The Control System for the Front-End Electronics of the ALICE Time Projection Chamber

M. Richter, J. Alme, T. Alt, S. Bablok, R. Campagnolo, U. Frankenfeld, C. González Gutiérrez, R. Keidel, Ch. Kofler, T. Krawutschke, D. Larsen, V. Lindenstruth, B. Mota, L. Musa, K. Røed, D. Röhrich, M. R. Stockmeier, H. Tilsner, and K. Ullaland for the ALICE collaboration

***Abstract*—The ALICE detector is a dedicated heavy-ion detector currently built at the Large Hadron Collider (LHC) at CERN. The detector control system (DCS) covers the task of controlling, configuring and monitoring the detector. One sub-system is the control system for the Front-end electronics of the time projection chamber (TPC). It controls in total 216 readout systems with 4356 Front-End Cards serving roughly 560 000 channels.**

The system consists of a large number of distributed nodes in a layer-structured hierarchy. The low-level node controlling the Front-end electronics is an embedded computer system, the DCS board, which provides the opportunity to run a light-weight Linux system on the card. The board interfaces to the Front-end electronics via a dedicated hardware interface and connects to the higher DCS-layers via the DIM communication framework over Ethernet. Since the experiment will be running in a radiation environment, fault tolerance, error correction and system stability in general are major concerns. Already the low level devices carry out intelligent error handling and act automatically upon several conditions. This paper presents the architecture of the system, the application of the DCS board and experiences from integration tests.

***Index Terms*—Control systems, distributed computing, gas detectors, microcontrollers, monitoring, nuclear physics, particle measurements.**

I. Introduction

THE ALICE experiment described in [1] will investigate Pb-Pb collisions at a center of mass energy of about 5.5 TeV per nucleon pair and p-p collisions at 14 TeV. The detectors are optimized for charged particle multiplicities of up to $\mathrm{dN_{ch}}/\mathrm{d}\eta$ of 8000 in the central rapidity region. In general, the Detector Control System (DCS) covers the tasks of controlling the cooling system, the ventilation system, the magnetic fields and other supports as well as the configuration and monitoring of the Front-end electronics. Detailed information on the components and architecture can be found in [2]. The various components not concerning the TPC Front-end electronics will not be investigated further.

It is important to notice that the control system is detached from the data-flow. The data is transported from the Front-end electronics to Data Acquisition (DAQ) through an optical link. The main task of the control system is to avoid occurring system errors interrupting the data-flow.

Sharing of devices between different sub-systems is avoided whenever it is possible, so that independent operation is ensured. This is also an important issue in development and commissioning of the system, so that each sub-system can be debugged and tested separately from other parts of the system. This technique is called *partitioning* and is a widely used feature in the design of ALICE.

The time projection chamber (TPC) is one of the main tracking detectors of the ALICE experiment. Charged particles ionize the gas volume on their way through the detector, the produced electrons drift in an electromagnetic field towards the end-caps where the charge is amplified and collected by a two-dimensional readout system. Together with the drift time this provides a three-dimensional resolution. The TPC consists of 36 sectors which are read out by 4356 Front-End Cards (FEC) serving roughly 560 000 channels. All FECs have to be configured and monitored. Furthermore programmable logic devices (PLDs) are widely used in all hardware devices of the TPC Front-end electronics to keep the system open and flexible. The configuration thus must include the upload of firmware to the FPGAs.

The ALICE detector works in a radiation environment. In such an environment errors in the Front-end electronics will occur, that may lead to a malfunctioning of the system. It is of high importance to detect and repair these errors as soon as possible, to prevent permanent damage and errors in the data-stream. Physical shielding of the sensitive parts cannot be implemented, as this will effect the particle measurements. The limited accessibility of the ALICE detector is also an important aspect. Hardware errors that may have occurred cannot be repaired without dismantling the whole detector.

