

# First experiences with the ATLAS pixel detector control system at the combined test beam 2004

Martin Imhäuser\*, Karl-Heinz Becks, Tobias Henß, Susanne Kersten, Peter Mättig, Joachim Schultes

*University of Wuppertal, Gaußstraße 20, 42119 Wuppertal, Germany*

Available online 12 June 2006

## Abstract

Detector control systems (DCS) include the readout, control and supervision of hardware devices as well as the monitoring of external systems like cooling system and the processing of control data. The implementation of such a system in the final experiment also has to provide the communication with the trigger and data acquisition system (TDAQ). In addition, conditions data which describe the status of the pixel detector modules and their environment must be logged and stored in a common LHC wide database system.

At the combined test beam all ATLAS subdetectors were operated together for the first time over a longer period. To ensure the functionality of the pixel detector, a control system was set up.

We describe the architecture chosen for the pixel DCS, the interfaces to hardware devices, the interfaces to the users and the performance of our system. The embedding of the DCS in the common infrastructure of the combined test beam and also its communication with surrounding systems will be discussed in some detail.

© 2006 Elsevier B.V. All rights reserved.

*Keywords:* ATLAS; Pixel; Detector control system; Slow control; SCADA; Communication; Conditions database

## 1. Introduction

For the first time, segments of all ATLAS subdetectors were integrated and operated together with a common Trigger and Data Acquisition (TDAQ), close to final electronics and the Detector Control System (DCS) at a CERN test beam.

During this test and certainly in the future experiment the overall aim of the DCS was and is to guarantee a reliable physics data taking and a safe operation of the detector. This is done by monitoring and controlling the DCS hardware, reacting to error conditions, providing several user interfaces and maintaining the communication to common infrastructure of the ATLAS experiment like TDAQ or database systems. Especially, the communication between TDAQ and DCS is of major importance for the operation of the pixel detector as tuning of the readout chain requires access to both systems in parallel.

## 2. Experimental set-up

The pixel detector module, shown in Fig. 1, is the smallest unit pixel that DCS can act on. It consists of a silicon sensor and 16 frontend chips as well as a Module Controller Chip (MCC) gathering hit data and servicing trigger requests.

Every detector module is connected to an optoboard which converts the electrical data signals transmitted from the modules to an optical signal for transmission to the off-detector electronics via optical fibres. At the same time it receives optical signals from the electronics and converts these to electrical signals for distribution to the modules. The off-detector component Back Of Crate card (BOC) which serves as the optical interface between the Read Out Driver (ROD) and the optoboard [1] is controlled by TDAQ while DCS takes care of the on-detector component optoboard.

To operate six pixel detector modules as a part of the whole pixel detector its DCS provided various equipment at the combined test beam (shown in Fig. 2). For more

\*Corresponding author.

*E-mail address:* [imhaeuse@physik.uni-wuppertal.de](mailto:imhaeuse@physik.uni-wuppertal.de) (M. Imhäuser).

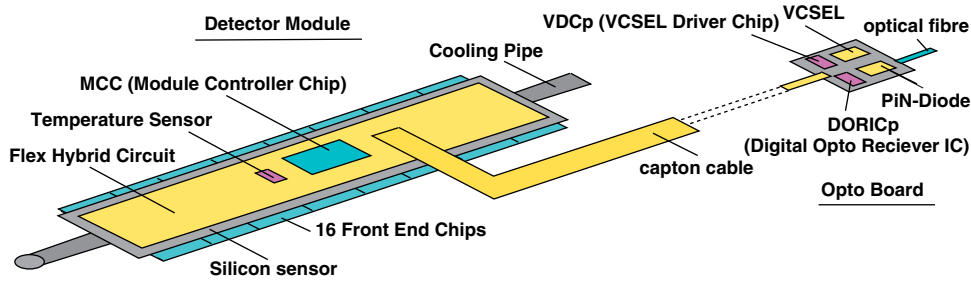


Fig. 1. Pixel detector module.

