UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES

SR1 ATLAS TRT DEVELOPMENT AND TEST FACILITY

M. Sc. THESIS
IN
ENGINEERING PHYSICS

BY
FARUK DİBLEN
SEPTEMBER 2010

SR1 ATLAS TRT Development and Test Facility

M.Sc. Thesis
in
Engineering Physics
University of Gaziantep

Supervisor
Assoc. Prof. Dr. Ayda BEDDALL

by

Faruk DİBLEN

September 2010

T.C.
GAZİANTEP UNIVERSITY
GRADUATE SCHOOL OF
NATURAL AND APPLIED SCIENCES
ENGINEERING PHYSICS

Name of the Thesis	:	SR1 ATLAS TRT Development and Test Facility
Name of the Student	:	Faruk DİBLEN
Exam Date	:	22.09.2010

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Ramazan KOÇ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. A. NECMEDDİN YAZICI
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ayda BEDDALL
Major Supervisor

Examining Committee Members:

Assoc. Prof. Dr. Ayda Beddall

Prof. Dr. Rauf Mirzababayev

Asst. Pof. Dr. Andrew Beddall

Asst. Prof. Dr. Hümbet Ahmedov

Instructor Dr. Ahmet Bingül

ABSTRACT

SR1 ATLAS TRT DEVELOPMENT AND TEST FACILITY

DİBLEN, Faruk
M.Sc. in Engineering Physics
Supervisor: Assoc. Prof. Dr. Ayda BEDDALL
September 2010, 76 pages

The Transition Radiation Tracker (TRT) is particle tracking detector of the ATLAS Inner Detector which is designed as a robust and powerful gaseous detector that provides tracking through individual drift-tubes (straws) as well as particle identification via Transition Radiation (TR) detec- tion. The TRT consists of 370000 straw proportional tubes of 4 mm diameter with a 30 micron anode wire, which operates with a Xe/CO2/O2 gas mixture at a high voltage of approximately 1.5 kV. This detector enters a new area that requires it to operate at unprecedented high rates and inte- grated particle fluxes. Full functionality of the detector over the lifetime (10 years) of the experiment is demanded.

This MSc. thesis presents development, testing process and cosmic data analysis of a replica has been built in an above-ground laboratory environment in SR1 at CERN.

Key Words: TRT, ATLAS Experiment, CERN, SR1, LHC

ÖZET

ATLAS TRT DEDEKTÖRÜ'NÜN SR1 ORTAMINDA GELİŞTİRME VE TEST ANALİZİ

DİBLEN, Faruk
Yüksek Lisans Tezi, Fizik Müh. Bölümü
Tez Yöneticisi: Doç. Dr. Ayda BEDDALL
Eylül 2010, 76 sayfa

Geçis ışınımı iz sürücüsü (TRT), ATLAS icindeki dedektörlerden biri olan parçacık iz sürücü dedektörüdür. TRT içerisinde bulunan drift tüpler(straw) sayesinde parçacık iz sürümünü yaptığı gibi, aynı zamanda Geçiş Işınımı kullanarak parçacıkların özelliklerinin tanımlanmasına da yardımcı olur.

TRT dedektörü 370000 adet 4 mm çapa sahip ve içerisinde 30 mikronluk anod tel bulunan tüplerden oluşur. Çalışması için yaklaşık olarak 1.5 kV yüksek gerilime ihtiyaç vardır ve geçiş ışınımının oluşabilmesi için Xe/CO2/O2 gaz karışımı kullanılmaktadır.

Bu Yüksek Lisans çalışmasında, şu anda CERN'de bulunan SR1 geliştirme ve test laboratuvarında geliştirme çalısmaları devam eden replika TRT dedektörünün kurulum, geliştirme, test çalışmaları ve kozmik data analizi yapılmıştır.

Anahtar Kelimeler: TRT, ATLAS Deneyi, CERN, SR1, LHC

I dedicate this work to my parents.

ACKNOWLEDGEMENTS

I would like to express my deeply-felt appreciation and sincere gratitude to my supervisor Assoc. Prof. Dr. Ayda Beddall, for her guidance from the beginning to the end in the course of my research. I also thank to Dr. Andrew Beddall who brought me into the world of research, guiding me through the writing of the thesis, and presenting me for the interesting and engaging viewpoints.

I am thankful to Ahmet Bingül for his support and helps during my studies, and also for being such a friendly teacher.

A warm thanks to Andrey Loginov for his invaluable encouragement and help during my work at CERN and during the writing process, seeing me through to the finish and for proof reading and giving me comments.

I am grateful to Christoph Remsber for his support during my studies at CERN.

I would like to thank Professor Serkant Ali Çetin and Professor Metin Arık of being so supportive during my all studies. Also want to thank TAEK for being the financial assistance, without this support my research would not have been possible.

I also want to express my gratitude to Ruslan Mashinistov, Peter Wagner, Mike Hence, Dominick Olivito, Vadim Kantzerov, Konstantin Zhukov, Markus Joos, Jahred Adelman, Elzbieta Banas, Zbyszek Hajduk, Jolanta Olszowska, Krill Egorov, Peter Cwetanski, who helped me during my fieldwork at CERN.

Also special thanks are owed to Gökhan Ünel, Erkcan Özcan, Enver Alagöz, Öznur Mete for their friendliness and sincereness.

To my friends who were at CERN during the time of this thesis, all I want to say is: thanks for just being there.

Finally, I am deeply grateful to my parents Birsen and Hasan Diblen, my sister Berna and my brother Anıl for the unerring support, belief and encouragement. I express my profound appreciation to my girlfriend Sinem for her encouragement, understanding, and patience even during hard times of this study.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

Table

LIST OF FIGURES

Figure

LIST OF SYMBOLS

fb: Femto barn

$\mathscr{L}$: Luminosity

ε: Efficiency

L_{max}: Maximum Luminosity

Thr: Threshold

γ: Lorentz factor

P_T: Transverse momentum of the particle

LIST OF ABBREVIATIONS

ADC: Analog-to-digital converter

ASDBLR: Amplifier-Shaper-Discriminator with BaseLine Restoration (ATLAS TRT)

ASIC: Application-Specific Integrated Circuit

ATLAS: “A Toroidal LHC Apparatus”; one of the detectors on the LHC

BX: Bunch-crossing

CERN: Counseil Europen pour la Recherche Nuclaire (The Council created in 1951 was a provisional body, that decided in 1953 to build a lab- oratory officially called Organisation europenne pour la recherche nuclaire or European Organization for Nuclear Research. However, the name of the Council stuck to the organization. It is a common mistake to think of the C of CERN as the first letter of Centre.)

CMS: Compact Muon Solenoid

CORAL: Oracle database used for detector configuration

CSC: Cathode Strip Chamber

DAC: Digital to analog converter

DAQ: Data acquisition

DCS: Detector Control System

DME: Dimethyl ether

DTMROC: Drift Time Measurement Read-Out Chip (ATLAS TRT)

ECAL/ECal: Electromagnetic Calorimeter

ELMB: Embedded Local Monitoring Board

FE: Front End

GOL: Gigabit Optical Link

HCAL/HCal: Hadronic Calorimeter

HTML: Hyper Text Markup Language

HV: High-voltage

HWI: Hardware Interlock

LAr: Liquid Argon

LHC: Large Hadron Collider

LHCb: A dedicated LHC Beauty experiment for precision measurements of CP-violation

LV: Low Voltage

MDT: Monitored Drift Tubes (of the ATLAS experiment)

MIP: Minimum ionizing particle

MSSM: Minimally Supersymmetric Standard Model

MWPC: Multiwire proportional counter

NSE: Network Search Engine

NTC: Negative Temperature Coefficient

OKS: Object Kernel Source

OPC: OLE for Process Control

PMG: Process Manager

PP: Patch Panel

PP2: Patch Panel box

PVSS: Software used for detector control

PCB: Printed-circuit board

ROD: Readout Driver

ROS: Readout System

RPC: Resistive Plate Chamber

SC: System Controller

SBC: Single Board Computer

SCT: Semiconductor Tracker (of the ATLAS experiment)

SEM: Scanning Electron Microscope

SM: Standard Model

SPD: Scintillator Pad Detector

SQS: Self-quenched streamer

SUSY: Supersymmetry

SWPC: Single wire proportional chamber

TGC: Thin Gap Chamber

TOTEM: A TOTal and Elastic Measurement experiment

TR: Transition Radiation

TRD: Transition Radiation Detector

TRT: Transition Radiation Tracker (of the ATLAS experiment)

TTC: Timing, Trigger and Control

VME: Virtual Machine Environment

WEB: Wheel End-cap Board (ATLAS TRT)

WTS: Wheel Test Station (ATLAS TRT)

CHAPTER 1

INTRODUCTION

1.1 Introduction

Given the complexity and inaccessibility of the Transition Radiation Tracker during ATLAS data taking, a replica has been built in an above-ground laboratory environment in SR1. The replica is built from the same components as the detector in the ATLAS detector to provide a testbed for investigations of the detector, its long-term behavior, debugging of problems, and studies related to modications and improvements. The system also serves as a platform for development, testing, and validation of the data acquisition, detector control, and calibration software.

After few years of LHC running it is planned to have a upgrade of detectors. If the Higgs and/or some other particles exist, it can be expected to discover them with order of $10 fb^{-1}$ of integrated luminosity [1].

To establish TeV scale physics firmly after discovery, it becomes important to make precision measurements of the discovered new phenomena, and also to extend and continue searches for new phenomena that have low rate or higher mass scale. Thus, it is planned to upgrade the LHC so as to bring a significant increase in luminosity, aiming for an ultimate goal of $L_{max} = 10^{35} cm^{-2} s^{-1}$, i.e. ten-times higher than the original LHC design (called super-LHC, sLHC). This project is planned in two phases. In the first phase ("Phase-1") $L_{max} = 3 \times 10^{34} cm^{-2} s^{-1}$ is a goal. During a shutdown in 2016, it is foreseen to upgrade the Linear accelerator systems. About few hundreds fb^{-1} of integrated luminosity will be delivered with about 8 years of operation by the end of the Phase-1.

The test stand in SR1 will be used for Phase-1 upgrade studies. Because of increase in luminosity new type of gases must be used. The replica detector will be used to find proper gas mixture for the higher luminosity. Also replica will be used for education of newcomers, testing new ideas about TRT, debugging problems in the ATLAS TRT detector.

This thesis will describe aspects of the Transition Radiation Tracker (TRT) detec-

tor in SR1 from development to commissioning period. We will begin with an introduction to the LHC, as well as an overview of the ATLAS detector. Chapter 2 also gives information about the theory of Transition Radiation and the working principle of the Transition Radiation Tracker. Chapter 3 focuses on development efforts and working principles of the electronics. The timing parameters for the control and readout of this detector and constraints are discussed. The Object Kernel Source (OKS) library which is used to create a configuration database for DAQ system is explained with giving some configuration examples.

CHAPTER 2

LHC & ATLAS EXPERIMENT

For the past few decades, physicists have been able to describe with increasing detail the fundamental particles that make up the Universe and the interactions between them. This understanding is encapsulated in the Standard Model of particle physics, but it contains gaps and cannot tell us the whole story. To fill in the missing knowledge requires experimental data, and the next big step to achieving this is with LHC. This chapter gives general information about Large Hadron Collider and ATLAS Experiment.

2.1 LHC

The LHC is the world's largest and most powerful particle accelerator. It mainly consists of a 27 km ring of superconducting magnets with a number of accelerating structures to boost the energy of the particles along the way.

Inside the accelerator, two beams of particles travel at close to the speed of light with very high energies before colliding with one another. The beams travel in opposite directions in separate beam pipes two tubes kept at ultrahigh vacuum. They are guided around the accelerator ring by a strong magnetic field, achieved using superconducting electromagnets. These are built from coils of special electric cable that operates in a superconducting state, efficiently conducting electricity without resistance or loss of energy. This requires chilling the magnets to about -271 ^{o}C a temperature colder than outer space. For this reason, much of the accelerator is connected to a distribution system of liquid helium, which cools the magnets, as well as to other supply services.

Thousands of magnets of different varieties and sizes are used to direct the beams around the accelerator. These include 1232 dipole magnets of 15 m length which are used to bend the beams, and 392 quadrupole magnets, each 5-7 m long, to focus the beams. Just prior to collision, another type of magnet is used to 'squeeze' the particles closer together to increase the chances of collisions. The particles are so tiny that the task of making them collide is akin to firing needles from two positions 10 km apart

with such precision that they meet halfway.

All the controls for the accelerator, its services and technical infrastructure are housed under one roof at the CERN Control Centre. From here, the beams inside the LHC are made to collide at four locations around the accelerator ring, corresponding to the positions of the particle detectors. Positions of the detectors at accelerator ring are shown in Figure 2.1.

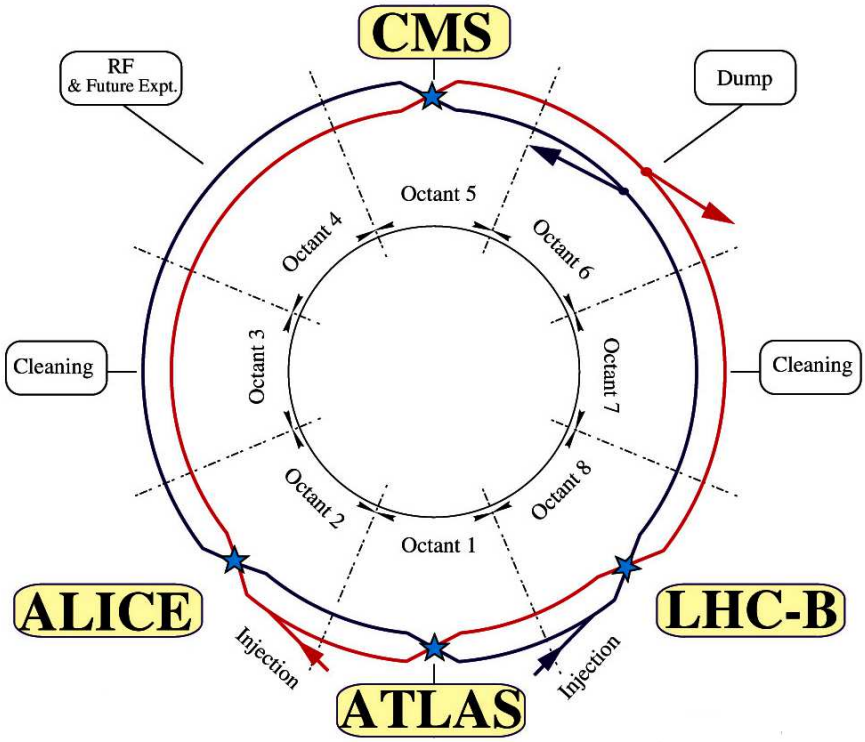


Figure 2.1: *LHC Layout*.