This paper is intended to present the working principle of the whole system. The functional components will be introduced and described briefly. The first section describes the hardware architecture. The software architecture and components are sketched in the next section. The paper will be completed by a section dedicated to the issues concerning radiation tolerance and a section on experiences made during integration tests.

Manuscript received June 17, 2005; revised September 5, 2005.

M. Richter, J. Alme, S. Bablok, D. Larsen, D. Röhrich, and K. Ullaland are with the Department of Physics and Technology, University of Bergen, N-5007 Norway (e-mail: Matthias.Richter@ift.uib.no).

T. Alt, V. Lindenstruth, M. R. Stockmeier, and H. Tilsner are with the Kirchhoff Institute of Physics, University of Heidelberg, D-69120 Heidelberg, Germany.

R. Campagnolo, C. González Gutiérrez, B. Mota, and L. Musa are with CERN, European Organization for Nuclear Research, CERN CH-1211 Geneva, Switzerland.

U. Frankenfeld is with the Gesellschaft für Schwerionenforschung (GSI), 64291 Darmstadt, Germany.

R. Keidel and Ch. Kofler are with the Center for Technology Transfer and Telecommunications, University of Applied Science Worms, 67549 Worms Germany.

T. Krawutschke is with the Institute of Communication Engineering, University of Applied Sciences Cologne, 50679 Cologne, Germany.

K. Røed is with the Faculty of Engineering, Bergen University College, N-5020 Bergen, Norway.

Digital Object Identifier 10.1109/TNS.2006.874726

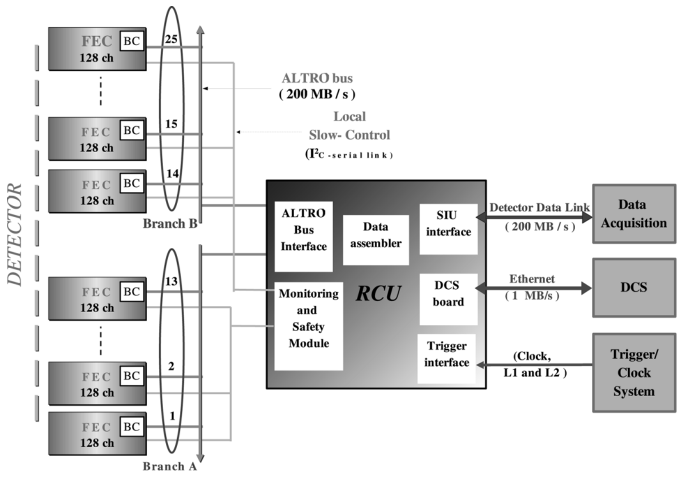

Fig. 1. Components of the TPC Front-end electronics and dataflow. The Readout Control Unit (RCU), the Board Controller (BC), and the DCS board are presented below.

II. Hardware Architecture

A. The TPC Front-End Electronics

Each of the 36 sectors of the TPC is read out by six identical subsystems. The TPC detector uses a specific hardware device, the Readout Control Unit (RCU), to control a set of Front-End Cards (FECs). An RCU contains the RCU motherboard, from now on referred to as RCU board, which hosts two additional interface boards customized for the ALICE experiment. The Detector Data Link Source Interface Unit (DDL SIU) is the ALICE standard interface to the DAQ. The second card, the DCS board, is an embedded computer which implements the DCS/Trigger interface.

Fig. 1 shows an overview of the electronics. Between 18 and 25 FECs are connected to one RCU board via two bus systems. A fast 200 MB/s bus, the *ALTRO* bus, is intended to transport the event-data and the configuration data between RCU and FECs. The Slow Control bus allows monitoring and controlling the cards without interfering with the data readout process. The event-data is converted from an analog to a digital signal by the ALTRO chips ([3]) on the FECs. The ALTRO provides also functionality for digital data processing and has the ability to reduce the data volume significantly. The data is then handled by the RCU and shipped out through an optical link that is sited on the SIU card. The complex data readout chain and the SIU card are not investigated further in this paper. More details about the electronics can be found in [4].

