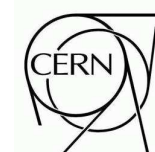# ATLAS INTERNAL NOTE

ATL-TILE-PUB-2007-XXX

November 29, 2007

TileCal TDAQ/DCS communication

C. Solans[1], A. Abdallah[1], G. Arabidze[2], B. Carneiro Ferreira[3], B. Sotto-Maior Peralva[4]

[1] *Departamento de Fisica Atomica, Molecular y Nuclear and IFIC, Spain*
[2] *University of Athens, Greece*
[3] *Universidade Federal do Rio de Janeiro (UFRJ), Brasil*
[4] *COPPE Escola de Engenharia (UFRJ), Brasil*

Abstract

This document describes the communication between the TDAQ and DCS systems of the Hadronic Tile Calorimeter detector of the ATLAS experiment, currently under commissioning phase at CERN. The aim of the implementation is to increase the robustness and understanding of the detector from the two systems involved. By design, the two systems operate in parallel, independently of each other. Hence, correlation between the states of the two systems is required in order to understand the operational state of the detector as required for further analysis of the archived data.

Contents

List of Figures

List of Tables

1 Introduction

The Hadronic Tile Calorimeter (TileCal) is one of the sub-detectors of the ATLAS experiment. It is divided into four barrels, two long and two extended barrels. Each of these, which define a Timing, Trigger and Control (TTC) partition, is azimutally sub-divided into 64 wedges, called super-modules. These super-modules house the so-called super-drawers that contain the front-end digitizer and read-out electronics.

The TileCal detector uses the Trigger and Data AcQuisition (TDAQ) [1] system for the control of the data-flow, monitoring and trigger of the detector, and Detector Control System (DCS) [2] for the control of the detector and common infrastructure of the experiment (cooling, ventilation, electricity, safety). The TDAQ software is implemented to run under Linux while the TileCal DCS software is implemented to run under Windows.

The exchange of information between TDAQ and DCS is based on the Distributed Information Management (DIM) system [3], a CERN development for the Delphi online software. DIM is based on the client/server paradigm and is implemented for both Linux and Windows. TDAQ applications act as a client to read/write services provided by DCS applications that act as servers.

The communication protocol is common to all TileCal sub-systems which take part in the data acquisition and can operate the front-end electronics independently. e.g. Cesium and Charge Injection System.

Section 2 describes the granularity of the information. Section 3 describes the data format of the information exported by the TDAQ and the DCS. Section 4 overviews the DCS reaction to the information reported by the DAQ on the operation of the super-drawers. Finally, Section 5 describes the operation of the TDAQ on the information reported by DCS.

2 Granularity

Each of the four TTC partitions define the TDAQ operation granularity of the front-end super-drawers. The minimum control of the front-end is done in groups defined by TTC partition. Therefore, the number of information elements that need to be sent from the TDAQ to the DCS is one per TTC partition.

From the DCS point of view, each super-drawer is an independent element that can be operated individually. Therefore, the number of information elements that need to be sent from the DCS to the TDAQ is 64 per TTC partition.

Figure 1 is a diagram of the TDAQ and DCS data exchange.
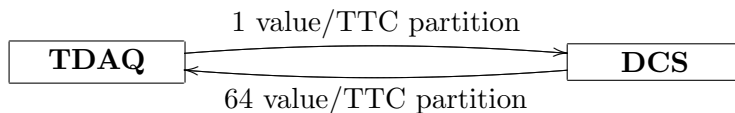

Figure 1: TDAQ and DCS data exchange.

3 Data format

The information exported from TDAQ to DCS, is based on the Run Control state [4] encoded into a single integer value. Table 1 lists the TDAQ operation IDs associated to each Run Control state.

State	ID
Error	1
None	2
Initial	3
Config	4
Connect	5
Ready	6
Running	7
Paused	8
ROISTOPPED	9
LVSVSTOPPED	10
EBSTOPPED	11
EFSTOPPED	12

Table 1: TDAQ status.