The first beam was circulated through the collider on the morning of 10 September 2008. CERN successfully fired the first protons around the entire tunnel circuit in stages. The particles were fired in a clockwise direction into the accelerator and successfully steered around it, and CERN next successfully sent a beam of protons in a counterclockwise direction.

On 19 September 2008, a quench occurred in about 100 bending magnets in sectors 3 and 4, causing a loss of approximately six tonnes of liquid helium, which was vented into the tunnel, and a temperature rise of about 100 K in some of the affected magnets. Vacuum conditions in the beam pipe were also lost. Shortly after the incident CERN reported that the most likely cause of the problem was a faulty electrical connection between two magnets, and that due to the time needed to warm up the affected sectors and then cool them back down to operating temperature it would take at least two months to fix it. Subsequently, CERN released a preliminary analysis of the inci-

dent on 16 October 2008, and a more detailed one on 5 December 2008. Both analyses confirmed that the incident was indeed initiated by a faulty electrical connection. A total of 53 magnets were damaged in the incident and were repaired or replaced during the winter shutdown.

After this period, low energy beams circulated in the tunnel for the first time since the incident on 20 November 2009, and the first particle collisions in all 4 detectors at 450 GeV on 23 November 2009. On 30 November 2009, LHC becomes the world's highest energy particle accelerator achieving 1.18 TeV per beam, beating the Tevatron's previous record of 0.98 TeV per beam held for 8 years. The LHC expected to continue operations ramping energies to run at 3.5 TeV for 18 months to two years, after which it will be shut down to prepare for the 14 TeV collisions (7 TeV per beam).

2.2 ATLAS Experiment

ATLAS detector which is shown in Figure 2.2 is one of the six particle detector experiments constructed at LHC. ATLAS is 44 meters long and 25 meters in diameter, weighing about 7000 tones which is designed to study the proton-proton collisions at CERN Large Hadron Collider (LHC), at a center of mass energy of 14 TeV and bunch crossing rate of 40 MHz.

ATLAS will search for new discoveries in the head-on collisions of protons of extraordinarily high energy. ATLAS will learn about the basic forces that have shaped our Universe since the beginning of time and that will determine its fate. Among the possible unknowns are the origin of mass, extra dimensions of space, microscopic black holes, and evidence for dark matter candidates in the Universe.

The ATLAS experiment offers a large physics potential with major focus on electroweak symmetry breaking, particularly the search for the Higgs boson, a hypothesized particle which, if it exists, would provide the mechanism by which particles acquire mass [2].

The configuration of the ATLAS detector is typical for experiments at particle colliders: subdetectors are arranged in concentric layers around the beam axis (the barrel section) and in wheels (the endcaps) that close both ends of the cylinder.

The first barrel layer, located next to the interaction point, is the Inner Detector; this apparatus is composed of three sub-detectors: (from inside to outside) the Pixel Detector, the Semi-Conductor Tracker (SCT) and the Transition Radiation Tracker. The data of the Inner Detector make it possible to reconstruct primary and secondary vertices and to measure momenta of charged particles. The reconstruction of secondary vertices is crucial for the identication of b-quarks.

The Inner Detector is surrounded by a cryostat, which contains the Central Solenoid and the Liquid Argon (LAr) calorimeter. The Central Solenoid has a superconducting

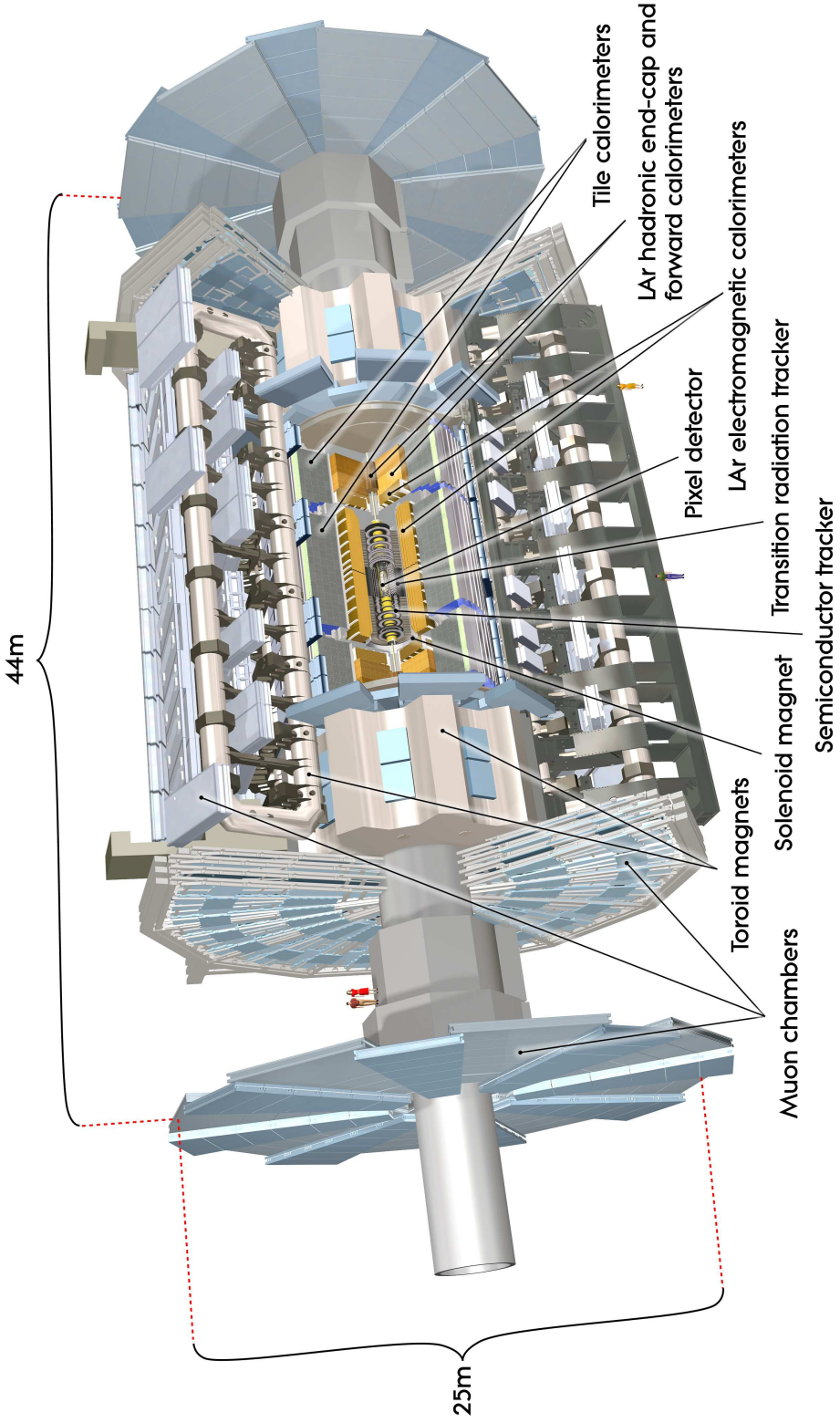


Figure 2.2: *The ATLAS Detector.*

coil and generates a 2T solenoidal magnetic field, oriented along the beam axis. This allows for the measurement of the momenta of charged particles in the Inner Detector.

The liquid argon calorimeter is responsible for the measurement of the energy of electromagnetic showers (electrons, photons). The ATLAS calorimetry in the barrel region is completed by the TileCal calorimeter, which measures the energy deposited by showers not contained in the liquid argon calorimeter mostly caused by hadrons. The steel structure of the TileCal is used as yoke for the return flux of the solenoid.

The outermost ATLAS barrel layer is the Muon Spectrometer. The spectrometer uses a separate magnetic field, generated by eight superconducting coils deployed radially around the beam axis; the coils create a 4T toroidal magnetic field that encompasses the whole spectrometer.

The Muon Spectrometer uses in the barrel region two types of chambers: Monitored Drift Tube (MDT) chambers for precision measurements, and Resistive Plate Chambers (RPC) for triggering.

The two endcaps are identical and consist of a cryostat containing the liquid argon electromagnetic and hadronic calorimeters. Behind the cryostat a wheel of Cathode Strip Chambers (CSC), which are part of the trigger of the Muon Spectrometer, is located.

The endcaps have their own 4 T toroidal magnetic fields generated by two eight–coil toroids that plug into the ATLAS barrel.

The Muon Spectrometer in the endcap region consists of MDT chambers and Thin Gap Chambers (TGC), which complete the muon trigger. MDTs and TGCs are mounted on wheels placed between the ATLAS endcaps and the walls of the underground experimental area.

Two high density Forward Calorimeters cover the very forward region of ATLAS and enhance the hermeticity of the detector. Two beam shields connect ATLAS to the LHC accelerator and protect the endcap instrumentation from beam radiation and RF fields.

2.3 The Inner Detector

The Inner Detector (in Figure 2.3) is the subdetector closest to the interaction point of the ATLAS experiment. The goals of this detector are:

- reconstruct with high efficiency and low fake rate all charged particle tracks with $|\eta| < 2.5$ and $p_T > 0.5$ GeV;
- measure precisely the p_T of these charged particles with a resolution of 30% at $p_T = 500$ GeV;
- measure accurately the position of the primary vertex and identify with high accuracy secondary vertices;

- provide high quality electron identification.

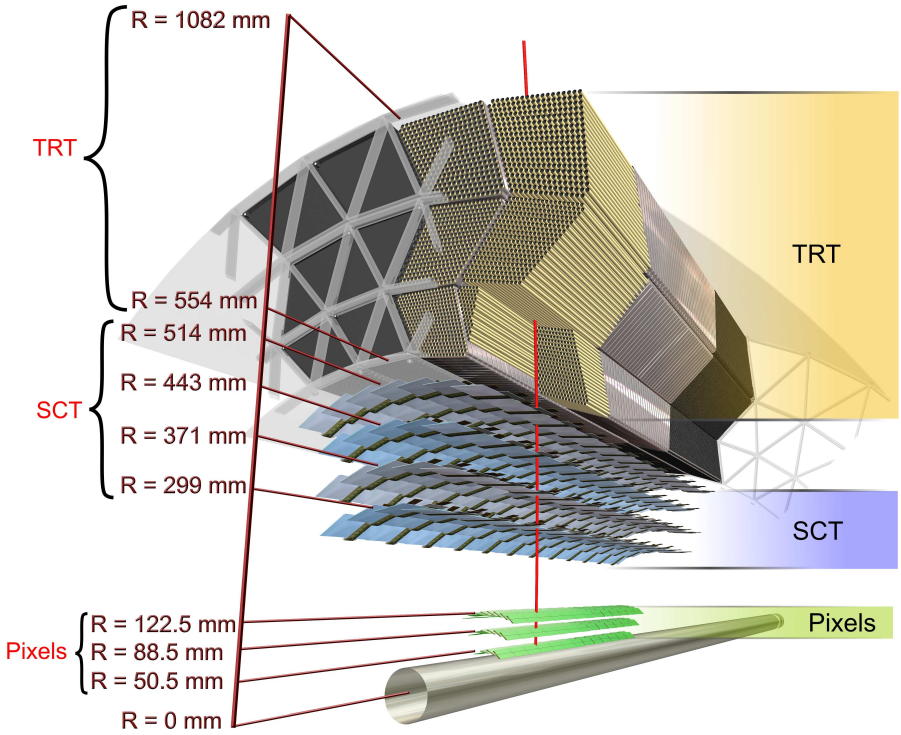


Figure 2.3: *Inner Detector.*

The Pixel Detector and the Semi Conductor Tracker are based on reverse biased semiconductor technology, while the Transition Radiation Tracker is a gaseous detector. The use of different technologies is the result of a trade–off between high spatial resolution and a lightweight construction. Semiconductor detectors guarantee a high precision in measuring space points and tracks typically the resolution is of the same order of magnitude as the size of the active elements of the detector, that is the order of 10 μm. The density of the semiconductor substrate and the need for cooling and readout services implies high interaction and radiation lengths, which can cause multiple scattering and energy loss to the traversing particle. This results in a degradation of the measurements' quality for the detectors, which are downstream of the Inner Detector. The choice of ATLAS was to limit the amount of semiconductor detectors and to include in the Inner Detector a gaseous tracker. The TRT is more transparent to incoming particles, reducing the amount of multiple scattering. The lower spatial resolution of the TRT is compensated by the higher number of space points measured along the

particle tracks. Moreover, the TRT can exploit the transition radiation phenomenon to discriminate between electrons and other charged particles.

