Involvement and structure @Bologna

S. Cecchini
INFN-Bo

KM3NeT Q Workshop
Amsterdam, Jan 30-31, 2014

People

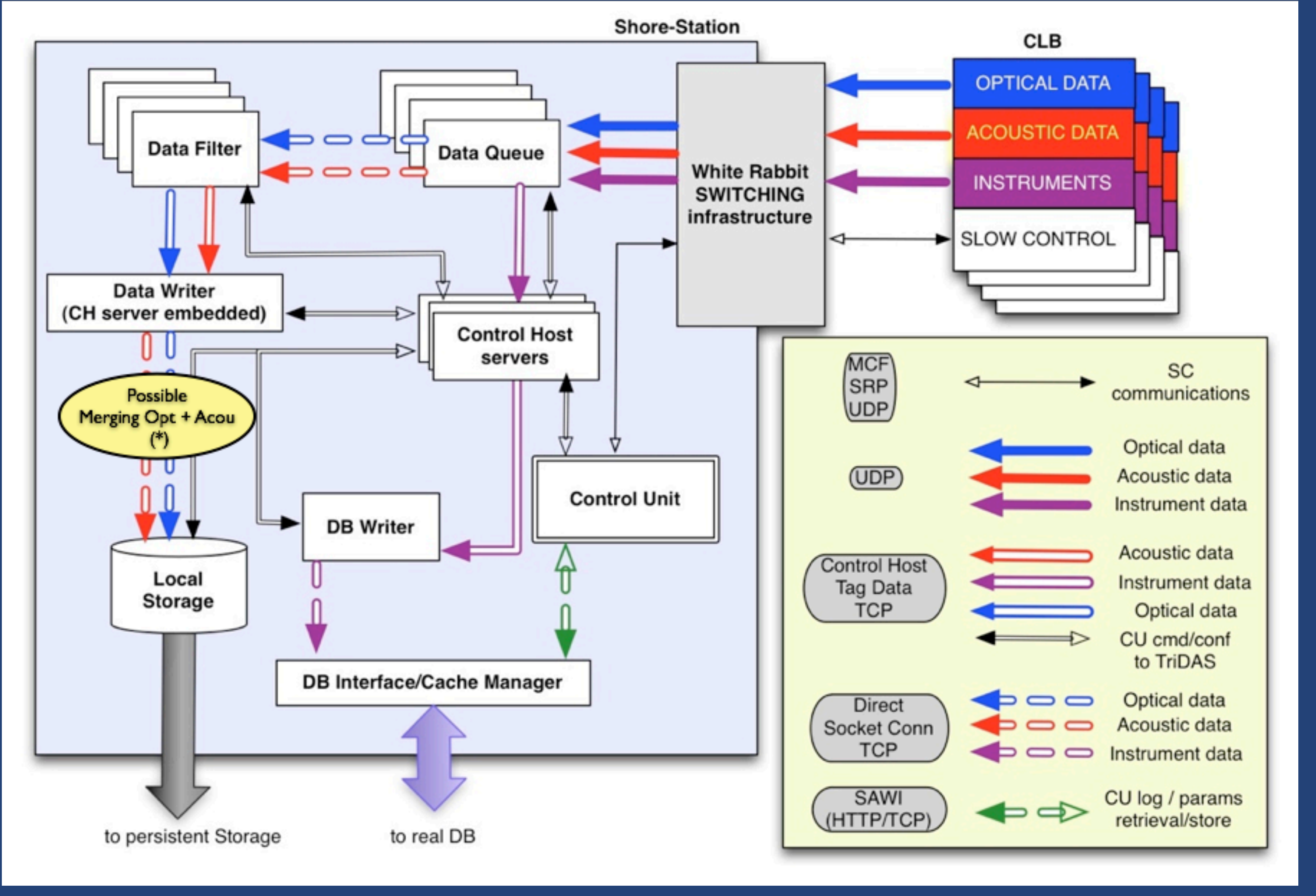
- 5 scientists
- 1 PhD
- 2 electronic supports
- 1 LQS – appointed Dec. 6th, 2013

Involvement

- DAQ implementation (Responsibility)
- PPM-DU (task =syncing & formatting)
- CLB (task = implementing test)

- Science (Topics, MC, Data analysis…)