B. The Readout Control Unit

The design of the RCU board is based on an FPGA with firmware update possibility and enough space to host other tasks in addition to the readout. Among others, it contains modules for data assembling, bus interfaces, a Monitoring and Safety Module and trigger interfaces. The DCS uses the RCU board as a sub-node to the DCS board. Specialized tasks which are adapted to the underlying hardware run on the low-level nodes while the high-level control tasks are running on the node itself. The functional parts concerning the DCS are introduced briefly. For further comprehensive information we refer to [5]. Among the firmware modules of the RCU, the Monitoring and Safety Module (MSModule) plays an important role. It is the actuator on the lowest level. The module monitors the health status of the FECs. It implements safety functions and acts automatically on the assertion given by the Board Controller of the FEC. In the case of a hard error (e.g., temperature or currents over thresholds), the corresponding FEC is switched off immediately to avoid damage of the system. By this means, the alarm is handled as close as possible to the source and the response time is minimized. A recovery procedure follows to diagnose the occurred error. These operations are independent from the communication to the other cards. In the case of a soft error, the FEC is removed from the data readout. The message channel of the DCS is used to signalize the alarm/error upwards and the operator will be informed. During normal operation the MSModule provides access to monitoring data. The data transport is described in Section III.

The ALTRO Bus Interface module provides access to the FECs through command sequences. It implements a sequencer which decodes instructions written to the RCU Instruction Memory. The instructions are executed in the form of a single ALTRO instruction (RCU micro instruction) or sequence of them (RCU macro-instruction). The macro instructions allow, for example, to write or to verify the content of a given ALTRO pedestal memory. Another example is an RCU macro that handles a complete trigger sequence to exercise the readout chain. The DCS board exploits these methods to read and write data from/to registers and memory on the FECs. The ALTRO Bus Interface module is widely used by the DCS during configuration of the Front-end electronics.

C. The Board Controller on the FEC

The Board Controller (BC) sited on the Front-End Card hosts the low-level monitoring functionality. It controls automatically critical values like currents, voltages and temperatures. Every FEC contains a 10- bit, 5-channel ADC with an on-chip temperature sensor and an $I^2C^{®}$ interface. In addition to the temperature, two voltages and two currents (analogue and digital) are measured at the input of the FEC. The BC logic includes an $I^2C^{®}$ master to read this ADC. Every 2 ms the BC reads the five parameters and updates the corresponding registers. The BC contains also the configuration, status and error registers, and a set of counters that measure a number of critical signals (e.g., the Level-1 and Level-2 triggers). At power-up, the reference range for the monitored quantities are downloaded into the BC by the RCU. As soon as one of these parameters goes out of range, the BC asserts an interrupt. The assertion is handled by the MSModule of the RCU firmware.

During the configuration phase, the communication between the RCU and the Board Controller is possible via both the ALTRO bus and the Slow Control bus. In the data-taking phase the latter can be used without perturbation of the readout process.

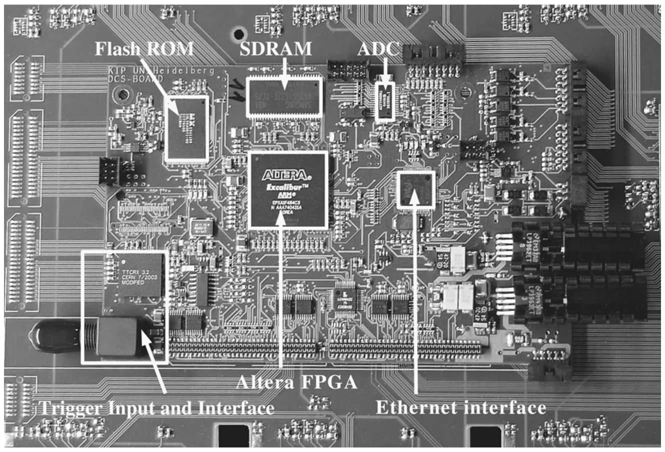

Fig. 2. The DCS board version 1.52.