The inner detector is designed to provide a very robust pattern recognition, high precision in both z and ϕ coordinates, good momentum and vertex resolution as well as electron identication. The very large track density expected at the LHC requires high granularity detectors at the inner radii in order to achieve good momentum and vertex resolution. The silicon pixel detector is formed by three pixel layers, which gives three high resolution space points per track, and is placed as close as possible to the beam pipe. The innermost layer, also called the B layer, is placed at 4 cm from the interaction point. The silicon microstrip detector (SCT) surrounds the pixel detector, and is composed by eight SCT layers, which give four space points per track. The total number of precision layers is limited because of the material they introduce and because of their cost. At the outer radii a large number of position measurements (on average 32 per track) is provided by the straws of the Transition Radiation Tracker, which provides continuous tracking at very low cost, and in addition provides electron identication capability.

2.3.1 Pixel Detector

The pixel detector is located in the innermost region of the inner detector. It is a very high granularity detector, composed of more than 140 million pixels, that has been designed to provide excellent pattern recognition; it determines the impact parameter resolution and the ability of the inner detector to find very short–lived particles, such B hadrons and leptons. Each track will cross at least three pixels, resulting in three high precision space points. The pixel dimensions are $50 \times 300 \mu$m. The occupancy, due to the very high granularity, is extremely low: only about one pixel out of ten thousand gives a signal at the LHC design luminosity.

The pixels are organized in modules, identical in the barrel and in the endcaps. Each module is 62.4 mm long and 21.4 mm wide for a total of 61440 pixel elements read out by 16 chips. The readout chips, bump–bonded to the detector, contain individual circuits for each pixel element, including buering to store the data during the level one trigger latency. The barrel part is formed by three layers of modules, at average radii of 4,10 and 13 cm, while the endcaps are formed by 5 disks of modules on each side, between radii of 11 and 20 cm.

2.3.2 Semiconductor Tracker

The SCT covers the intermediate radial range of the inner detector. It provides eight precision measurements (four space points) per track over the full rapidity coverage, contributing to the momentum, impact parameter and vertex position measurement.

Each barrel module consists of four elements. On each side of the module, two detectors are wire bonded together to form 12.8 cm long strips. Two of such detector pairs are then glued together back to back at a 40 mrad angle, separated by a heat transport plate; the electronics is mounted above the detector. The readout chain consists of a frontend amplier and discriminator, followed by a binary pipeline which stores the hits above threshold during the level one trigger latency. The endcap modules are very similar in design but use tapered strips (one set aligned radially), whose length, varying between 6 to 12 cm, has been chosen in order to optimise the rapidity coverage. The barrel modules are mounted on carbonfiber cylinders which carry the cooling system. Four complete barrel layers, linked together, are located between radii of 30.0 and 52.0 cm. The endcap modules are mounted into nine wheels, and interconnected by a space–frame.

2.3.3 The ATLAS Transition Radiation Tracker

The Transition Radiation theory

When a fast-charged particle crosses the boundary between two media with different dielectric constants, the abrupt change in the electric field triggers the emission of electromagnetic radiation which is called Transition Radiation(TR). The emitted radiation is the homogeneous difference between the two inhomogeneous solutions of Maxwell's equations of the electric and magnetic fields of the moving particle in each medium separately. In other words, since the electric field of the particle is different in each medium, the particle has to "shake off" the difference when it crosses the boundary. The total energy loss of a charged particle on the transition depends on its Lorentz factor $\gamma = E/mc^2$ and mostly directed forward, peaking at an angle of the order of $1/\gamma$ relative to the particle's path. The intensity of the emitted radiation is roughly proportional to the particle's energy E.

Transition Radiation was first predicted by V. L. Ginzburg and I. M. Frank in 1944, and in 1957 G. M. Garibian showed the feasibility of functional transition radiation detectors (TRDs). Since then TRDs have been successfully used in a number of experiments, mainly in high-energy particle physics and astrophysics.

Transition radiation has three major features that influence the design of these detectors:

- the total radiation energy emitted at the boundary is proportional to the particle's Lorentz factor - the ratio of its total energy to its rest mass energy;
- the average number of photons emitted per boundary is rather small;
- the emitted photons travel in practically the same direction as the charged particle.

The first feature suggests two major applications; either to discriminate between particles with different masses and the same momentum or to measure the energy of a known particle. The second feature means that, in order to have a significant number of photons, many boundaries are required. This can be achieved either by means of "regular radiators" built with many evenly spaced foils or with "irregular radiators" made of foams or fibers.

The characteristics of this electromagnetic radiation makes it suitable for particle discrimination, particularly of electrons and hadrons in the momentum range between 1 GeV/c and 100 GeV/c. The transition radiation photons produced by electrons have wavelengths in the X-ray range, with energies typically in the range from 5 to 15 keV. However, the number of produced photons per interface crossing is very small. Usually several layers of alternating materials or composites are used to collect enough transition radiation photons for an adequate measurement for example, one layer of inert material followed by one layer of detector (e.g. microstrip gas chamber), and so on. By placing interfaces (foils) of very precise thickness and foil separation, coherence effects will modify the transition radiation's spectral and angular characteristics. This allows to obtain a much higher number of photons in a smaller angular "volume". Applications of this x-ray source are limited by the fact that the radiation is emitted in a cone, with a minimum intensity at the center. X-ray focusing devices (crystals/mirrors) are not easy to build for such radiation patterns.

The ATLAS Transition Radiation Tracker

The Transition Radiation Tracker is the outermost subsystem of the inner detector at ATLAS which is immersed in a 2 T solenoid field. The TRT is a straw drift-tube detector operated in proportional mode with a total of 350848 readout channels. It contributes significantly to the precision of the momentum measurement of charged particles and to the identification of electrons. As a continuous tracker it provides a large number (~ 30) of measurement points (straw hits) on track with a hit resolution of 130 μm. The straw drift tubes are made of thin Kapton-based multilayer material. They have a diameter of 4 mm and contain a 30 μm diameter, gold-plated tungsten anode wire in the center. The small tube radius limits the maximum drift time to $\sim 50ns$. The length of the straws has been chosen such that the counting rate per wire is not more than $\sim 20MHz$ at LHC design luminosity. The straw wall, which lies at a potential of $\sim 1.5kV$ relative to the wire, provides a separation between the transition radiation medium and the active gas ($70\%Xe, 27\%CO_2, 3\%O_2$). To minimize the number of radiation lengths only light materials were used throughout the detector. When a charged particle traverses a straw it typically loses a few hundred eV energy to ionization clusters of electrons and ions in the gas. The liberated electrons drift towards the wire where the

high electric field leads to secondary ionizations in the gas with an average gain of 25000. This signal is modified with an analog Amplifier Shaper Discriminator Baseline Restorer (ASDBLR). To permit a hit resolution of 130 μm, the signal is then recorded in a digital pipeline of 3.125 *n*s wide time bins in a Digital Time Measurement Read-Out Chip (DTMROC) before it is sent to the back-end electronics. The DTMROC can further;

- configure the ASDBLR to inject a test pulse used to estimate the effects of radiation damage,
- sense the low voltage applied to both ASDBLR and DTMROC,
- transmit configuration and sense data to the trigger, timing and control (TTC) back-end during data taking used for e.g. single event upset recovery, and
- transmit a trigger signal to the TTC back-end.

CHAPTER 3

SR1 FACILITY

3.1 Introduction

This chapter focuses on the replica which has been built in laboratory environment. Information about mechanical construction, detector readout, timing properties and synchronization of signal are described.

The test setup has both TRT barrel and endcap sectors in SR1, commissioning the barrel part first and working on the endcap part in parallel.

3.2 Mechanical Construction

The test stand was built from spare Barrel and Endcap components of the detector in the ATLAS detector. The Barrel (Section 3.2.1) and Endcap (Section 3.2.2) parts of the test stand have been installed in a custom made Bosch profile frame shown in Figure 3.1.

3.2.1 Barrel

The barrel test stand which consists of 2 sectors, sector 24 and sector 25 has been assembled from spare modules. The assembly process included;

- Installation of Type-I, Type-II and Type-III modules to a 2-sector TRT barrel frame (Figure 3.2A)
- Quality tests and installation of protection boards which plug directly into the straw to protect the ASDs from over current (Figure 3.2B)
- Quality tests and installation of Front End (FE) boards (1BS, 1BL, 2BS, 2BL, 3BS, 3BL for the Back Side and 1FS, 1FL, 2FS, 2FL, 3FS, 3FL for the Front Side). For the barrel, the ASDs and DTMs are mounted on FE boards (Figure 3.2C)

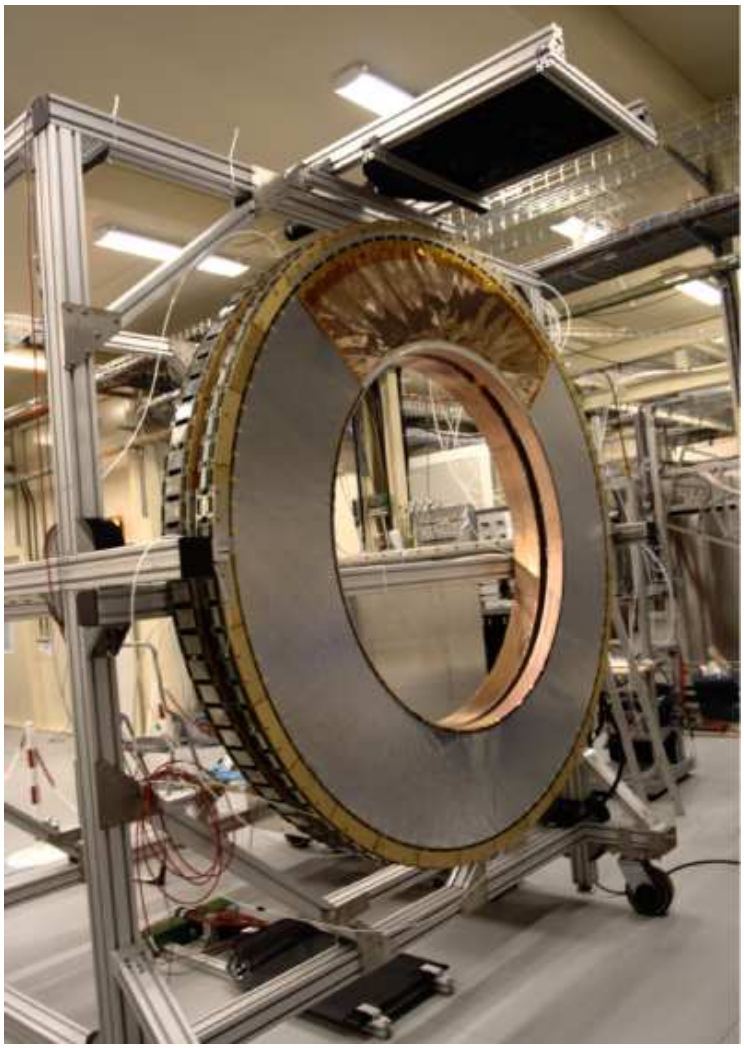
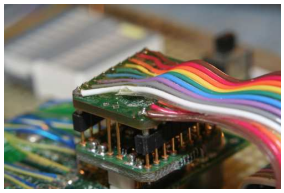


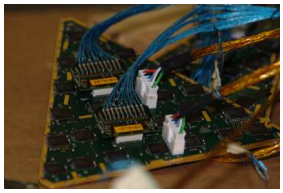
Figure 3.1: System installed in to a Bosch Profile frame. Detector components, readout system (PP2 boxes with electronics, See Section 3.3), cabling is setup in the Bosch profile to provide mobility of the test stand. The system is designed keeping in mind a possibility to use the setup at a test beam for Phase I upgrade studies.

- Installation of custom cooling plates
- High Voltage connections and tests
- Cable connections from the detector to PP2
    - DATA and TTC
    - Low Voltage
- Cooling, active gas connections and relevant leak tests

A: The TRT Barrel Frame



B: Testing a protection board



C: Barrel FE Board

Figure 3.2: Barrel Assembly

3.2.2 Endcap

The endcap in the test stand consists of two wheels, a wheel A and a wheel B has been assembled from spare wheels. The assembly process includes;

- Installation of the wheels to a Bosch profile
- Quality tests and installation of endcap FE boards. There are two types of boards for the endcaps,
  - triplet boards, which have the DTMs,
  - a type A or a type B ASD board corresponding to the two types of Endcap wheels.

  For each of A wheels or B wheels, one ASD board plugs into a single (third) of triplet.
- High Voltage connections / tests
- Cable connections from the detector to PPF1 and then from PPF1 to PP2
  - DATA / TTC (blue harness)
  - Low Voltage (yellow/golden harness)
- Cooling / active gas connections and relevant leak tests

3.3 Components Used For The Setup

This sections presents components used in setup.

FE electronics is connected to PPs and powered from PP2 boxes, where Low Voltage (LV), Trigger, Timing, Control (TTC) and Read-Out Driver (ROD) patch panels (PP) are installed.

Table 3.1 shows the number of the components used for the setup.

Component	**Barrel**	**EC**
LV PP	4	2
PP2 with LV Distribution Box	2	1
FE to LV PP Cables	4	2
PP2 to Maraton Cables	8	4
TTC PP + TTC Passive Boards	4	2
Type 3 TTC cables	4	2
TTC	1	1
FE TTC/ROD Harness	4	2
ROD PP + ROD Passive Boards	4	2
ROD	2	1
ROD PP to ROD Fibers	16	1
eth cables	2	1
FE to PPF1 Harness	N/A	1

Table 3.1: List of components used for the setup. The barrel setup has 4 sectors total: 2 sides, 2 sectors on each side. The endcap setup has ∼15-30% of the wheel A and the wheel B equipped with FE electronics / readout.

3.4 High Voltage System

To provide the straws in Barrel and End-cap with High Voltage the HV Power Supply is used [3]. It consists of crate (operated by computer through controller) and high voltage modules.