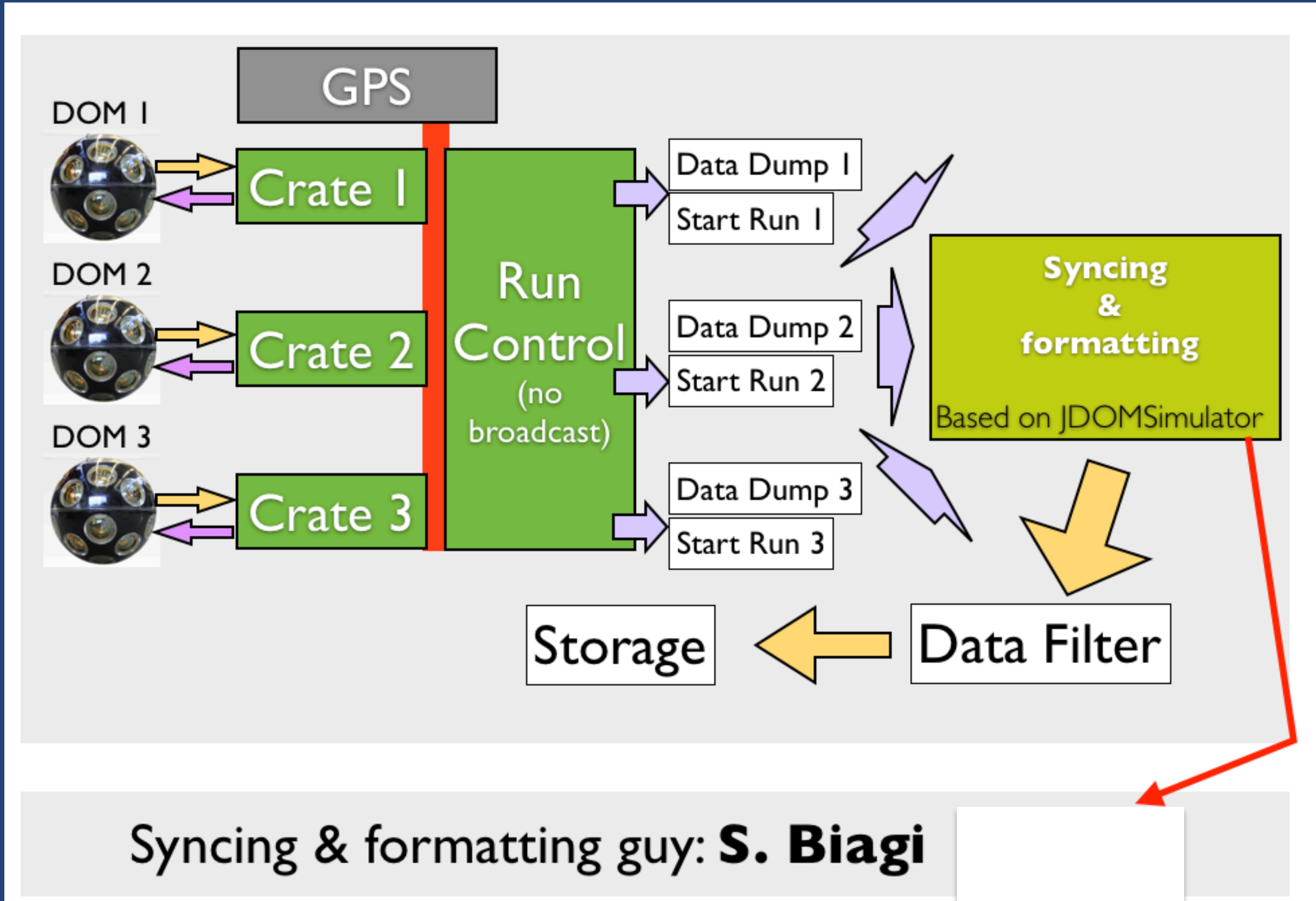
General DAQ design



(*) Compass and Tiltmeter data embedded in the written files

PPM-DU/CLB

3 “asynchronous” DAQs but merged quasi-online



Milestones @ Bo

- DAQ TDR - by the beginning of March 2014

- PRR – when/which ? (AG: + 1 month TDR) according to the last DAQ-WG plan

- PPM-DU quasi-on-line proc-soft by March 2014
- Useful tools for line integration by May 2014
- A theoretical global readiness by July 2014