The DCS exports a word per super-drawer that contains information about it's status. This word is a summary of all the super-drawer components encoded into an integer as defined by Eq. 1, where the HV(PMT) values are listed in Table 2, the HV_CAN with only two values are listed in Table 3 and the Low Voltage Power Supply (LVPS) values, MotherBoard (MB) (LVPS) and HV(LVPS), are listed in Table 4.

$$HV(PMT) \times 10000 + HV\_CAN \times 1000 + MB(LVPS) \times 100 + HV(LVPS) \tag{1}$$

State	ID
Off	0
Not all quarters On	1
All On	2

Table 2: DCS HV status.

State	ID
No communication	0
Ok	1

Table 3: DCS HV communication status.

State	ID
Unknown	20
Output voltage is Off	21
Output voltage is On	22
Output voltage is at minimum	23
Alarm	30

Table 4: DCS LVPS status.

4 Detector Control System

The Detector Control System (DCS) [5] is responsible for the safe operation of the detector. All actions initiated by the operator and all errors, warnings and alarms concerning the hardware of the detector are handled by the DCS.

The TileCal DCS architecture consists of a distributed Back-End (BE) system running on PCs and different Front-End (FE) systems. It acquires data from the FE, offers supervisory control functions and handling of commands, messages and alarms. In order to provide such functionality, the BE system of TileCal is composed of five machines, four Local Control Station (LCS) and one Sub-detector Control Station (SCS). This hierarchy allows the sub-detector to be divided into independent partitions which have the ability to operate in stand-alone or integrated mode.

4.1 High Voltage System

The TileCal High Voltage (HV) System is based on HV bulk power supplies which supply all PMT voltages inside the super-drawers with a common input voltage of 830 V. In order to have the desired behaviour of the signals created from the collision and due the intrinsic characteristics of the PMTs, a HV regulator system was implemented to provide a certain voltage for each individual channel (PMT). By design, the super-drawer is divided into four quarters, each of which is operated independently by the HV regulator system.

4.2 Low Voltage System

In order to supply all TileCal front-end electronics inside the super-drawer, a dedicated system was developed. The main component is the Low Voltage Power Supply (LVPS) [6] which is located at the end of each girder. The LVPS recieves 200 V from a bulk power supply as an input voltage and provides in total eight different output voltages. Three of them (HV +15, HV -15, HV +5) supply the HV distributor and regulator cards. The others (MB -5, MB +5, MB +15, DIG +5, DIG 3.3) feed the read-out electronics. If any of the output voltages has the minimum value, the status of the LVPS is trimmed to the minimum voltage.

4.3 DIM services

DCS provides 64 DIM services, integer format, per TTC partition based on the convention given by Eq. 1. These services follow the naming convention XXX_LVPSYY, where XXX stands for the TTC partition name (LBA, LBC, EBA, EBC) and YY is the number of the super-drawer. For example: the service LBA_LVPS32 published by DCS with value 212222, means that all the PMT voltages are ON on this super-drawer (LBA32), CAN communication is OK, MB(LVPS) output voltages (MB bricks) are ON and the HV output (HV bricks) are ON.

The DCS also subscribes to services, integer format, published by the TDAQ, which follow the naming convention XXX_DAQ where, as in the previous convention, XXX stands for the TTC partition name. This information contains the status of the TDAQ activities which is stored in the Oracle Archive.

5 TDAQ

Several TDAQ operations are performed according to the DCS information. The read-out is configured according to the information retrieved from the Information Server (IS) [7] services,

which is provided by DCS. A TileCal IGUI DDC panel is available for each partition to inform about the DCS state of each super-drawer. Futhermore, the list of available super-drawers at the beginning of the run,extracted from the DCS are included into the metadata [8] of the raw data file.

5.1 IS information

The IS information objects used to describe the super-drawer information from DCS and from the ROD front-end link status is the TileCal_IS_Drawer. The schema for these objects is described in the TileConfiguration package of the TileCal online software repository [9] and C++ and Java interfaces are available. Table 5 details the attributes of these objects.