The High Voltage System (HVS) was designed to provide powering of large arrays of chambers or photomultipliers in physical detectors in large-scale experiments. Depending on particular detector requirements the HVS may consist of different type of cells and controller(s). A cell is a unit where high voltage is generated from a base voltage (220V DC). Cockroft Walton multipliers are used for generation of high voltage. Every cell has its own intelligence i.e. microprocessor, which controls the cell, checks its operations and communicates with a System Controller (SC). Every system controller may feed and control up to four branches with up to 127 cells on every branch. A custom dedicated system bus connects the system controller to high-voltage cells. It

consists of power and signal lines. The system is designed according to a sectional principle. Each section, consisting of a system controller and up to 4 branches with cells, is functionally completed device and does not depend on other sections. The functional chart of high voltage system (HVS) is presented on Figure 3.3.

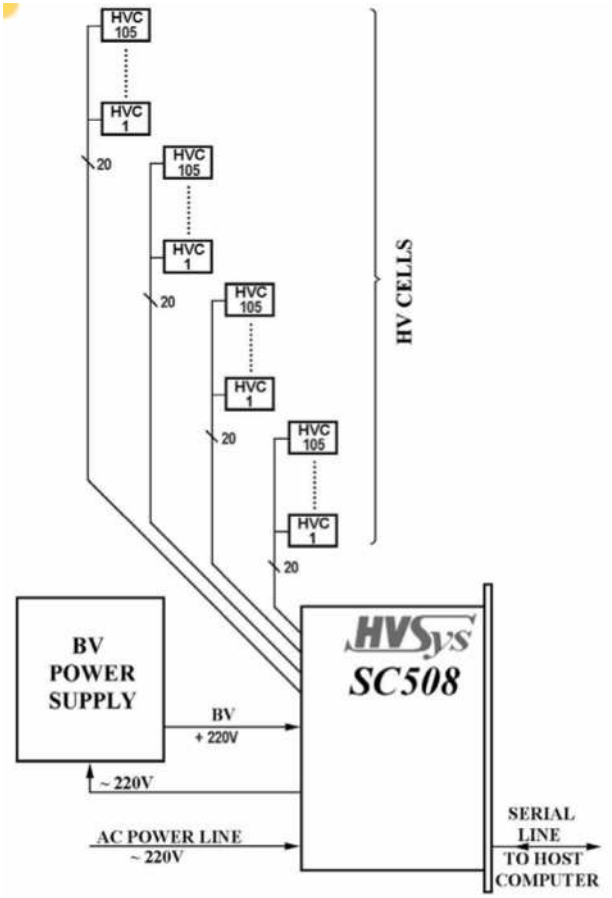


Figure 3.3: *High Voltage System functional chart*

3.5 Low Voltage Power Supply System

This section presents a low voltage power supply system which has to deliver to the front end electronics of the detector. The front end electronics of the TRT is based on two custom designed ASICs: ASDBLR (amplifier-shaper-discriminator- baseline-restorer) which amplifies, shapes and discriminates the charge signal from straw anode and DTMROC (drift-time- measuring-read-out-chip) which measures the time-of-arrival of the signal with respect to the bunch crossing, encodes the charge value and serves as the read-out controller [4]. Both chips are designed in radiation hard technologies. Total power consumption of individual channel is 40 mW/channel for ASDBLR and 21 mW/channel for DTMROC. Modern fast electronics requires very

low voltage power supplies which results in high supply currents. Then a length of the cables, their cross-section and voltage drops are important factors for the power supply system design.

The FE electronics of the TRT requires +/-3 V (analogue part) and +2.5V (digital part). The auxiliary circuitry requires additional supply of +5V with relatively low current.

To minimize the need for development, service and maintenance effort commercially available hardware WIENER MARATON power system has been chosen. The system, developed originally for the needs of LHC experiments, is able to operate in the moderate magnetic and radiation fields existing in the ATLAS experimental cavern during LHC operation.

The voltage distribution has been based on the availability of radiation hard voltage regulators. The granularity of the sub-detector front-electronics is such that each granule consumes a current which can be delivered by 1 (or max. 2) voltage regulator i.e. $\sim 3A$. The regulators together with control and monitoring electronics has been arranged onto boards which are housed in specially designed liquid cooled simple crates called Patch Panels 2 (PP2).

Figure 3.4 shows the current schematic for power supply cabling in SR1.

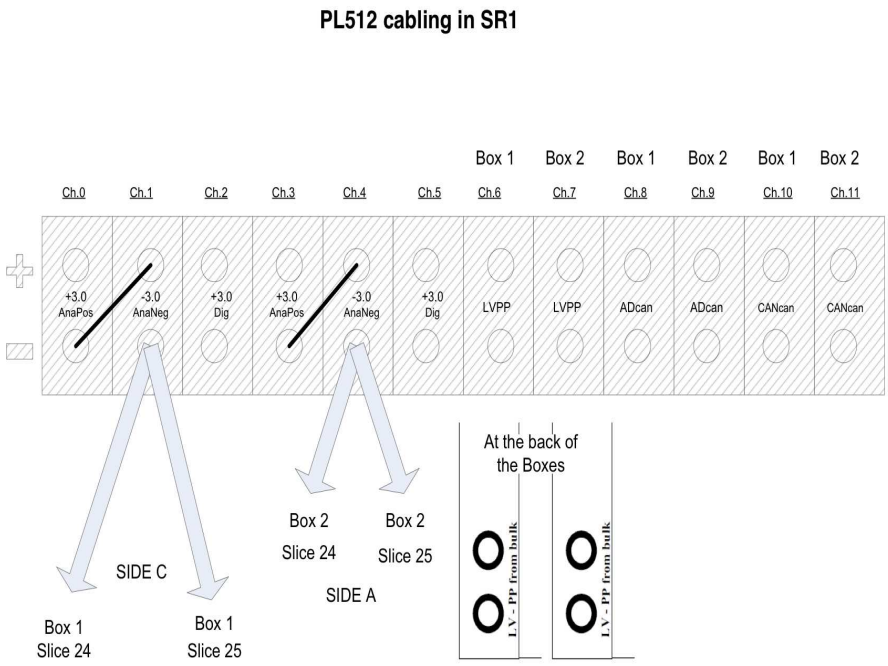


Figure 3.4: *Cabling schematic of the LV System.*

(1) Radiation hard voltage regulators

The LHC(X)913 is a positive/negative voltage regulator family. Housed into power SO-20 slug-up, it is specifically intended for applications in rugged environments, subject to ionizing and neutron radiation. Input voltage ranges from (-)3 to 12(-9) volts. It is characterized by a low voltage drop (0.5 volt@ I_{out}=1A and 1.5V@I_{out}=3A). The over-temperature, and over-current protections guarantee robust operation. Output short circuit monitoring, signalled by TTL output allows for status detection. The regulator can be enabled/disabled by external TTL signal which in our application is going to be used for control of each output.

(2) Low Voltage Patch Panels

Groups of regulators supplying geographically close parts of detector front-end electronics have been placed on a printed circuit board together with the control and monitoring electronics. One card of 440x200 mm size houses up to 36 regulators (positive and negative) delivering 6/8 voltage sets : +/- 3 V for analogue electronics and +2.5 V for digital part. Three boards are used to power a 1/32 phi slice of the detector: 1 for the barrel part and two for the end-cap wheels. Due to high power dissipation on the board (60 - 70 W) and to the requirement of being thermally neutral, it is equipped with an aluminium cooling plate which transports heat to the crate body where it is taken away by liquid cooling circuit. Thermal contact between regulators packages and cooling plate is assured by a layer of special thermo conductive rubber. Figure 3.5 presents the board photo (control side).

Figure 3.5: Low voltage patch panel board

(3) Voltage regulation and setting

The regulators used are of the adjustable version. Changing the voltage adjust input allows output to be set at the value assuring the proper operation of the f-e electronics during life cycle of the experiment when semiconductor devices could change its characteristics due to radiation. The variable voltage is delivered by radiation hard DAC embedded in the DTMROC chip. The settings of the DACs are controlled over simple 3-wire serial protocol. The current swing of the DAC output allows for regulators output to be varied by 0.5V up to 1.2 V, which is fully covering expected change of the voltage supplies of FE electronics.

(4) Voltage and current measurements in each individual output line.

The board contains embedded controller an ELMB (Embedded Local Monitoring Board), standard control unit designed for ATLAS. All voltage outputs are connected to the ELMBs ADC allowing for output voltage measurements. The same ADC is measuring the output currents, by monitoring the voltage drop on 22 mOhms serial resistors inserted in output lines.

(5) Over Current protection.

The regulators have an over-current protection which, when current drawn exceeds designed value, operates the regulator in constant current mode. Such a state of regulator is signalled by a level on one of the pins. This signal is latched by board logic and can be read-out by ELMB and if enabled will switch the affected channel off.

(6) Enabling and disabling individual output lines(on/off).

The DTMROC has an output which via the proper command can be set true or false. This output is connected to the inhibit input of the pertinent regulator thus providing a mean to switch the regulator output on/off. The same input is as well accepting the signal from over-current-monitor circuitry.

The MARATON power supplies are delivered with an OPC (OLE for Process Control) server package which serves for the remote control of the unit. The server operates via Ethernet. The MARATON system has been included into FRAMEWORK which makes its integration very easy. Figure 3.6 and Figure 3.7 show typical PVSSII control panels for MARATON system which can be tailored to specific user needs.

The control commands of the MARATON system are limited. Basically the only important action which can be undertaken by user is switching the given output on or off. Such an action is triggered by clicking ON or OFF buttons on the panel.

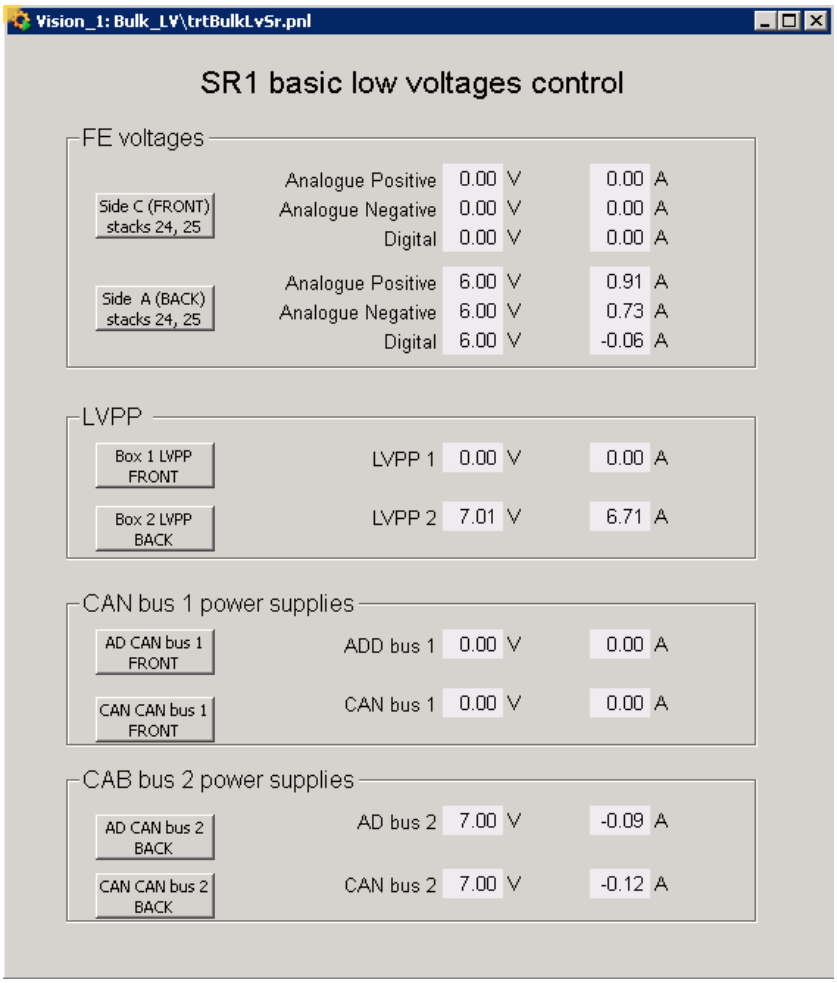


Figure 3.6: Custom GUI for main power control

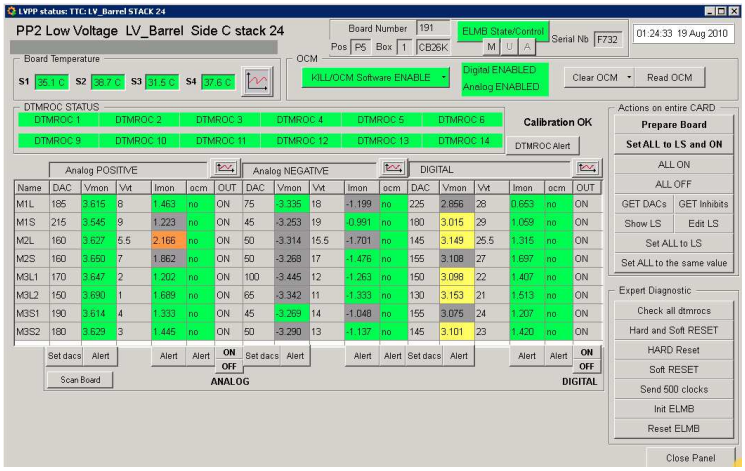


Figure 3.7: Custom GUI for FE power control

3.6 TRT Hardware Interlock (HWI)

3.6.1 Description of the HWI

Backup protection against overheating of the TRT front-end electronics, beyond that provided by the DCS, is guaranteed by the TRT Hardware Interlock (TRT-HWI). This system of custom electronics monitors temperatures on the front-end boards read out using Negative Temperature Coefficient Thermistors (NTC), which also provide inputs to the DCS system. If a specified number of NTCs register temperatures above pre-determined thresholds for a user-adjustable period of time, the system will switch off the Low Voltage (and also High Voltage, if desired) to the entire TRT. In addition, auxiliary inputs to the TRT-HWI are also provided to allow other conditions (for example, the proposed DCS watchdog status) to cut power to the TRT. The TRT-HWI is designed to be as fail-safe as possible: using no programmable logic or remotely setable parameters, while being monitored but not controlled, by the DCS.