- CLB Critical Design Review by Jan 30^{th} 2014

Facilities

Laboratory “Reserved area” for

- electronic card development and test
- test-bench applications

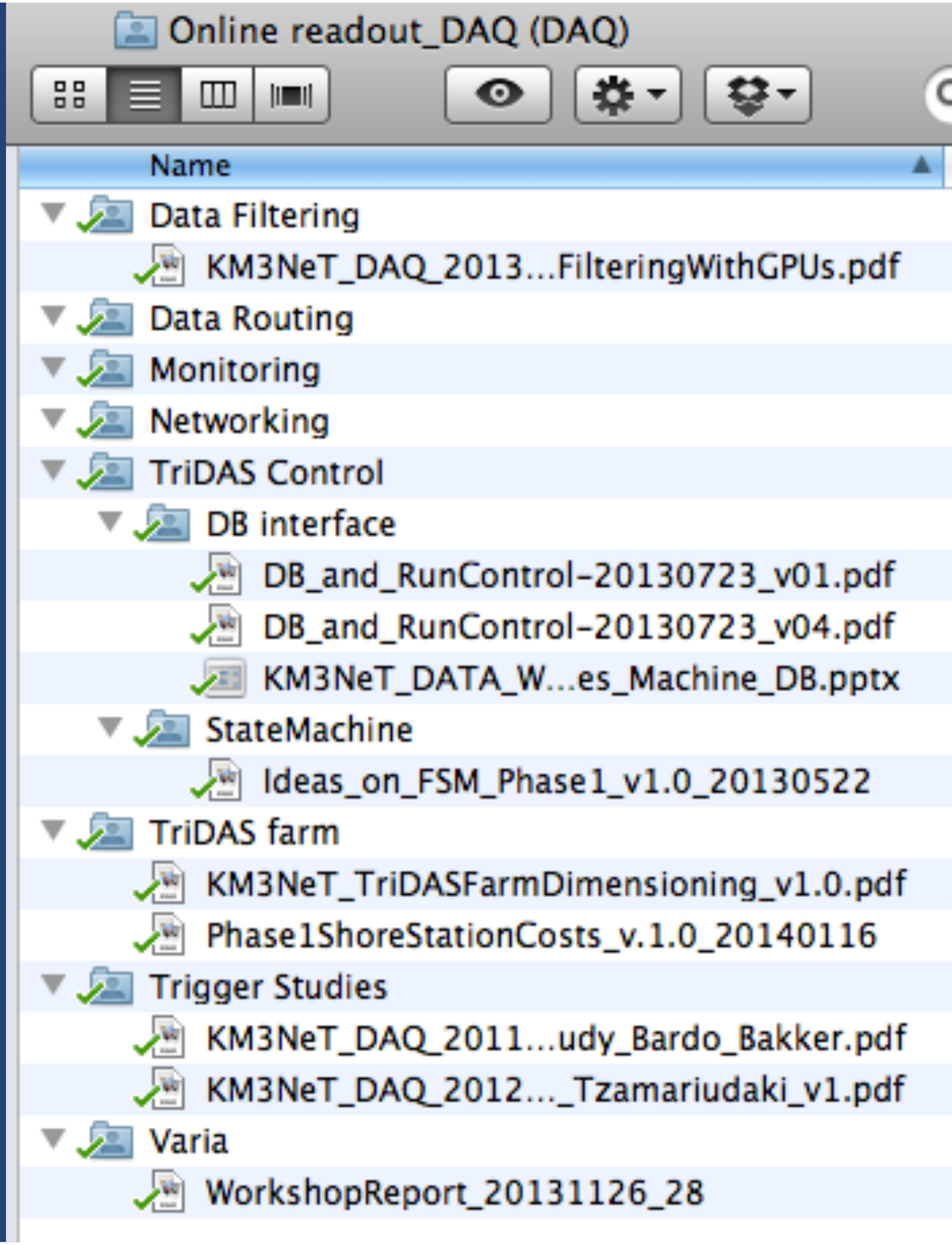
Inventory of Bologna produced docs so far

- in GoogleDrive reserved areas
- numerous entries in the ELOG[ELEC] = quick communications inside sub-groups

BUT

[in my opinion this tool is of low reliability for reconstructing and insufficient for recording final decisions]