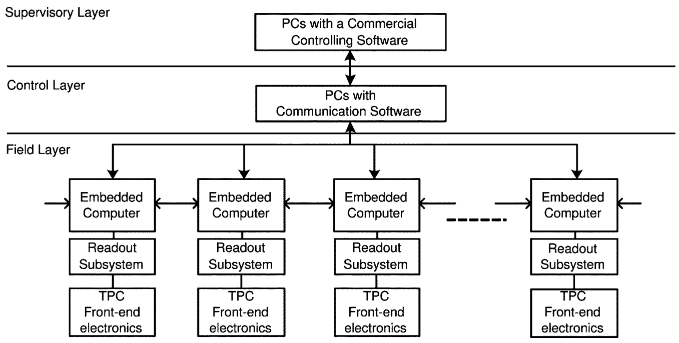
Supervisory Layer
PCs with a Commercial Controlling Software
Control Layer
PCs with Communication Software
Field Layer
Embedded Computer
Readout Subsystem
TPC Front-end electronics

Fig. 3. Schematic view of the software architecture

D. DCS Board Embedded Computer

The DCS board is used in several detectors of the ALICE experiment and flexibility has always been a major concern. The core of the system is an Altera EPXA1, containing a 32 bit ARM processor with cache and Memory Management Unit (MMU). Among other features there are 100k gates of Programmable Logic Device (PLD) available.

In addition to the FPGA, the board hosts a radiation tolerant 8 MB Flash ROM, 32 MB SDRAM, an Ethernet interface, an Analog–Digital Converter (ADC) for voltage and temperature monitoring, a JTAG connector, as well as dedicated data-lines to the RCU board connector. All these components make the DCS board a custom-made, fully-functional computer.

The components of the DCS board make it capable to run a light-weight version of Linux. The PLD in combination with the Linux operating system is the chief cause for the flexibility.

Registers and memory on the RCU board are accessible from the Linux operating system on the DCS board, either directly or indirectly. For that purpose memory mapped interfaces are defined and can be accessed via device drivers which define an interface between software and hardware. A change in the firmware usually only requires adaption of the driver. This modularization enables easy design changes without harming the whole system.

The DCS boards work in parallel in a distributed system, performing sophisticated error-handling and other tasks related to the control system. In addition a DCS board is able to reconfigure its neighboring boards via a JTAG connection in case of malfunction.

Fig. 2 shows the final version of the device. A detailed description of the functionality and parameters can be found in [6].

III. Software Architecture and Components

A. Functional Layers

A sketch of the system is given in Fig. 3 showing the principles of the architectural layout. From top to bottom this is:

- *The Supervisory Layer*
  The Supervisory Layer consists of a number of PCs and provides user interfaces to the operator. It also interfaces to external systems and services, e.g., the LHC.
- *The Control Layer*
  The Supervisory Layer communicates with the Control Layer mainly through a LAN network. This layer consists of PCs, PLCs (Programmable Logic Cells) and PLC like devices. The Control Layer collects and processes information from the Field Layer, as well as sending commands and information from the Supervisory Layer to the Field Layer. It also connects to the Configuration Database.
- *The Field Layer*
  The Field Layer consists of all field-devices, sensors, actuators and so on. The DCS board and the readout electronics are located in this layer.

The tasks of the three layers are carried out by dedicated programs. They work in parallel, feeding the operator with useful information concerning the status of the system, or responding to commands given at the top-level. The components which cover the task within the three functional layers are shown in a more detailed view in Fig. 4.

A SCADA (Supervisory Control And Data Acquisition) system acts in the Supervisory Layer, through which the operator can access and monitor data points related to the hardware devices. A commercial controlling software, PVSS (Prozess-Visualisierungs- und Steuerungs- System by ETM[1]), has been chosen for the ALICE experiment. The DCS is not restricted to this specific controlling software but can feature any SCADA system.