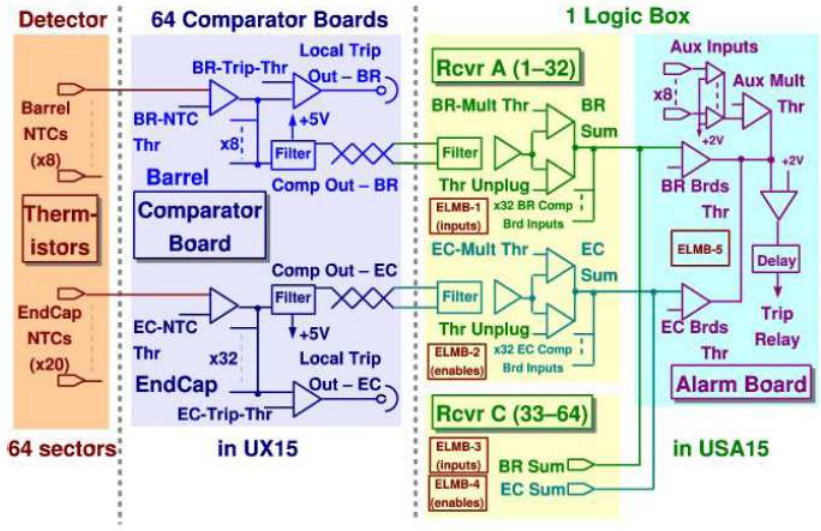


Figure 3.8: Block diagram of the TRT-HWI system.

A block diagram of the major elements of the TRT-HWI is shown in Fig. 3.8. The system itself is comprised of two elements:

(1) *Comparator Boards*, sitting on the PP2 boxes in SR1, which compare NTC outputs in 1/32 of each end of the detector to pre-determined thresholds and send out current approximately proportional to the number of NTCs over threshold in the barrel and the endcap

(2) *the Logic Box*, located in SR1, which is itself composed of two elements: *Receiver Boards* that receive all of the outputs of the Comparator Boards for the

A and C sides of the detector, and "count" Comparator Boards with more than a specified number of NTCs over threshold; and an *Alarm Board* that uses the "counts" from each Receiver Board, as well as auxiliary inputs to generate a KILL signal based on a simple algorithm.

3.6.2 Comparator Boards

Each of the 64 Comparator Board receives input from a total of 28 NTCs mounted on TRT front-end electronics boards: eight from the barrel and 20 from the endcaps. These NTC signals are routed to the Comparator Boards from the TTC PP2 boards where they are passively split off from the main TTC DCS path. On the Comparator Boards, the output signal from the NTCs is compared to thresholds using rad-tolerant comparators (LM339). The input to the comparators from the NTCs is pulled down to ground, so that both shorted and disconnected NTCs appear, to the comparator, to be in a high-temperature state, thus potentially causing a KILL signal to be generated.

Thresholds for all barrel and endcap NTC comparators can be set independently using two socketed resistors on the Comparator Board. This scheme allows the thresholds to be adjusted if conditions change, but prevents them from being set remotely to incorrect values. The state of each comparator (above or below threshold) is indicated by an LED.

The current outputs of all eight barrel comparators and all 20 endcap comparators are summed separately to form the two primary board outputs: BR_SUM and EC_SUM. These sums contain a small current offset to allow disconnected cables to be detected at the Logic Box.

If a given NTC is known to be bad, its comparator output can be individually removed from the sum (disabled) using a set of switches mounted on the board.

The BR_SUM and EC_SUM outputs can be used in two ways, although both methods will not be used simultaneously.

(1) They can be sent to the Logic Box for use in constructing the global KILL signal.

(2) They can be sent, via another set of comparators, to disable inputs on the PP2 MARATON power supplies (LOCAL_KILL).

Outputs to the Logic Box are sent differentially using two wire pairs each (for BR_SUM and EC_SUM) on $\sim$100 m ethernet-style cables. The alternate LOCAL_KILL signal, which is used during commissioning, is generated by two other comparators. The threshold for each of these can be set using socketed resistors. Two LOCAL_KILL signals (for barrel and end-cap) are sent out through LEMO connectors. The end-cap

signal is split with a simple “T” to accommodate the two end-cap LV boards in a PP2 crate.

The Comparator Boards, which are fabricated as 75×125 mm, four layer PCBs, are collected at each PP2 location into mechanical housings, which can hold five or eight boards. This housing is screwed onto the side of the PP2 structure. The Comparator Boards receive power, via a daisychain connector, from the same LV supply used to power the PP2 TTCs.

3.6.3 The Logic Box

Comparator Board output signals are received by instrumentation amplifiers on one of two Receiver Boards that are part of the Logic Box: one of which deals with signals from side A, the other for side C. After reception each Comparator Board output sum serves as input to two comparators. One comparator detects whether the cable from that Comparator Board is disconnected, and the other compares the signal to a multiplicity threshold, set separately, by socketed resistors, for barrel and endcap sums. The results of the two comparators for each Comparator Board input are summed to give two output signals, `BR_MULT_SUM` and `EC_MULT_SUM` corresponding to the number of Comparator Boards in the barrel and end-cap that have more NTCs over threshold than that specified by the multiplicity threshold.

Individual Comparator Board inputs can be excluded from this sum (disabled) using switches on the Receiver Board. Because this action would leave a significant portion of the electronics unprotected, several levels of monitoring have been added to the Receiver Boards including LED visual indicators and independent monitoring by DCS to ensure that disabled boards do not go unnoticed.)

The `BR_MULT_SUM` and `EC_MULT_SUM` signals from each Receiver Board are daisy chained (effectively summing them from each board) and sent by a short ribbon cable to the Alarm Board, where they are compared against barrel and end-cap board-count thresholds, again set by socketed resistors. If any of the barrel, end-cap, or auxiliary inputs are over their respective thresholds, an `ALARM` condition is generated. To reduce sensitivity to transient noise in the system, this `ALARM` condition is only translated into the `KILL` signal if it persists for a specified period of time (set by an RC circuit, and currently ∼45 s). The `KILL` signal activates a relay, which produces the signal to turn off power to the TRT, via a cable to the LV (and possibly HV) racks in SR1.

Critical voltages and signals (the various `_SUM`, `ALARM`, `KILL`, `DISABLED`, etc.) are monitored by a set of five ELMBs. One ELMB monitors the Alarm Board states

and settings and for each side (A and C) of the detector. Two ELMBs monitor the _SUM voltages and disabled states of the 32 sectors. A PVSS panel integrated into the DCS framework monitors and logs all abnormal conditions.

In order to be able to power up the TRT after a power failure or a trip, it is necessary to temporarily disable the TRT-HWI. This reset can be accomplished in two ways. First (and normally) a push-button disable is provided on the front of the Alarm Board. When pressed, the button disables the KILL output signal for approximately five minutes, after which time the signal is automatically re-enabled and can again cause TRT power to be cut. For those occasions when a longer period of inactivity is desired, a "hidden" switch is provided that disables KILL until the switch is returned to its enabled position.

3.7 Trigger

The Trigger system uses criteria to decide if a cosmic particle is passing through the detector. Trigger system is necessary because of the limitation in data storage capacity. Data is recorded only when the DAQ is triggered by a passing particle. To trigger cosmics the setup uses two different triggers; a scintillator trigger and a Fast-OR trigger.

3.7.1 Scintillator Trigger

Scintillators are used to trigger on cosmic muons that cross the test stand. Trigger is setup as shown in Figure 3.9 to get hits from both the barrel and the endcap components of the setup. Scintillators are aligned so that the cosmic particle crossing the scintillators also crosses the detector. When the trigger signal arrives to electronics data taking starts.

Coincidence Measuring

The Scintillator trigger electronics gets signal from both scintillators and tries to find the coincidence of these two signals. When the coincidence occurs trigger signal is sent to data acquisition system for data taking. Figure 3.10 shows the electronic setup used for scintillator trigger system.

The operating voltage for the first scintillator is U_1 = 2350 V, where rate is f_1 = 200 Hz respectively for the second scintillator U_2 = 2500 V, rate f_2= 400 Hz.
The pulse width is the same for both scintillators which is $t_{1,2}$ = 100 ns.

The probability of "hitting" a pulse from scintillator 1 within one second is
$200Hz \times 100ns/1sec = 2/100000$.
The probability of a random coincidence is then

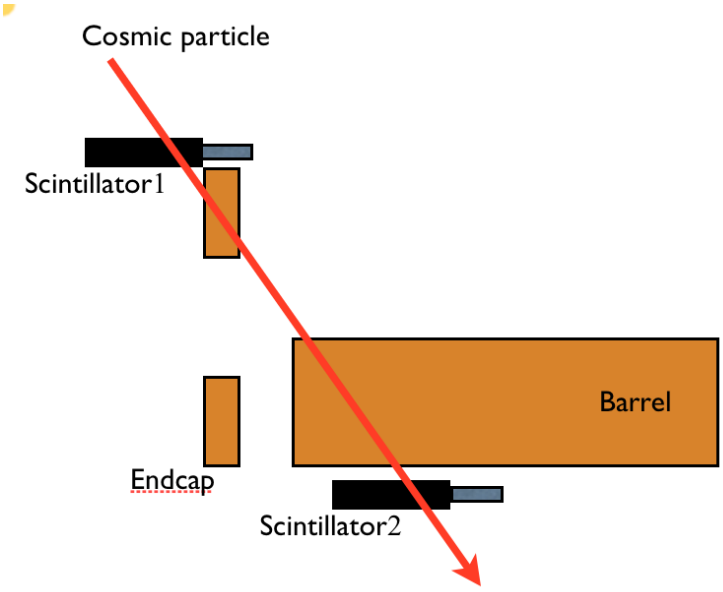


Figure 3.9: Layout of the scintillators used in the TRT SR1 setup

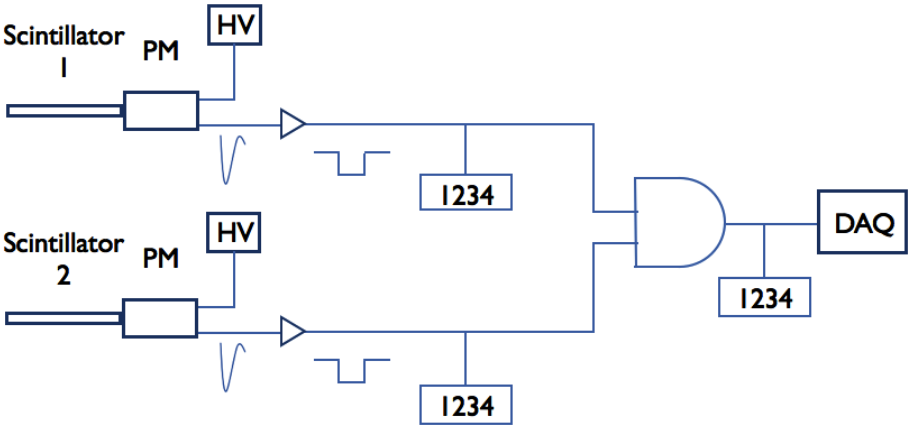


Figure 3.10: Trigger electronics TRT SR1

$400Hz \times 2/100000 = 8/1000 = \sim 1\ \%$.
Where threshold levels of the discriminators are $Thr_2 = Thr_1 = 0.310$ V.

3.7.2 TRT Fast-OR Trigger

The TRT "Fast-OR" trigger [5] is the highest track rate cosmics trigger of the Inner Detector at ATLAS. It utilizes a fast trigger generation circuit on the front-end DTMROCs and a simple trigger logic on the TRT trigger, timing and control board. We set up Fast-OR trigger for the detector in SR1 in the same fashion as done in the ATLAS

detector.

When the LHC incident in September 2008 promised an extension of the commissioning period using cosmics, the decision has been taken to finalize the implementation of a TRT trigger. It allowed the TRT to collect tracks from cosmics independent from other subsystems, with rates in both barrel and endcap that were significantly higher than what other triggers had been able to produce.

A Fast-OR circuit on the DTMROC permits sending a trigger signal to the TTC board if it receives a discriminated straw signal from any of the 16 associated readout channels. The TTC back-end receives trigger signals from groups of 10-15 DTMROCs over a communication line usually used for configuration and sense data transmission. Independent logic circuits on each of the 16 TTC boards in the barrel system can generate a trigger signal if the number of communication lines that carry a signal within a clock cycle (25 ns) exceeds a configurable number. To ensure a high fraction of hits on track as well as very low noise, the trigger electronics on the DTMROC was configured to generate a signal from the high threshold that was lowered to minimum ionizing particle levels. Each TTC board was configured to send a trigger signal if at least 4 communication lines carry a signal. This parameter can be changed according to needs. Detector in SR1 has only two stacks so this parameter set as 3 to have it working well.

As minor disadvantages, this configuration makes TR calibration difficult and is not compatible with configuration or sense data transmission. More than 90% of the triggers arrive in one clock cycle, It helps improve detector readout timing. The trigger rate is $\sim 10Hz$ and the efficiency of collect- ing cosmics tracks is $\sim 75\%$.

The logic of Fast-OR trigger implemented on TTC booard shown in Figure 3.11. The number of TTC lines (MTPC) that are enabled (IN SEL) in the trigger configuration and have a trigger pulse within the window WIN is compared (CMP) against a configurable multiplicity (TOTAL MTPC for the total group). A subset of those lines (GROUPN SEL for group N) can be selected to be compared to the multiplicity of group N (GROUPN MTPC). The trigger decision of each group can be selected to be in (anti-)coincidence with the result of the total group. The TRIGGER output signal is a NIM signal that is sent either to the NIM modules, or to another TTC board as input signal (FAST INPUT FROM PREVIOUS TTC) that can be selected to be in (anti)coincidence for a more sophisticated track selection.