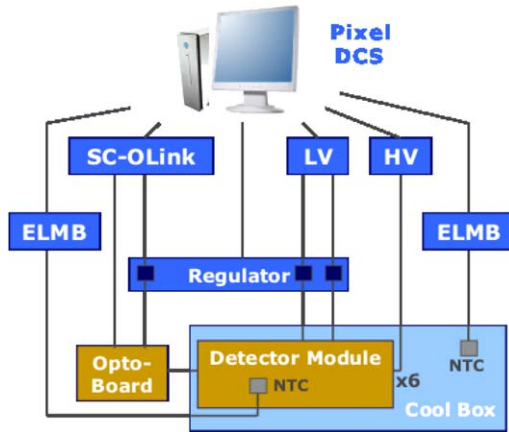


Fig. 2. Pixel DCS test beam set-up.

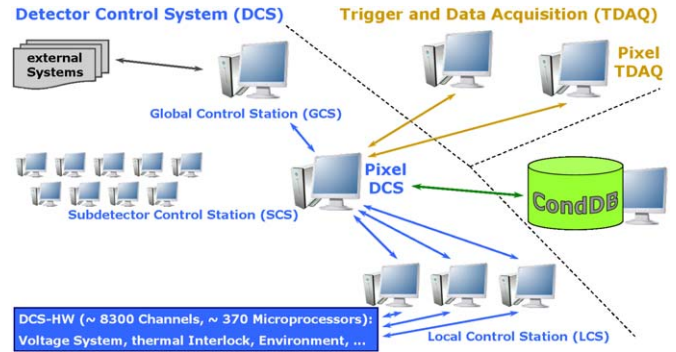


Fig. 3. Detector control system.

details about the design of the pixel DCS, please refer Ref. [2].

The following hardware have come into operation:

- general purpose IO devices (ELMB<sup>1</sup>) for the readout of the DCS hardware developed by the ATLAS DCS group,
- a proprietary development supply system of three low-voltage sources together with a reset signal to operate the optoboard (SC-OLink<sup>2</sup>),
- a regulator system for protecting the FE chips of the detector modules, developed by INFN Milano, together with a high-voltage source for the depletion of the sensors and
- and also temperature and humidity sensor.

To integrate the hardware and to supervise the system, in-house developed software tools have been used.

### 3. Detector control system

The ATLAS detector is hierarchically organised in a tree-like structure into sub-detectors, sub-systems, etc. This has to be reflected in the design and implementation of the DCS (Fig. 3).

<sup>1</sup>Embedded Local Monitor Board.

<sup>2</sup>Supply and Control for the OptoLink.

Therefore, DCS is organised into three functional layers:

- the *global control station* which e.g. provides tools for the overall operation of ATLAS,
- the *subdetector control station* which e.g. provides full stand-alone control capability and synchronises the supervision of all subsystems below, and
- the *local control station* which e.g. reads data from the DCS hardware.

The core of the software is based on the commercial Supervisory Control And Data Acquisition (SCADA) package PVSS.<sup>3</sup> PVSS allows to gather information from the DCS hardware and offers the implementation of supervisory control functions such as data processing, alert handling and trending. It has a modular architecture based on functional units called managers. Applications can be distributed over many stations on the network which defines a distributed system [3].

#### 3.1. Distributed system

At the combined test beam we embedded three PVSS stations as a distributed system based on a full detector simulation as shown in Fig. 4. This test demonstrated successfully the partitioning over several computers and their interconnection in a common environment.

<sup>3</sup>Prozeß-Visualisierungs und Steuerungs- Software, ETM, Austria.

### 3.2. Software tools

The software of the pixel DCS consists of several subprojects such as tools for the implementation of the DCS hardware in the software environment and the configuration in an automated way, tools for combining all information concerning one detector module in a flexible way (see 8) and also graphical user interfaces. For example Fig. 8 shows the System Integration Tool (SIT) which follows the detector hierarchy and therefore maps the real cabling structure into the software.

All these software tools were used at the combined test beam and the experience now helps to develop advanced tools for the experiment.

## 4. DAQ–DCS communication

TDAQ and DCS are controlled by finite state machines which consist of different states and transition functions which map a start state to a next state. Both systems are independent while TDAQ has the master control during detector operation. This means that the TDAQ finite state machine has to be able to cause transitions in the DCS finite state machine. Furthermore, TDAQ applications have to transfer selective data to DCS as well as DCS must make required data available to TDAQ. Nevertheless, TDAQ must be informed about state conditions. To cover all the required transfers, the DAQ–DCS Communication (DDC) software [4] has been developed by the ATLAS DCS group (see Fig. 5).