The PVSS connects to the *InterComLayer*, a specific communication software acting as the Control Layer and connecting the hardware devices in the Field Layer to the controlling system in the Supervisory Layer. The system uses the communication framework *DIM* (Distributed Information Management System, [7]), which is based on the client-server principles. Several abstraction layers have been introduced:

- PVSS and InterComLayer communicate through a specific interface, the Front-End Device (FED), which is common among different sub-detectors within the ALICE experiment. The InterComLayer implements a server which the PVSS can subscribe to as a client.
- Each hardware device implements a Front-End Electronics Server (FeeServer), which the InterComLayer subscribes to as a client.

[1]ETM professional control GmbH, http://www.etm.at

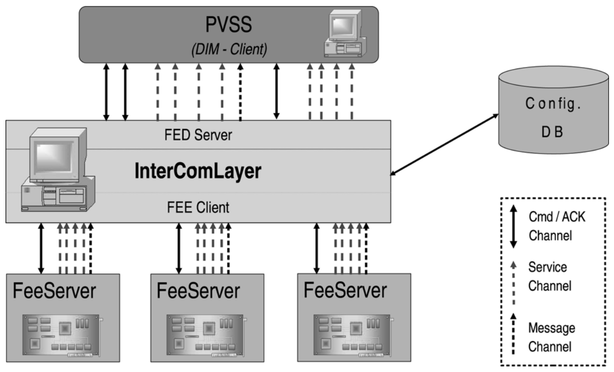

Fig. 4. Software components and data flow

The InterComLayer connects to several FeeServers and pools data before distributing it to the SCADA system. Vice versa the InterComLayer distributes configuration data to the FeeServers. In addition it implements an interface to the Configuration Database containing all specific configuration data for the hardware devices. A detailed description of the communication software can be found in [8]. A few features will be outlined here.

B. Communication Protocol

Communication between all layers is based on the DIM protocol. DIM is an open source communication framework developed at CERN. It provides a network-transparent inter-process communication for distributed and heterogeneous environments. TCP/IP over Ethernet is used as transport layer. A common library for many different operating systems is provided by the framework. DIM implements a client-server relation with two major functionalities.

- Services: The DIM server publishes so-called services and provides data through a service. Any DIM client can subscribe to services and monitor their data. The DIM clients get notified about current values via a callback from the DIM server.
- Commands: A DIM server can accept commands from DIM clients. Server and client have to agree on the format of the command.

A dedicated DIM name-server takes control over all the running clients, servers and their services available in the system. Each server registers at startup all its services and command channels. For a client the location of a server is transparent. The control system benefits from the features of the DIM framework, e.g., independence of machine architecture, process recovery and load distribution.

C. The Front-End Electronics Server

Fig. 4 shows a detailed view of the software components and the data flow. The DCS as described in this paper is based on so-called Front-End Electronics Servers (FeeServers) which run on the DCS board. A FeeServer abstracts the underlying Front-end electronics to a certain degree and covers the following tasks:

1) interfacing hardware data sources and publishing data;
2) receiving of commands and configuration data for controlling the Front-end electronics;
3) self-tests and Watchdogs (consistency check and setting of parameters).

The core of the FeeServer is device-independent. It provides general communication functionality, remote control and update of the whole FeeServer application. Some features are related to the configuration of the data publishing. In order to reduce network traffic, variable *deadbands* have been introduced. Data is only updated if the variation exceeds the deadband. The core can be used for different devices, i.e., different detectors of the ALICE experiment.

The device dependent functionality of the FeeServer is implemented in a separate part, the so-called *ControlEngine* (CE). The CE provides data access in order to monitor data points and executes received commands specific for the underlying hardware. The *ControlEngine* has contact to the specific bus systems of the devices. The access is encapsulated in Linux device drivers. This makes the functionality of the CE independent from the hardware/firmware version.