3.8 Active Gas System and Cooling System

3.8.1 Detector Environmental Gas

The TRT detector requires an overall CO_2 envelop gas acting as a barrier between the active part of the detector and the environment, to prevent contamination from nitro-

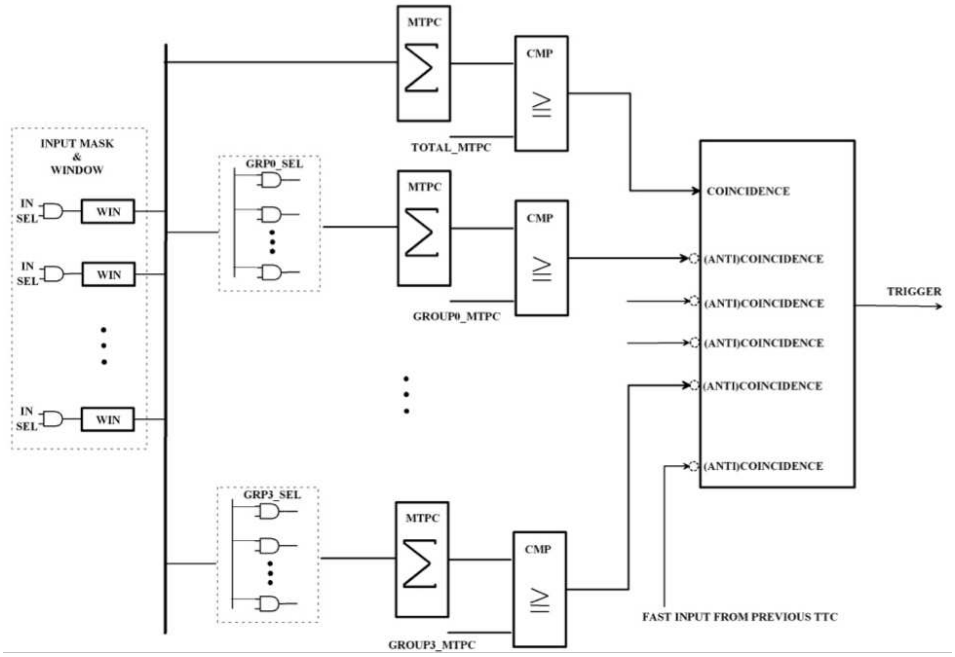


Figure 3.11: *The Fast-OR logic as implemented on a TTC board.*

gen, possibly other sources of pollution, high-voltage discharges and the accumulation of xenon due to possible gas leaks which would absorb the transition radiation photons. The module shell serves as a gas manifold for CO_2.

3.8.2 Active Gas System

The active gas for the straw drift tubes is a mixture of Xe(70%), CO_2(27%), O_2(3%). Now for testing and adjustment the mixture Ar(70%) and CO_2(30%) is used. It allows to see the tracks but without drift-time measurements. This gas is supplied through two lines (one for Barrel, the latter for End-cap) and exits into the atmosphere. In the future, when Xe is used in the mixture, the exit gas will be recovered, cleaned and recirculated.

A schematic view of the SR1 TRT setup gas circuit is shown in Figure 3.12. The flow of active gas mixture should be about 20 liters when the setup is used and about 10 liters when it's stopped.

In Figure 3.13 rack panel of the active gas system is shown. The valve on the left side is used for Barrel active gas and Right side for Endcap active gas.

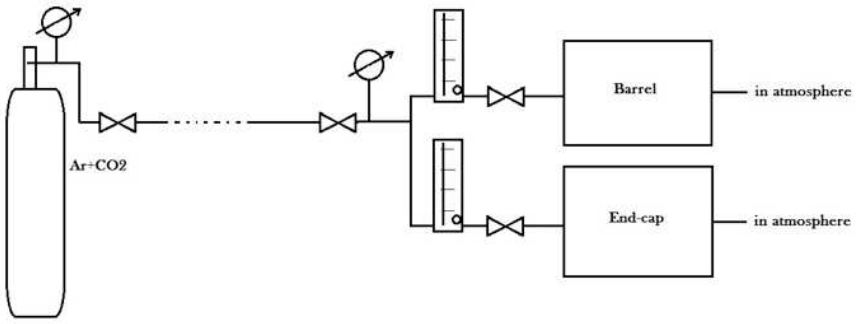


Figure 3.12: *Active gas circuit*.

Figure 3.13: *Active gas system rack panel*.

3.8.3 Cooling System

Cooling system is one of the important systems of the detector. Cooling system is necessary because during the operation of the detector electronics get hot even it can burn. To provide cooling for detector electronics the Freon Cooling System was installed. In Figure 3.14 the scheme of the Freon distribution system is shown.

To control flow of the cooling system valves on the rack panel is used. In Figure 3.15 the rack panel of cooling system is shown. Normal flow in the cooling pipes is about 4.3 digits of the flow controller that means approximately 1.6 l/min.

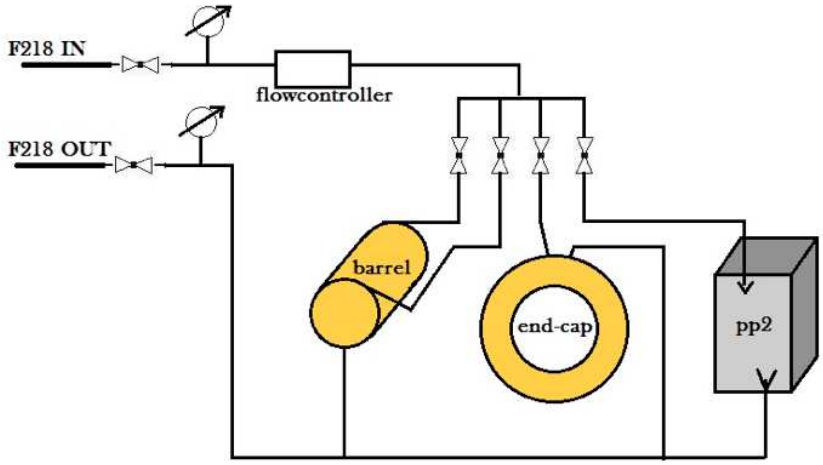


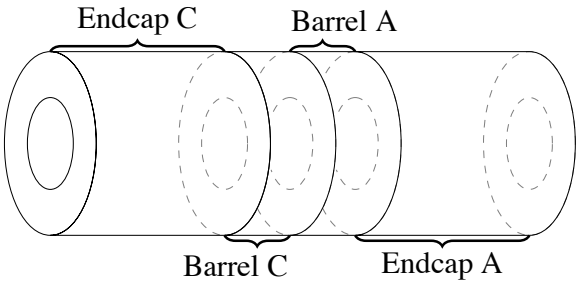
Figure 3.14: *Schematic of cooling system*.

Figure 3.15: *Cooling system rack panel*.

3.9 Detector Readout

The detector is composed of straws, placed to provide hits on a track for charged particles passing through the detector.

The detector can then be thought of as being three large pieces: one *barrel* part which consists of two stacks and two *endcap* wheels. The barrel is further divided into two pieces, as each straw is read out at both ends. Thus, in the readout system, there are four large pieces, or *partitions*: Endcap A, Barrel A, Barrel C, and Endcap C [14].



The custom TRT read out chain begins with the Timing and Trigger Controller, or TTC, an off-detector VME module. The TRT TTC is responsible for steering the front end (on detector) electronics: setting configurations, providing reset signals, and sending triggers. Those signals are sent out over TTC *lines* to a *patch panel*, where the signals are distributed to the front end. At the front end, the straws are organized in groups of 16, and each group is associated with a single digital time measurement read-out chip (DTMROC). When a trigger is received, the DTMROC packages the data from these 16 straws, and sends it to the off-detector electronics.

After leaving the DTMROCs, the data first reach a different patch panel, this one associated with the TRT read out driver (ROD). The ROD is responsible for packaging the data from a large group of DTMROCs into a single buffer that will be collected by some software process. The patch panel associated with the ROD receives DTMROC data and transmits it, fiber-optically, to the ROD.

The ROD will package the data for some number of DTMROCs , along with some header and trailer words, and send those data to whatever software process requests them. When the DAQ is collecting ROD data, the data can be acquired via VME, or over the S-Links by a read out system (ROS). A ROD refers to a *logical* rod, not a *physical* ROD (9U VME card). There are two logical RODs per physical ROD; those two logical RODs share some resources, but from a readout standpoint they are treated separately.

3.10 Front-End Electronics and Data Structure

3.10.1 Straw Tracker Basics

When a sufficiently energetic charged particle travels through a gas, coulomb interactions between that particle and the valence electrons of the gas will result in the liberation of some of those electrons from their respective nuclei. By placing a wire down the middle of a straw filled with such a gas, and holding the straw wall at (negative) high voltage, any liberated electrons will be collected by the wire. The resulting current can be detected by sufficiently sensitive electronics at the end of the wire. In the case of the TRT, those electronics amplify and shape the current pulse, and apply a threshold to the resulting pulse shape to determine if it should be digitized.

The electronics developed to perform this current collection must further account for the positively charged gas ions which will drift *away* from the wire, and which produce a mirror current on the wire. Unlike the current produced by the electrons, which begins and ends quickly, the current from the ion drift takes a significant amount of time to fade away. To be able to read out at high trigger rates, the electronics compensate for this long tail by restoring the baseline very quickly.

3.10.2 Analog Readout

The TRT solution to the analog electronics problem is a custom designed ASIC developed to do amplification, shaping, discrimination, and base-line restoration called ASDBLR [9]. The ASDBLR is capable of detecting signals at two different thresholds: the first, a tracking threshold, discriminates near 200 eV; the second, the transition radiation threshold, is much higher, closer to 5 keV. The output from the ASDBLR is ternary - a "0" level if no threshold has been crossed, a "1" level if either the low or the high threshold has been crossed, and a "2" level if both low and high thresholds have been crossed. Each ASDBLR monitors up to eight straws, producing output for each of those straws and passing it directly to the corresponding DTMROC.

3.10.3 Digital Readout

The DTMROC supports several different types of readout [12]. The most important mode is what is used in normal data taking: the DTMROC reports 75ns of straw data for each of 16 straws. There is also an option to have the DTMROC respond with multiple copies of its own address; this is called "send-ID mode". Finally, one can program a customized pattern into the DTMROC to be sent to the ROD; this is commonly called "test-mask mode". All three are available by setting the appropriate bits in the configuration register of the DTMROC.

Normal Physics Data

Every 25ns, the DTMROC samples the output of the ASDBLR 8 times per straw. For each time bin (3.125ns), it creates one bit which tells whether that straw was over the tracking (low level) threshold in that bin. One additional bit is used to record whether or not that straw crossed the transition radiation (high level) threshold at any time during the 25ns. This gives 9 bits per straw per 25ns.

When a trigger is received by the DTMROC, the DTMROC packages three time bins worth of data for each straw, and sends those data out to the ROD. Each ASDBLR reads out up to eight straws, and each DTMROC is responsible for two ASDBLRs, so this gives

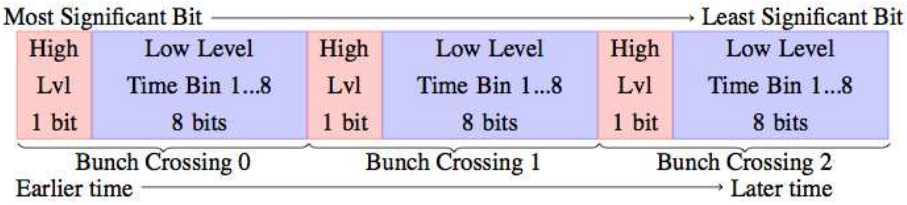


Figure 3.16: *Data structure*.

$9\frac{\text{bits/straw}}{\text{bunch}} \times 3\text{bunch} \times 16\text{straws} = 432\text{bits}$. Along with the straw data, the DTMROC also inserts a three bit preamble and a nine bit header, for a total of 444 bits.

Data	Bits	Note
Preamble	1	101
Send-ID	1	9 bit Header
L1ID	3	
BCID	4	
Error	1	
Straw 0	27	"Straw 0" = 3×9 bits(BC0, BC1, BC2)
Straw 1	27	
...	...	
Straw 15	27	

Table 3.2: *DTMROC data structure*

Each DTMROC sends these 444 bits on a single small-twisted-pair cable to a data patch panel, where it is received by a Gigabit Optical Link (GOL). The GOL transmits the data of up to 30 DTMROCs over an optical fiber to the TRT ROD.

Send-ID Data

The DTMROC can also create a data buffer composed of multiple copies of its own address. This functionality is used for testing purposes. If the chip is in "send-ID" mode, then upon receipt of a level-1 accept, the pipeline is filled with 48 copies of the chip's address, corresponding to 16 straw words each containing three copies of the address encoded in 9 bits.

So, for instance, a DTMROC with an address of 1 would have straw data that looks like 0x40201; an address of 2 would be 0x80402, an address of 3 would be 0xc0603, etc.

Test-Mask Data

The "test-mask" on the DTMROC serves two purposes: it can allow the DTMROC to ignore certain straws when sending data to the ROD, or it can output the mask itself

as the straw words. The mask is composed of 16 nine bit words. When in test mode, the first 9 bits (repeated 3 times, to make a 27 bit straw word) will appear as the first straw word, the second 9 bits will appear as the next straw word, and so on

This functionality is especially useful in diagnostic tools and automatic mapping methods. By programming the DTMROC to report back its own address, the TTC line, and so on, one can see very quickly which DTMROCs end up in each ROD channel.

3.10.4 Modularity of the Front-End Electronics

As described above, the DTMROC is responsible for 16 straw channels, the data from which are reported to the ROD for further packaging. The DTMROCs and ASDBLRs are mounted on printed circuit boards in a variety of patterns, depending on their physical location on the detector.

