC.2014.7097531>
10. S. Yamada, R. Itoh, T. Konno, Common readout subsystem for the Belle II experiment and its performance measurement. *IEEE Trans. Nucl. Sci.* **64**, 1415–1419 (2017). <https://doi.org/10.1109/TNS.2017.2693297>
 11. VadaTech, MicroTca overview: a brief introduce to micro telecommunications computing achitecture concepts, Technical file Version 1.1 (2014). https://www.vadatech.com/media/article_MicroTCA_Overview.pdf
 12. C.J. Wang, Z.A. Liu, J.Z. Zhao, Design of a high throughput electronics module for high energy physics experiments. *Chin. Phys. C* **40**, 066102 (2016). <https://doi.org/10.1088/1674-1137/40/6/066102>
 13. E. Hazen, The AMC13XG: a new generation clock/timing/DAQ module for CMS MicroTCA. *J. Instrum.* **8**, C12036 (2013). <https://doi.org/10.1088/1748-0221/8/12/C12036>
 14. C.G. Larrea, K. Harder, D. Newbold, IPbus: a flexible Ethernet-based control system for xTCA hardware. *J. Instrum.* **10**, C02019 (2015). <https://doi.org/10.1088/1748-0221/10/02/C02019>
 15. R. Frazier, G. Iles, D. Newbold, Software and firmware for controlling CMS trigger and readout hardware via gigabit Ethernet. *Phys. Procedia* **37**, 1892–1899 (2012)
 16. M. Jeitler, CMS collaboration. The upgrade of the CMS trigger system. *J. Instrum.* **9**, C08002 (2014). <https://doi.org/10.1088/1748-0221/9/08/C08002>
 17. S. Bologna, G. Codispoti, G. Dirks, et al., SWATCH: Common software for controlling and monitoring the upgraded Level-1 trigger of the Compact Muon Solenoid experiment, in *2016 IEEE-NPSS Real Time Conference (RT)*, Jun. 6–10 (2016). <https://doi.org/10.1109/RTC.2016.7543077>
 18. R. Frazier, G. Iles, et al., *The IPbus Protocol (version 2.0)*, (Technique document) (2013), http://ohm.bu.edu/~chill90/ipbus/ipbus_protocol_v2_0.pdf
 19. I.M. De Abril, C.E. Wulz, J. Varela, Conceptual design of the CMS trigger supervisor. *IEEE Trans. Nucl. Sci.* **53**, 474–483 (2006). <https://doi.org/10.1109/TNS.2006.872631>