D. Intercomlayer

The InterComLayer takes the task of the Control Layer. It runs independently from the other system layers on a separate machine outside of the radiation area. It provides three interfaces (see also Fig. 4):

- *Front-End Electronics Client* (DIM client) to connect to the Field Layer;
- *Front-End Device Server* (DIM server) to connect to the Supervisory Layer;
- Interface to the Configuration Database, using a database client.

The InterComLayer connects to all FeeServers. After the connection is established, the InterComLayer subscribes to the services of the FeeServers and controls their message channels. Filtering of messages according to the log-level is performed on each layer to reduce network traffic. The service channels of the FeeServers are pooled together and re-published to the upper layer. By this means the source of the services is transparent to the SCADA system on top. The InterComLayer is an abstraction layer, which disconnects Supervisory and Field Layer.

In order to transport configuration data to the Front-end electronics, the InterComLayer has an interface to the configuration database. Neither database nor InterComLayer know about the format of the data. The data will be handled as *BLOBs* (Binary Large OBjects).

In addition, the InterComLayer provides functionality for maintenance and control of the FeeServers. Servers can be updated, restarted and their controlling properties can be adjusted to any requirements.

IV. RCU Board Radiation Tolerance

The functionality of both the DCS board and the RCU motherboard is based on FPGAs which can experience errors due to the radiation environment. These errors are not of a permanent nature, and can be corrected by reloading the firmware into the configuration memory of the FPGA. The firmware files are stored in radiation tolerant Flash memory on the different boards. However, reloading the configuration and rebooting

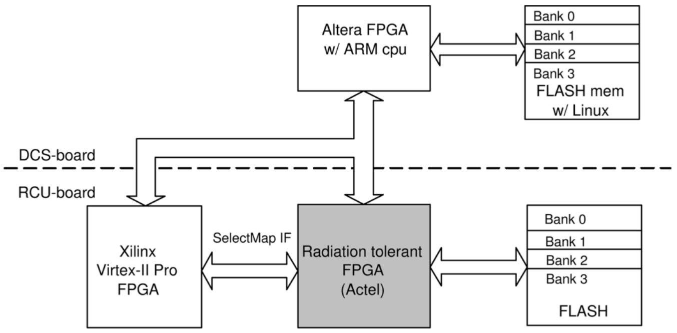

Fig. 5. Reconfiguration scheme for the Xilinx FPGA on the RCU board.

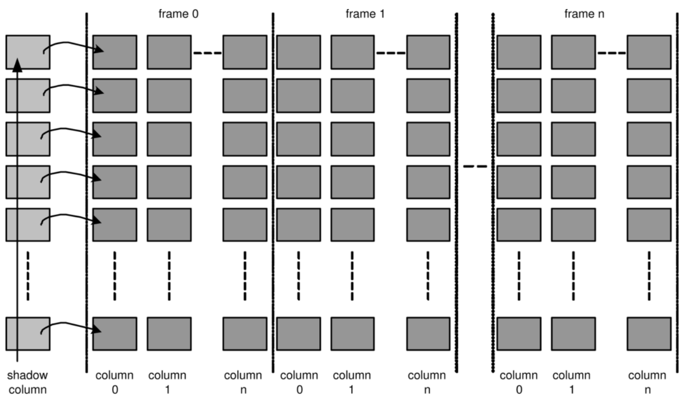

Fig. 6. Internal architecture of the Xilinx Virtex-II Pro FPGA.

causes downtime of the specific node. Since the data-path is independent of the status of the DCS board, occasional downtime of the DCS board node is irrelevant.