Endcap

The straws are mounted radially in the endcaps, and the electronics sit at constant R, on the outside of the endcap. The DTMROCs are mounted on square boards in groups of four, three of which boards are connected together to form an *endcap triplet*. The triplet has a single connection for TTC and ROD signals, and for low voltage. Each triplet forms one 32nd (in ϕ) of an endcap wheel. There are then 12 A-wheel triplets and 8 B-wheel triplets that form one 32nd of one endcap. So, in one 32nd of one endcap there are 240 DTMROCs.

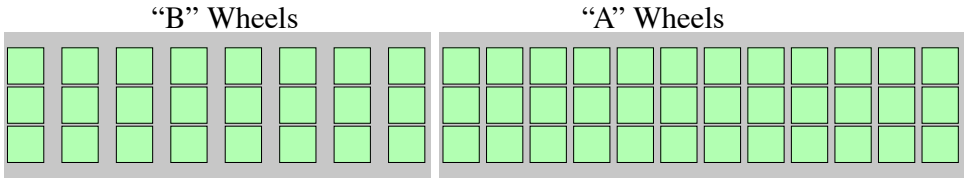


Figure 3.17: *One 32nd of a* TRT *endcap*

The ASDBLRs are mounted on separate boards that are connected to the bottom of the triplets, and are then connected to the straws via kapton flaps.

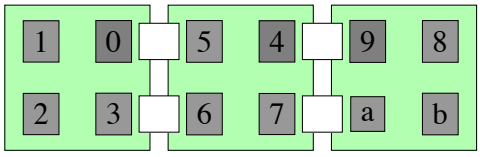


Figure 3.18: *Endcap triplet*

Barrel

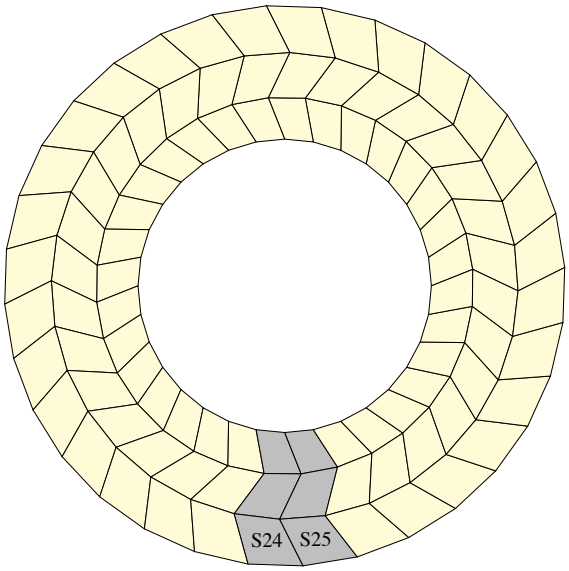


Figure 3.19: *Barrel layout*

On the barrel, the situation is slightly more complicated. The straws on the barrel are arranged axially, and the readout electronics sit on either side of the barrel. The straws were built into one of three kinds of modules, depending on their radius: modules of type 1 sit at smallest R; modules of type 2 sit outside of type 1's; and modules of type 3 sit at largest R. Each module is a trapezoidal prizm - there are 32 modules of each type that make up the barrel.

The Figure 3.19 represents the two barrel sectors used in the test stand, sector 24 and sector 25. Each sector has a collection of boxes which contain TTC, ROD, and low voltage patch panels. A single patch panel box will contain enough patch panels for one half of one 32nd of the detector: this means three low voltage patch panels, two TTC patch panels, and three ROD patch panels.

At either end of the barrel modules, two triangular circuit boards are mounted, distinguished by their size relative to each other. Because of the module shape, the two triangles will not have equal area - thus, there are small ("S") and large("L") boards for each module type.

These circuit boards contain both the ASDBLRs and the DTMROCs. On the both side ASDBLRs lie on the bottom part of the board, closer to the straws. They are connected to the straws through a series of pins that take signals between the *tension plate*, a plate that holds the straw wires at high tension, and the ASD inputs. The DTMROCs are mounted on the top of the board. Each triangular board can be associated with either one or two TTC lines, depending on its type: type 1 boards and the small type 2 boards

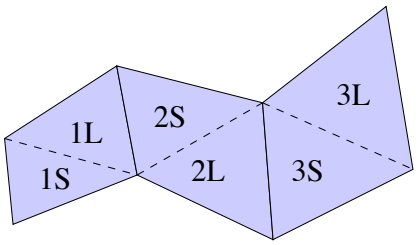


Figure 3.20: *Barrel Module*

have only one TTC line each, while all type 3 boards and the large type two board each have two TTC lines.

The order of the DTMROCs on a board is done in a snake-like pattern: the lowest address is the chip closest to the outside corner of the module, with addresses increasing sequentially for most boards.

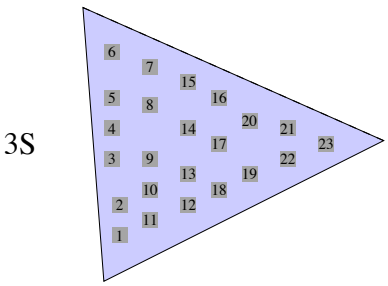


Figure 3.21: *DTMROCs on a board*

For the type 2 and type 3 boards, a single triangle board can have two TTC lines associated with it; the division between the two lines is made as close to half-way through the board as possible. The two halves are referred to as either "1" and "2", or "Q"(uadrilateral) and "T"(riangle). Much of the DAQ software tends to use the former, while the integration software will tend to use the latter.

3.11 ROD Data Structure

Each logical TRT ROD is responsible for collecting straw data from some fraction of its TTC partition. For the barrel, this means 1/32nd of one side, or 104 DTMROCs; for each endcap, this means 1/64th, or 120 DTMROCs.

Each DTMROC buffer is received by the ROD, on one of four serialized optical lines (which handle the data for roughly 30 DTMROCs each). After this point, the ROD identifies DTMROCs by an address, from 0 to 119, corresponding to the input line for that DTMROC.

Side	Board Name	Number of DTMROCs	Addresses
A	AR1BS	10	0x01...0x0a
	AR1BL	11	0x11...0x1b
	AR2BS	15	0x01...0x0f
	AR2BL1	9	0x11...0x19
	AR2BL2	9	0x1a...0x22
	AR3BS1	11	0x01...0x0b
	AR3BS2	12	0x0c...0x17
	AR3BL1	13	0x21...0x2d
	AR3BL2	14	0x2e...0x3b
C	AR1FS	10	0x01...0x0a
	AR1FL	11	0x11...0x1b
	AR2FS	15	0x01...0x0f
	AR2FL1	9	0x11...0x19
	AR2FL2	9	0x1a...0x20,0x31,0x32
	AR3FS1	11	0x01...0x0b
	AR3FS2	12	0x0c...0x17
	AR3FL1	13	0x21...0x2d
	AR3FL2	14	0x2e...0x3b
A & C	All Endcap Boards	12	0x00...0x0b

Table 3.3: Front end board DTMROC configuration. The addresses of the DTMROCs are used by the TTC when communicating with a particular chip.

Once the data are received by the ROD, the ROD packages the data into a single output buffer. The packaging of the straw data can take three forms: *uncompressed*, *minimally compressed*, and *fully compressed*. (For technical details on the compression scheme, see [9].) Whichever form the straw data take within the output buffer, they are wrapped by additional information inserted by the ROD, in the form of a header, a trailer, and a variable-width error block.

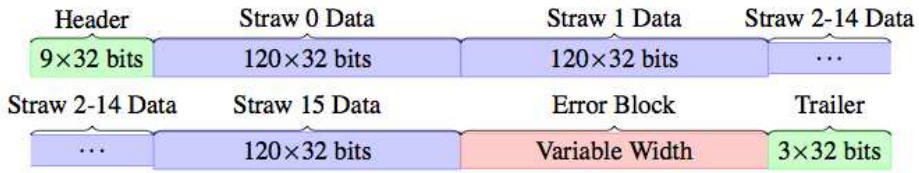


Figure 3.22: *ROD data structure*.

3.11.1 Header and Trailer Blocks

The format for the header and trailer blocks in the ROD fragments are defined by the ATLAS Event Format [10]. The full contents of the blocks are detailed in table 3.4.

3.11.2 Uncompressed Straw Data

Each 32 bit word contains the 27 bit straw pattern in the lowest bits, and the upper 5 bits are set to '1's or '0's. The data from the DTMROCs are handled in a parallel fashion, and then serialized into a single block of 1920 32-bit words.

3.11.3 Error Block

After the straw data, the ROD can insert up to 8 "scratch" words. These scratch words can be used, for instance, to mark an event during a calibration run as corresponding to a certain step in a scan. However, they have no inherent meaning, and therefore are skipped by most byte-stream converters if they aren't expected. Preceding these scratch words will be a counter, indicating the number of scratch words that follow it, plus one (to count itself).

Following the scratch words is the error block. The error block is of variable length, and contains any DTMROC headers which are different than what the ROD expects. The ROD checks the BCID and L1ID to see that they match the numbers in the ROD header, and checks the generalized error bit to see if it's active. If the BCID or L1ID is incorrect, or if the error bit is active, then the 9 bit DTMROC header is packaged with a 7 bit address (this can be 0-119, and does *not* correspond to the output order of the

Index within buffer	Example word	Meaning	Changes event to event?
ROD Header			
0	0xee1234ee	ROD header marker	No
1	0x00000009	Number of header words	No
2	0x03000001	Format version number	No
3	0x00310100	ROD identifier	No
4	0x76543210	Run number	No
5	0x61000001	Extended L1ID	Yes
6	0x000003ab	BCID	Yes
7	0x00000000	Trigger Type	Yes
8	0x87654321	Detector Event Type	Yes
ROD Trailer			
N-2	0x0000003d	Size of error block (in words)	Yes
N-1	0x00000780	Size of data block	Yes
N	0x00000001	Error block position indicator	No

Table 3.4: The meaning of the different words in the TRT ROD header and trailer, for a ROD buffer of N words. The example words are examples only, though some of them (the header marker, 0xee1234ee, for instance) will always appear as shown here.

Status Word = 0x8e119e8f					
DTMROC Position	9-bit Header	Send-ID On?	L1ID	BCID	Error On?
0x11	0x11c	Yes	0	0xe	No
0x0f	0x13d	Yes	1	0xe	Yes

Table 3.5: An example decryption of a status block element that reports DTMROC header words. A single 32 bit word (here 0x8e119e8f) contains 16 bits for each of two DTMROCs - the least significant 7 bits represent the position of the DTMROC in the ROD output buffer (0 to 119), while the most significant 9 bits reproduce the DTMROC header as received by the ROD.

straw data) and inserted into the error block as half of a 32 bit word. For an example decoding, see table 3.5.

3.12 Back End to Detector Mapping

The detector mapping shows how the electronic components of the detector connected. The full mapping for one half of one 32nd of the detector is shown in Figure 3.23 for both Endcap and Barrel.

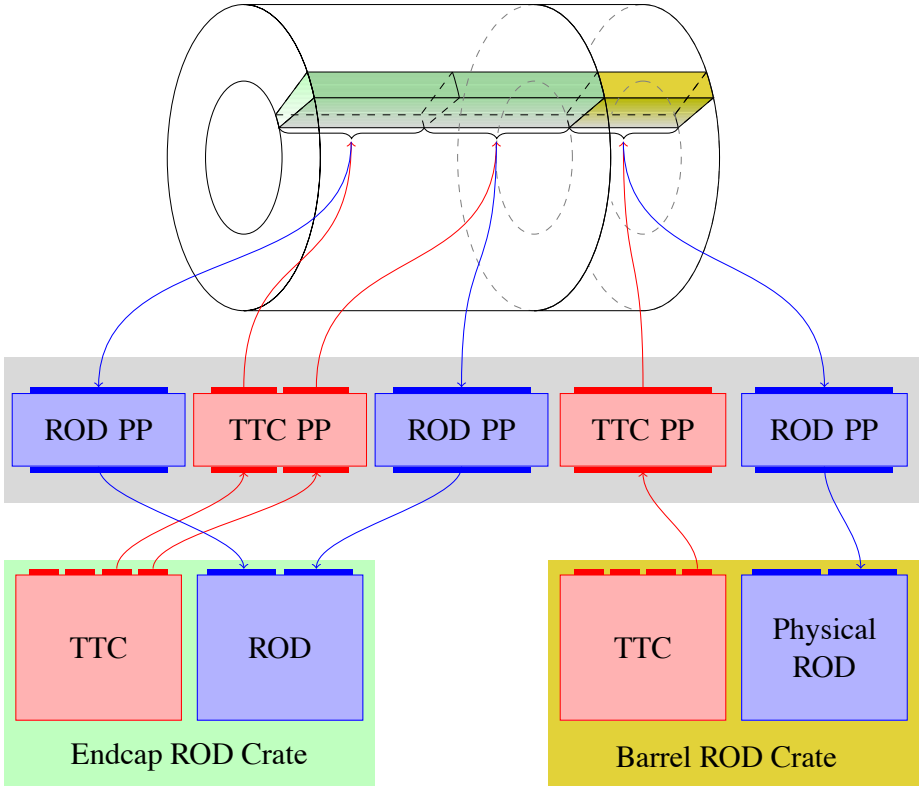


Figure 3.23: *Mapping for one half of one 32$_{nd}$ of the detector*

The mapping of the detector in Figure 3.23 has following properties;

- The gray box represents a single patch panel box; this box will hold the five patch panels shown, as well as three low voltage patch panels (two for the endcaps, one for the barrel).
- The ROD patch panels (and passive boards) are identical in every way between barrel and endcap.
- The TTC patch panels differ between barrel and endcap in two ways:

  (1) The passive boards have five connectors for endcap type 2 cables, three connectors for barrel type 2 cables;

(2) The polarity of the reset signal is reversed for barrel and endcap boards - this change is made via a jumper on the patch panel.