Attribute	Value
State	DCS information word
LinkUp	ROD link up boolean status
ReadOut	ROD read-out boolean status

Table 5: TileCal_IS_Drawer IS information objects.

5.2 IGUI panel

The TileCal IGUI panel includes a DDC information panel for each TTC partition which displays all information stored in the IS objects through a color code detailed in table 6. The TileCal IGUI DDC panel is presented in Figure 2.

MB LVPS	HV LVPS	HV	Status	Color
UNKNOWN			Unknown	Grey
OFF			Off	Blue
RAMP_UP			Ramp_up	Yellow
ON	!ON	!ON	ON	Lightgreen
ON	ON	ON	HVLV ON	Green
ALARM			ALARM	Red

Table 6: TileCal IGUI DDC panel status color convention.

5.3 Metadata

A sub-set of the IS objects populated by the DCS go to raw data file as metadata. This allows the offline analysis to know upon a missing ROD fragment or if there was a problem either in the read-out or a super-drawer was never selected to be included in the run.

Bits	31 ... 24	23 ... 13	12 ... 0
Value	Reserved	Barrel ID	Super-drawer index [0 - 63]

Table 7: TileCal fragment ID word composition.

The proposal on the metadata fragment implementation foresees up to three different tags. The first one describes the list of super-drawer modules included into the read-out. The second one describes the list of powered modules. The third one is the list of modules included in the
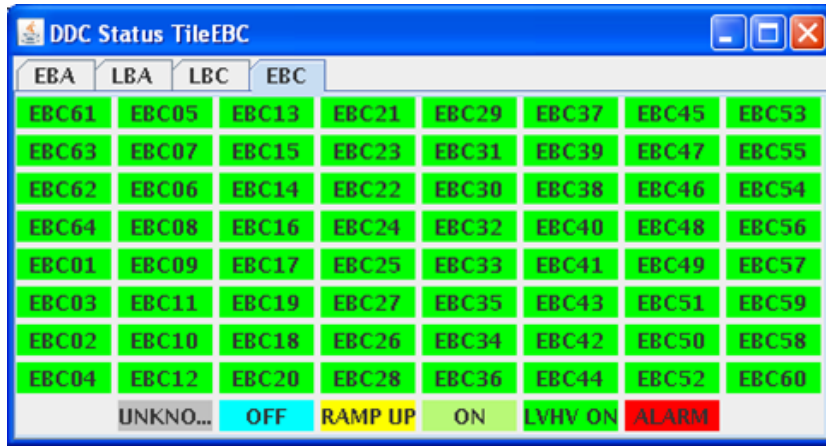
Figure 2: TileCal IGUI DDC panel.

read-out. Table 7 describes the fragment ID word composition. These tags are implemented as a list of fragment IDs as described in Table 8.

Attribute	Value
TilePoweredModules	List of fragment IDs of powered modules
TileLinkUpModules	List of fragment IDs of modules with link up status
TileReadoutModules	List of fragment IDs of modules included in the read-out

Table 8: Metadata strings.

An extra algorithm is needed to write the metadata into the raw data file. Once DDC IGUI is updated, an event is triggered and the user meta-data tags attribute of the run tag list object of the configuration database is updated. However, the values will not be populated into the raw data unless the UserMetaData object in the RunParams IS server is updated. Another event has to be triggered to update the UserMetaData object with the new IS information. Eventually, when the user clicks start, the information from the IS is retrieved from the event builder and is written as metadata to the raw data file.

5.4 Operation

At the beginning of a run, the shifter is able to configure the read-out according to the current DCS status. The shifter overviews the DDC panels and enables the appropriate channels for the run.

Futhermore, the metadata information will be used by the offline reconstruction algorithm to determine which super-drawers were included in the raw data file.

5.5 Integration

Two separate applications are used to handle the bidirectional TDAQ-DCS communication. Both applications depend on each TTC segment of the configuration database.

Tile DCS to DAQ is an infrastructure application that publishes to the IS the information concerning the super-drawers read-out from the DCS. This component is common to all partition dependant applications which want to retrieve the DCS status of the super-drawers.