This simple approach is not satisfying for the FPGA on the RCU board, as it will interrupt the data-flow. To overcome this problem, automatic checking and refreshing of the firmware has been implemented. The scheme is sketched in Fig. 5. It is based on an FPGA which allows to refresh the firmware without interrupting the operation. This technique is called *Active Partial Reconfiguration*. In the current system a Xilinx Virtex-II Pro device [9] has been chosen. The architecture of the Virtex-II is divided into elements called frames, that can consist of either RAM elements, Configurable Logic Blocks (CLBs), IOs and such. These frames are again divided into smaller elements called columns, which are the atomic elements of the FPGA (see Fig. 6). When doing Active Partial Reconfiguration, one addresses these frames and columns by using frame-address and column-address, and it is in this way possible to refresh only one single column. When refreshing the configuration memory of the addressed column, the configuration data stream is first transmitted to a shadow column. When the data in shadow column is ready, all registers in the addressed column are updated in parallel.

On the RCU board, configuration and reconfiguration is done using the *SelectMap* bus interface, which is an 8 bit parallel bus. A complete configuration of a Xilinx Virtex-II VP7 will take approximately 40 ms. To control the SelectMap bus and communicate with the radiation tolerant Flash Memory, a CPLD (Complex Programmable Logic Device) is used. This is also radiation

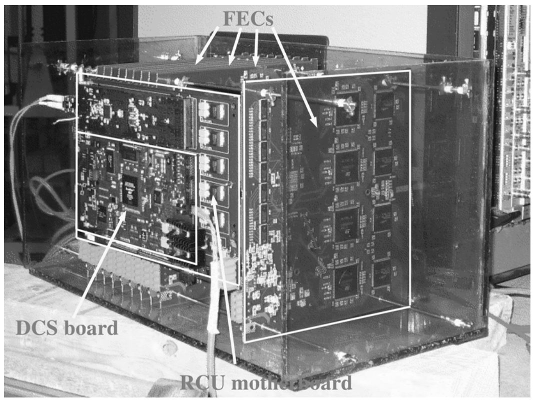

Fig. 7. The setup for the irradiation tests at TSL.

tolerant, and its custom-made firmware will make it possible to continuously refresh the configuration memory of the FPGA, or if needed, read back the configuration memory and verify it against the original binary-files stored in the Flash memory. When using the latter solution, it is also possible to interrupt the read process to correct an error in a column by refreshing the complete column. When the error has been removed, the read process will continue. This setup makes it possible to remove errors in configuration memory of the FPGA before they affect the behavior of the system.

The firmware in the CPLD also makes both the Flash memory and the Configuration Memory of the FPGA available for the DCS board computer, making firmware upgrade possible, or to do more sophisticated error-handling.

V. Integration Tests

Several small scale control systems consisting of the DCS board and applicable readout boards have already been used as stand alone systems in major tests, these include setups used for the TPC detector, the PHOS detector, as well as the TRD detector. The modularity of the design makes it possible to test the complete readout chain of data, only using a subset of the complete Control System. Two tests will be pointed out here because of there importance in the development.

A. TPC Fee Irradiation Test

During irradiation tests of the TPC Front-end electronics at the The Svedberg Laboratory (TSL) in Uppsala/Sweden in May 2005, the control system has been tested extensively. The setup is shown in Fig. 7. It consisted of an RCU motherboard, a DCS board and nine Front-End Cards attached to the RCU. A second setup used one RCU board equipped with a DCS board running without any FEC attached. This setup was mainly used to test active reconfiguration schemes and the interference between different tasks of the control system.

The FeeServer was running on the DCS board publishing 10 data points per FEC and a few memory locations in the RCU memory. Furthermore, a PC was running the InterComLayer. This integration test did not involve the SCADA system in the Supervisory Layer. All steering of the setup has been done via command line interfaces. The InterComLayer subscribed to the services provided by the FeeServer and wrote the values into files. In addition, all log messages from the FeeServer were collected, filtered and written to a file. In addition, the Active Reconfiguration functionality was tested.

A few problems occured in the startup phase including problems with the DIM framework which caused the FeeServer to crash under certain circumstances. These has been identified as timing problems and the software has been adapted to avoid them. After solving the problems, the control software ran stable during three days until the end of the irradiation test. The integration test has been continued afterwards and no major problem could be observed. There is some work necessary to avoid interference between the FeeServer and the Active Reconfiguration.