DAQ


Online readout_DAQ (DAQ)
Name
Data Filtering
KM3NeT_DAQ_2013...FilteringWithGPUs.pdf
Data Routing
Monitoring
Networking
TriDAS Control
DB interface
DB_and_RunControl-20130723_v01.pdf
DB_and_RunControl-20130723_v04.pdf
KM3NeT_DATA_W...es_Machine_DB.pptx
StateMachine
Ideas_on_FSM_Phase1_v1.0_20130522
TriDAS farm
KM3NeT_TriDASFarmDimensioning_v1.0.pdf
Phase1ShoreStationCosts_v.1.0_20140116
Trigger Studies
KM3NeT_DAQ_2011...udy_Bardo_Bakker.pdf
KM3NeT_DAQ_2012..._Tzamariudaki_v1.pdf
Varia
WorkshopReport_20131126_28

Ideas on the Finite State Machine for KM3NeT

Version 1.0 - Wednesday, May 22, 2013

*The present document is made by integrating various discussions held with the components of the Electronics WG (**D. Real**, IFIC, and **V. Van Bevern**, Nikhef), the Offline Software WG (**K. Graf**, ECAP, **C. Bozza**, INFN, **A. Albert**, UHA) and also with contributions from **M. Circella**, INFN, **Eric Heine** and **M. de Jong**, Nikhef .*

1. Introduction

The processes in the DAQ system can be considered as a set of concurrent finite state machines (FSM). Generally speaking, a FSM is an automaton having a finite set of states, and a finite number of binary inputs and outputs. One or more states are identified as initial states, from which the machine starts. A set of mappings determines the next state and next output values, given the current machine state and input values.

In the case of the DAQ of a km^3-sized neutrino telescope, the main challenging issue is concerning the offshore world made of thousands of detection elements , the Digital Optical Modules (DOM), and its onshore counterpart, the computing facility. As it will be reported ahead in Section 4 , the DAQ principle is based on the *all-data-to-shore* approach and on the *time-slicing aggregation* of the occurring hits from the whole detector. This leads to an expected incoming throughput larger than 200 Gbps which must be parted and processed within a framework of high performance parallel computing. Both the submarine detectors and the shore online servers must bookkeep their status and retrieve their configuration parameterization from an offline data base (DB).

1.1 *RUN un*biased Finite State Machine

Because of the high number of concurrent devices and processes, one primary requirement for the experiment is minimizing the measurement dead time due to the (re) configuration of the system as a whole or part of it.
This should lead to a revision of the concept of *RUN* itself, possibly differentiating it with respect to its previous role in the pilot projects ANTARES and NEMO. The considerations which follow are of course functional to the general design of the FSM.

One may think to have the detector always on, and all the meaningful events are recorded whenever they happen. Of course this can only work if one can determine the exact conditions of data taking at the time that each event is detected. Note that in such scenario it would make sense to store the recorded events directly into the DB, since it should contain an updated view of the status of the apparatus, which should be continuously checked also during data analysis.

The implementation of such a DAQ, continuous regardless the occurring status of the detector, requires however compelling problems to solve. First of all one should define the

DB PART OF THE RUNCONTROL VERSION 0.4

ARNAULD ALBERT, CRISTIANO BOZZA, TOMMASO CHIARUSI,KAY GRAF

1. INTRODUCTION

Runcontrol is the central part of the data flow management during data taking period. It ensure the configuration upload to detector element and ask and retrieve all monitoring data from Central Logic Board(CLB). In this framework, database would have an important role to play as well as provider of configuration data, as storage to monitoring data. The aim of this document is to present this role and the way to perform assigned tasks.

2. RUNCONTROL CONTEXT

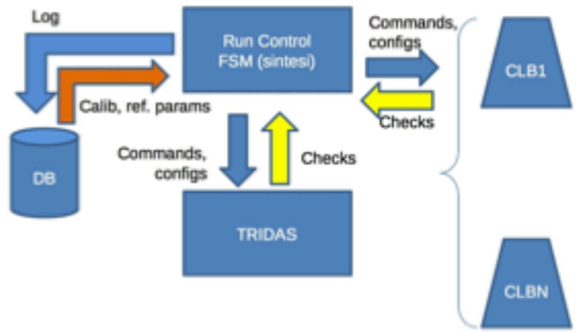


FIGURE 1. Runcontrol state machine

Figure 1 show the different data flux between runcontrol, Trigger and Data acquisition System (TriDAS), Central Logic Board embed on DOM and DataBase (DB). Runcontrol apply setting parameters to all CLBs and TriDAS and retrieve some informations like HV, threshold, Trigger,... regularly from CLB and TriDAS. All these logging data will be stored in DB. Configuration sets are also stored in database to be bookkeep for offline analysis. In order to upgrade efficiency of the bidirectional data transfer between runcontrol and DB, and also prevent data loss in possible network connection cut, we have added a new layer between runcontrol and DB, the cache manager