DDC was set up in pixel configuration by the authors and was running for four months in the combined environment. During this time the pixel-specific DDC application was tested intensely.

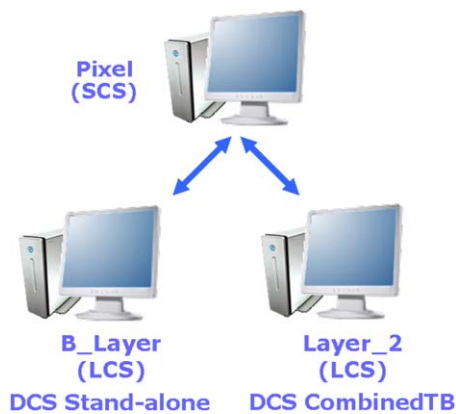


Fig. 4. Distributed system.



Fig. 5. Schematic of DDC.

Concerning the command transfer, we were able to show that the used pixel DCS finite state machine reacted in a well-defined way on TDAQ transitions. Additionally, pixel DCS directly computed actions via DDC in response to three TDAQ transitions at the combined test beam. Furthermore, the possibility to set DCS hardware with TDAQ applications without changing the TDAQ state was tested successfully.

Regarding the data transfer, DCS visualised data-like temperatures, depletion and low voltages of the detector modules or the states of DCS hardware for TDAQ, while DCS received data from TDAQ, like the status of TDAQ services or run parameters. Especially, the run number was used for storing run relevant DCS data. In combination with a shown dynamical integration of more transfer data this was done very efficiently at the combined test beam.

For the message transfer we built up a DCS finite state machine to monitor the parameters of the detector modules and to generate corresponding states. Pixel DCS sent messages with severity flags which were read by the shifter during data taking.

Performing timing studies, certain DCS actions were connected to TDAQ transitions. For this the interconnection between on- and off-detector parts of the optical link is of special interest. Thus, the setting of the reset signal of the SC-OLink which allows a slow and controlled recovery of the delay control circuit was linked to the transition 'LOAD' (see Fig. 6).

The total time for a transition is composed of the time of the DDC process and the time of the DCS process. For the above example, we measured 50ms for the DDC and around 5s for the DCS process. Due to these measurements we were able to optimise the control code during the test beam period together with changes in the hardware properties.

To verify the full functionality of DDC by the shifter during the experiment, additional tools for data analysing (see Fig. 9) are inserted in the structure of the DCS system which did not affect the normal operation. Checking the command transfer is done by switching and setting any number of virtual power supplies while checking the message transfer is done by simulating differently weighted temperatures of a detector module and sending corresponding messages with severity flags. Reviewing the data transfer, one could observe from the TDAQ side a simulated high-voltage scan of a virtual detector module inside DCS. On the other hand, simulated TDAQ data are visible in DCS.

These tools were very helpful during the operation and they are now an inherent part of the DCS.

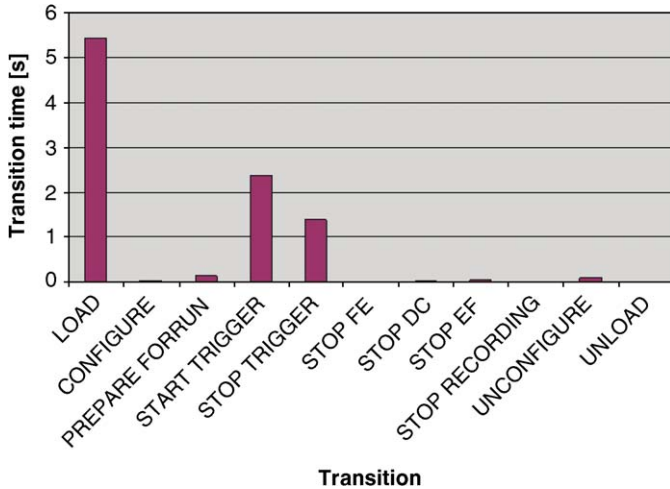


Fig. 6. Required time for various transitions.