Tile DAQ to DCS is an application that writes to the DCS the DAQ run control state on change and after an elapsed time. This application is started on *Boot* and stopped on *Shutdown*. This application is meant to send the status of the general data acquisition to the DCS.

At the level of the TileCal TDAQ configuration databases [9] [10], this application is attached to the TTC segment for every TTC partition. This is important if there is a set-up in which the front-end is not used and therefore the TTC segment for a given TTC partition is disabled.
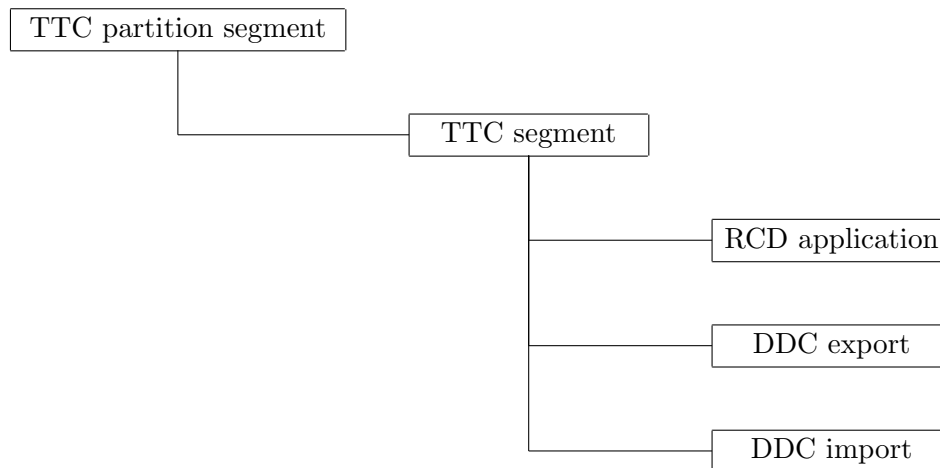

Figure 3: TileCal TDAQ configuration database DDC application layout.

To be able to use this write mechanism from all TileCal TDAQ applications, the TileDDC library is available in the TileDDC package. This library contains a subset of algorithms from the TDAQ DDC package compiled together to ease the interaction with DCS via DIM, which can be implemented into any binary application, including Detector Verification System (DVS) [11] test applications.

References

[1] The ATLAS Data Acquisition and High Level Trigger
http://twiki.cern.ch/twiki/bin/view/Atlas/HltDaqMainPage.

[2] C. Marques *et al.*, TileCal Detector Control System, 2006, Proceedings of ICAP, pp 118-124, Chamonix, France.

[3] C. Gaspar and M. Donszelmann, DIM a distributed information management system for the DELPHI experiment at CERN, 8th Conference on Real-time Computer Applications in Nuclear, Particle and Plasma Physics, 1993.

[4] G. Lehmann Miotto, Operations at different Activity Stages, Internal Report ATL-TDAQ-CONTROLS-2005-001, CERN, 2005.

[5] ATLAS Detector Control Systems
http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/DCS/dcshome.html.

[6] TileCal Low-Voltage Power Supply
http://atlas.web.cern.ch/Atlas/SUB_DETECTORS/TILE/elec/lvps.

[7] M. Caprini, P.Y. Duval, R. Jones, S. Kolos, Information service for Atlas DAQ Prototype -1.
http://atddoc.cern.ch/Atlas/Notes/031/Note031-1.html.

[8] D. Costanzo *et al.*, Metadata for ATLAS, Preprint, CERN, 2007, ATL-D-MN-0001.

[9] TileCal DAQ online software repository
http://twiki.cern.ch/twiki/TileDAQ.

[10] ATLAS TDAQ configuration databases
http://atlas-onlsw.web.cern.ch/Atlas-onlsw/components/config-db/Welcome.html.

[11] M. Barczyk *et al.*, Verification and Diagnostics Framework in ATLAS Trigger/DAQ, Internal Report ATL-DAQ-2003-033, CERN, Geneva, May 2003, DVS Web pages: http://atddoc.cern.ch/Atlas/DaqSoft/components/diagnostics/Welcome.html.