3. CACHE MANAGER

As it is illustrated on figure 1, Two exchange fluxes will operate between runcontrol and DB; first one will concern setup parameter, upload from database to set each detector component; second one will concern the log data, coming from detector, which will be stored in database in order to follow state of the detector. In order to simplify access and to secure data transmission, all upload in database and download from database will be manage through a cache manager. This software element must be able to perform the following tasks :

Dimensioning the KM3NeT TriDAS farm

Version 1.0 - Thursday, October 3, 2013

The present document aims to dimension the on-line computing farm for each KM3NeT shore station. This Version 1.0 presents preliminary results.

1. Detector layout
The estimations presented here are related to both the *Reference* and *Phase 1* detectors.

1.1. *Reference detector*

The KM3NeT reference detector is defined as the following [1]:

- it is composed of 5 building blocks.
- each building block is composed of 120 strings;
- each string consists of 18 optical modules;
- each optical module consists of 31 PMTs;
- the average horizontal distance between strings is 100 m;
- the vertical distance between optical modules is 36±1 m;

1.2 *Phase 1 detector*
The Phase 1 detector is assumed to be limited to **26 strings**, but maintaining the other features listed for the reference detector.

2. Detector throughput
The dimensioning of the on-line computing farm is strongly influenced by the data throughputs coming from the whole detector.
The main contribution to the data-stream from off-shore to on-shore is produced by the optical detectors. The other sources of information, such as the hydrophones or the oceanographic instruments, contribute to the global throughput with smaller data-streams, if not negligible. So, for this estimation, we basically focus our attention to the data recoded by the PMTs. Further corrections that take into consideration other data types are globally of the order of 5-10%.

2.1 *Hit rates*
The hit information is sized 6 bytes [2]. The TDC electronics can withstand a continuous hit frequency up to 200 kHz per each one of the 3" PMT of the DOM. Of course this is a theoretical limit which is completely unrealistic. Just for reference it would imply a continuous single rate of 2 MHz on a classic 10" PMT. In sites like the Capo Passero one a fairly single hit rate on 10" PMT (with a threshold of 0.3 p.e.) is ranging from 50-60 kHz.

INFN - Sezione di Bologna

Report from DAQ/Readout Workshop in Bologna

November 26-28 2013

T. Chiarusi

The pseudo-octopus board (KM3NeT_ELEC_WD_2013_010),
Simone Biagi, Giuliano Pellegrini, and Riccardo Travaglini

Date: 06/12/2013 Page 1 of 6

The pseudo-octopus board

KM3NeT_ELEC_WD_2013_010

Simone Biagi[1,2], Giuliano Pellegrini[2], and Riccardo Travaglini[2]

[1]Università di Bologna and [2]INFN Sezione di Bologna

Abstract

In this note the design and schematics of the pseudo-octopus board are shown. A proposal for testing the board with the CLBv2 is presented.

Recipients

The KM3NeT Collaboration

Document Status

Revision	Date	Comment	Reviewed by	Approved by
1	10/01/2013			
Draft	06/12/2013	Very first draft	--	--

Revision History

Revision	Date	Description
1	10/01/2013	
Draft	06/12/2013	Very first draft

	KM3NeT_ELEC_WD _2013_0	
	Date: 11/12/2013	Page 1 of 13

CLBv2: Operate (WRPC+IPMUX+LM32_2nd) on the KC705 without the SoftPLL FMC card. (Draft)

KM3NeT_ELEC_WD_2013_010

Peter Jansweijer (NIKHEF Amsterdam, peterj@nikhef.nl)

Vincent van Beveren (NIKHEF Amsterdam, v.van.beveren@nikhef.nl)

Nikhef

Abstract

This document describes how to get the (WRPC+IPMUX+LM32_2nd) up and running on the KC705 board without the need for the SoftPLL FMC card pluged onto it. This enables Ethernet data transfer. However, without the SoftPLL card White Rabbit timing accuracy is not possible.

Recipients

The KM3NeT Collaboration

Document Status

Revision	Date	Comment	Reviewed by	Approved by
Darft 1	06/12/2013	--	--	--

Revision History

Revision	Date	Description
Draft 0	06/12/2013	First Draft

DOM Production Plan for 16 KM3NeT strings

(Version 4 of June 18, 2013)

The KM3NeT Collaboration

Contents