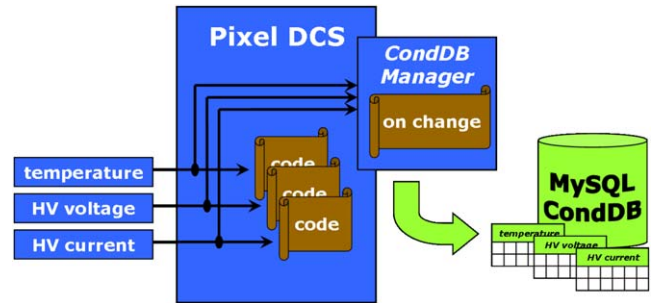


Fig. 7. Schematic of data flow through the interface.

Based on about 150 Bytes per storage for a detector module, pixel DCS would produce more than 37 GBytes of data for physics analysis per year at an estimated 5 min storage interval.

## 6. Summary

At the combined test beam we have built up a pixel detector control system which worked very well during the four month beam period. Pixel-specific software tools were used with good acceptance by shifters. Many functionality issues could be studied sufficiently.

The DAQ–DCS communication software was tested intensely and was established very successfully in the pixel configuration. We were able to use the full functionality of DDC (Figs. 8 and 9). We provided commands for several actions inside DCS. TDAQ data were computed by DCS in a well-defined way while DCS data was used by TDAQ for monitoring. Messages with severity flags were available. From this point all further requirements to pixel DCS coming with a system scale-up could be achieved by this package.

DDC is the appropriate tool to handle the interaction between on- and off-detector parts of our optical link. It allows us to develop tuning algorithms to find the optimal operation point for the components of the readout chain. As a first step, a graphical user interface which shows inside DCS various parameters of the BOC is currently under development.

The used interface to the conditions database did not cover all the pixel DCS aims. After the combined test beam, ATLAS intended to use the LHC Computing Grid (LCG) framework for developing a new interface to the conditions database which makes available general database tools and interfaces for subsequent analysis. Further better configurability and more flexibility for filtering data as well as the possibility to read data from the conditions database in PVSS has to be considered.

## Acknowledgements

These studies are a result of sharing knowledge in the ATLAS pixel DAQ group and the ATLAS DCS group. We would like to thank all people being involved in the work,

## 5. Interface to the conditions database

Conditions data are seen as every data needed for reconstruction besides the event data itself, thus it has to reflect the conditions the experiment was performed and the actual physics data were taken [5]. For the pixel DCS this includes basically parameters of the detector modules such as voltages, currents and temperatures but also parameters and status information of further DCS devices.

As already mentioned, the ATLAS DCS is based on the software PVSS. PVSS contains an internal archive to store and to read back the history of DCS values, but does not allow to access the data from outside the PVSS framework.

Therefore, a PVSS API<sup>4</sup> manager was developed by the Lisbon TDAQ group. This custom-made manager is based on a C++ interface between PVSS and a MySQL database. When running, it connects to each of the DCS parameters defined by the user and stores the values together with a timestamp in the database. When a value change occurs, the previous value is updated by replacing the timestamp by a time interval and the new value is stored in the database in the same way as the first value (Fig. 7).

During the combined test beam the system reliability of the pixel DCS set-up, the amount of data and the handling of the interface have been examined.

Due to the given storing mechanism described above, no data filter or smoothing processes could be used. As one result we had about five storing processes per second and per value which produced a nonacceptable amount of data and necessitated a limitation of changes. By the integration of a new storing mechanism with a storage at the begin of a run, every minute<sup>5</sup> and at the end of the run we were able to reduce the amount of data significantly.

<sup>4</sup>Application Programming Interface.

<sup>5</sup>If a monitored value ran out of its limits this storage interval was scaled down to get more information about the behaviour