The analysis of the irradiation test is ongoing. Preliminary results show that the error rate is within the expected limits.

B. Test of the Communication Software

A major test of the communication software has been carried out during the TRD beam-test in October 2004. A stack of 6 TRD layers was mounted and placed into the beam line in an experiment hall at CERN. The test was mainly intended to check detector properties and the data readout chain, but served also as a good possibility to test the communication software on a larger scale.

Each of the six TRD layers contained one readout board with an attached DCS board. In addition, the top most and lowest readout boards were connected to a second DCS board respectively for testing purposes. One more was used to control the whole readout-chain. In total, the control system consisted of nine DCS boards and two PCs running the InterComLayer and the PVSS, respectively.

The control system ran continuously during the two weeks of the test-beam. The FeeServer on each DCS board computer monitored up to seven independent values, which were read from an ADC placed on the DCS boards. All these values were collected by the InterComLayer and delivered further up to PVSS. Control and configuring commands were sent as broadcasts from this SCADA system as well. The actual control data was read from small sized configuration files by the InterComLayer and distributed to the FeeServers on the DCS boards.

VI. Summary and Conclusion

The Detector Control System for the TPC Front-end electronics of the ALICE experiment has been presented. Each of the 36 readout sub-systems is based on custom-made hardware devices. Already the low-level devices provide intelligent error-handling, which ensures a very short latency. This is of high importance to decrease the possibility for permanent failures.

The system is designed to be independent of physical intervention. Software and firmware are easily reconfigurable. Together with the distributed and module-based design with well-defined interfaces, this increases the flexibility and testability of the system.

The DCS board embedded computer is a fundamental part in the distributed system which allows running complex controlling software under the operating system Linux. Tasks can be processed in parallel on the DCS board and on the connected custom hardware devices. Furthermore, the Linux operating system on the embedded computers provides flexibility and well known interfaces.

The system is still under development. The modularity makes it possible to test and review each sub-system on its own, independently of the complete setup, and several tests have been performed with satisfying results.

References

[1] ALICE—Technical Proposal For A Large Ion Collider Experiment at the CERN LHC ALICE Collaboration, CERN/LHCC 1995-71, 1995.

[2] Technical Design Report: Trigger, DAQ, HLT, DCS ALICE Collaboration, CERN/LHCC/2003-062.

[3] R. Esteve Bosch *et al.*, "The ALTRO chip: a 16-channel A/D converter and digital processor for gas detectors," *IEEE Trans. Nucl. Sci.*, vol. 50, no. 6, pp. 2460–2469, Dec. 2003.

[4] L. Musa *et al.*, "The ALICE TPC front end electronics," *Proc. IEEE Nuclear Science Symp.*, pp. 3647–3651, 2003.

[5] C. González Gutiérrez *et al.*, "The ALICE TPC readout control unit," in *Proc. 10th Workshop on Electronics for LHC and Future Experiments*, Boston, MA, 2004.

[6] H. Tilsner *et al.*, "Hardware for the Detector Control System of the ALICE TRD," in *Proc. 9th Workshop on Electronics for LHC Experiments*, Amsterdam, The Netherlands, 2003.

[7] C. Gaspar *et al.*, "DIM, a Portable, Light Weight Package for Information Publishing, Data Transfer and Inter-process Communication," presented at the Int. Conf. Computing in High Energy and Nuclear Physics, Padova, Italy, 2000.

[8] S. Bablok *et al.*, "Front-End-Electronics Communication software for multiple detectors in the ALICE experiment," *Nucl. Instrum. Methods*, vol. 557, no. 2, pp. 631–638, Feb. 2006.

[9] Virtex-II Pro and Virtex-II Pro X User Guide, Xilinx UG012 (v3.0) Xilinx Corp., Aug. 2004.