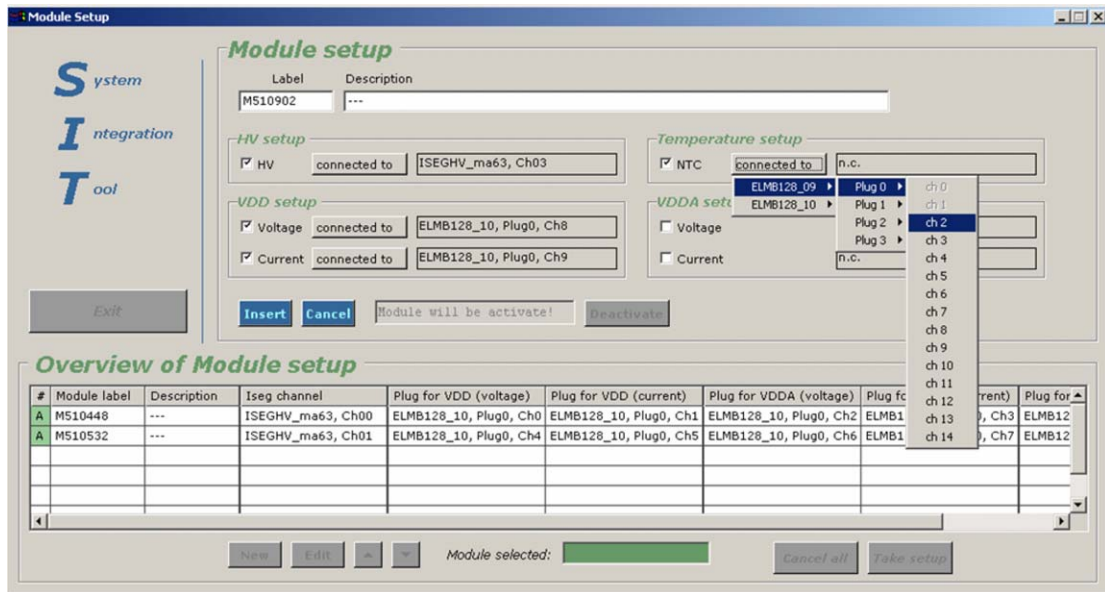


Fig. 8. Graphical user interface—SIT.

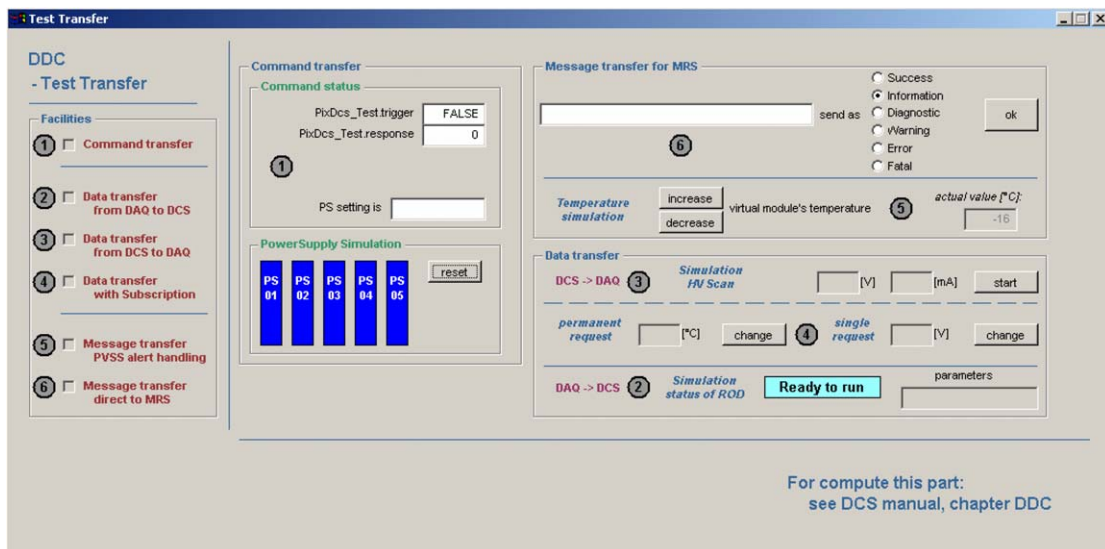


Fig. 9. Graphical user interface—DDC.

especially V. Khomoutnikov for support during the test beam period. He was always open for discussions and gave us a lot of fruitful hints.

## References

- [1] T. Flick, et al., Optical read out in a multi-module system test for the ATLAS pixel detector, Proceedings of the Pixel 2005, Bonn, September 2005.
- [2] M. Imhäuser, et al., IEEE Trans. Nucl. Sci. NS-51 (2004) 502.
- [3] H.J. Burckhart, et al., Vertical slice of the ATLAS detector control system, Seventh Workshop on Electronics for LHC Experiments, Stockholm, Sweden, September 2001.
- [4] V. Khomoutnikov, ATLAS DAQ–DCS communication software, User's Guide, Draft 4.1, CERN, March 2004.
- [5] A. Amorim, et al., Experience with the open source based implementation for ATLAS Conditions Data Management System, CHEP03, La Jolla, CA, USA, March 2003.