- Each endcap ROD-crate will contain up to 6 TTCs; each barrel ROD-crate will contain 4 TTCs.
- For detail on the connection of type-II cables within a PP2 box, see the diagram on page 48.

3.12.1 Hardware Identifiers

To identify all of the configurable components in the DAQ, unique identifiers have been created for each component. The conventions are consistent within a type of component, but vary across different components.

VME Addresses

Before going through the various hardware identifiers, a note on VME addresses. For the ROD crates, VME addresses are assigned geographically, i.e. by slot number. Each module is given a VME address based on its slot number - the RODs have addresses 0x01000000$\times$(slot), while the TTCs have addresses 0x08000000$\times$(slot)

In the TTC crates, the addresses are assigned using dials mounted on the circuit boards. The scheme used for assigning VME addresses in the TTC crates (which serve as identifiers for those modules) is given below.

TTC Identfier Convention

The TTC identifiers are created from their location in SR1, i.e. their crate number and slot number, as well as their partition. A TTC identifier will have the format 0xXXYYZZ, where:

XX = 35 for Barrel A
36 for Barrel C
37 for Endcap A
38 for Endcap C

YY = Crate number, from 01-04 for barrel, 01-06 for endcap

ZZ = Slot number (hex)

ROD Identifier Convention

The ROD identifiers are created from their partition number and detector sector being read out. The identifier will have the format 0xXXYYZZ, where:

XX = 31 for Barrel A
32 for Barrel C
33 for Endcap A
34 for Endcap C

YY = (Phi) Sector number, 1-20 (hex)

ZZ = 00 for the barrel
01 for low $|z|$ Endcap wheels
02 for high $|z|$ Endcap wheels

For the endcap RODs, 0xXXYY01 corresponds to the top half of the ROD, and 0xXXYY02 corresponds to the bottom half of the ROD.

DTMROC Identifiers

The DTMROC identifier is based on its partition, phi sector, and hardware address. The formula is:

Chip ID = 0x10000×(Partition Number) + 0x100×(Phi Sector) + (counter from 0 to 0x67 for barrel, 0 to 0xEF for endcap)

Where the partition number is 0x31 for Barrel A, 0x32 for Barrel C, 0x33 for Endcap A, and 0x34 for Endcap C.

ROB Identifiers

The mapping of RODs to ROBs is 1:1, so the ROB ID for a given ROD will be identical to that ROD's ID.

LTP Addresses

The LTP VME addresses have the form **0xYZ0000**, where:

Y = E for Endcap
B for Barrel

Z = A for Side A
C for Side C

TTCvi Addresses

The TTCvi VME addresses have the form **0xYZ0100**, where

Y = E for Endcap
B for Barrel

Z = A for Side A
C for Side C

ROD Busy Addresses

The ROD Busy Module VME addresses have the form **0xYZ0200**, where

Y = E for Endcap
B for Barrel

Z = A for Side A
C for Side C

TTC Group	TTC Line	Barrel Board	Endcap Board
3	9	—	B7
	8	1S	B8
	7	1L	B6
	6	2S	B5
	5	2L1	B4
	4	2L2	B3
	3	3S1	B2
	2	3S2	B1
	1	3L1	A61
	0	3L2	A62
2	9	—	A51
	8	1S	A52
	7	1L	A41
	6	2S	A42
	5	2L1	A31
	4	2L2	A32
	3	3S1	A21
	2	3S2	A22
	1	3L1	A11
	0	3L2	A12
1	9	—	B7
	8	1S	B8
	7	1L	B6
	6	2S	B5
	5	2L1	B4
	4	2L2	B3
	3	3S1	B2
	2	3S2	B1
	1	3L1	A61
	0	3L2	A62
0	9	—	A51
	8	1S	A52
	7	1L	A41
	6	2S	A42
	5	2L1	A31
	4	2L2	A32
	3	3S1	A21
	2	3S2	A22
	1	3L1	A11
	0	3L2	A12

Table 3.6: Mapping of TTC lines to front-end board type. A real TTC will have either all barrel or all endcap boards in the order described above. When looking at a physical TTC, one counts the connectors from the bottom, starting from 0, and the connector number is the same as the group number. To refer to a specific TTC line in the hardware library, multiply the group by 10 and add it to the line number; eg. board A61 in group 1 would be TTC line 11.

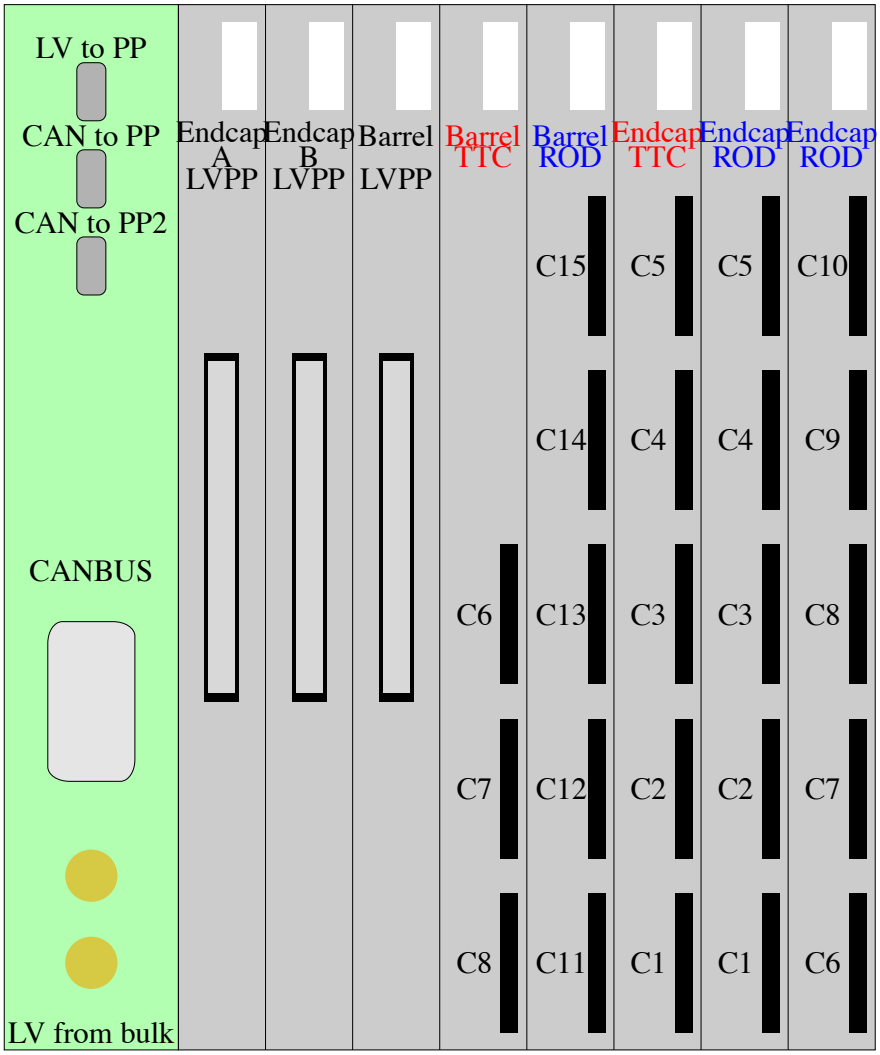


Figure 3.24: *Cable connections at PP2*

The mapping of type-II cable connections at PP2, as seen from the type-II cable side.

The mapping of connectors to the detector electronics is described in the table below.

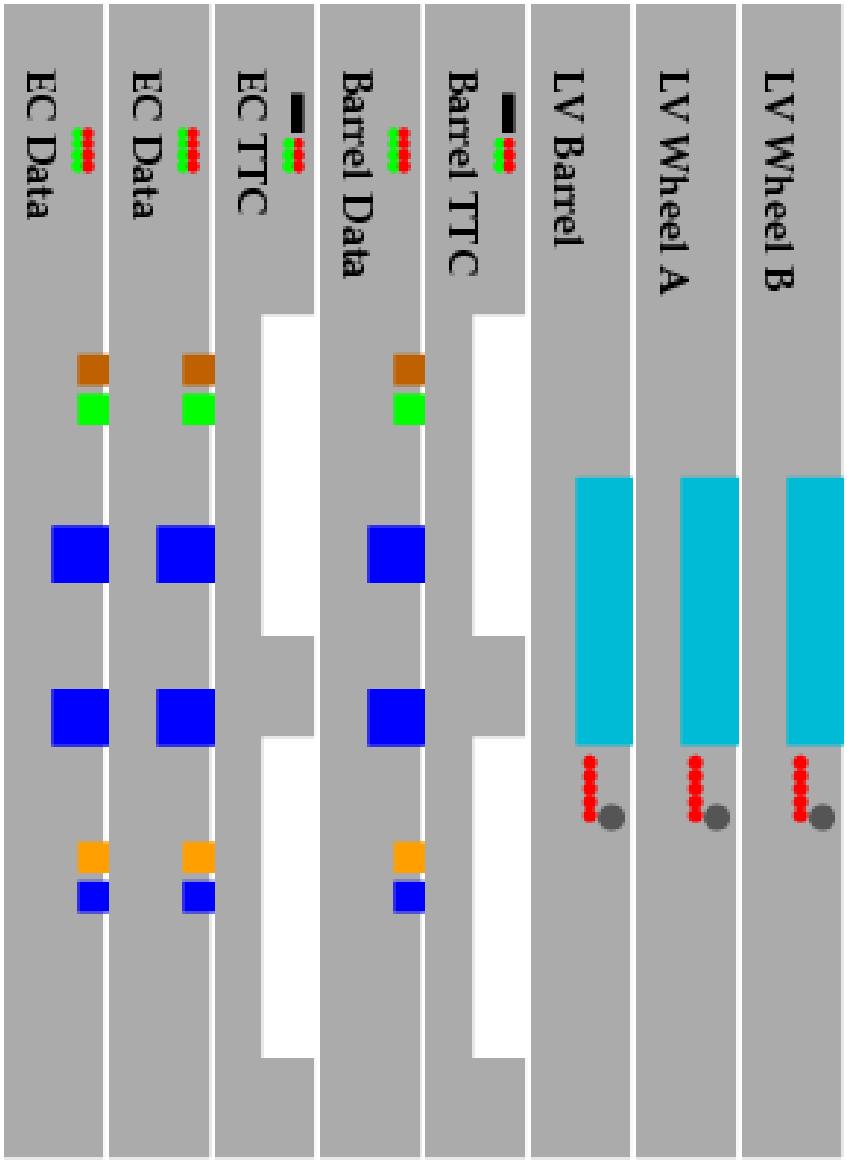


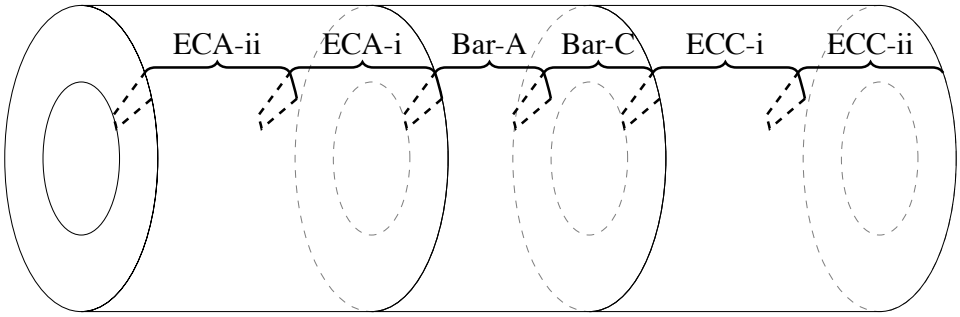
Figure 3.25: A PP2 ‘box’ seen from the type-III cable side. Each box contains three data patch panels, two TTC patch panels, and three low voltage patch panels, as shown. The data boards have two ethernet connections and four data lines. The EC TTC patch panel connects to the TTCs through two white MDR cables and also to the hardware interlock. The barrel TTC PP has only one MDR cable along with its interlock link. Each of the LV boards has one blue plug and one lemo connection to the interlock.

Detector	TTC Connector	ROD Connector	Boards
Barrel	C8	C15	AR3xL1, AR3xL2
		C14	AR3xS1, AR3xS2
	C7	C13	AR2xL1, AR2xL2
		C12	AR2xS
	C6	C11	AR1xL, AR1xS
A wheels	C1	C1	ECA12, ECA21
		C2	ECA11, ECA22
	C2	C3	ECA32, ECA41
		C4	ECA31, ECA42
	C3	C5	ECA52, ECA61
		C6	ECA51, ECA62
B wheels	C4	C7	ECB1, ECB4
		C8	ECB2, ECB3
	C5	C9	ECB5, ECB7
		C10	ECB6, ECB8

Table 3.7: Mapping of the connectors connected to boards

3.13 ROD to Detector Mapping

The TRT-RODs have been built to handle a maximum of 120 DTMROC inputs. This was motivated by a target granularity; 2 logical RODs per endcap 32nd, and 2 logical RODs per barrel 32nd. This means 6 logical RODs per TRT 32nd.



3.14 TTC Granularity

Each TRT TTC can control 40 front-end boards, or 40 TTC *lines*. In the endcap, a group of 10 TTC lines corresponds to one logical ROD; in the barrel, a group of 9 TTC lines corresponds to a logical ROD (in the barrel, 4 TTC lines will not be used). The TTC has 4 collections of 10 TTC lines, so a single TTC will be associated with 4 logical RODs.

3.15 Aligning the Readout for Particles

Having configured system for stable operation is clearly not sufficient for the system to be stable - the data produced by the system need to be useful as well. This section deals with the use of the various timing parameters to capture the necessary data for effective tracking.

3.15.1 Physics

As explained in section 3.10.1, when a charged particle passes through the TRT straws, Coulomb interactions between the charged particle and the valence electrons in the gas will result in the liberation of some of those electrons from their respective nuclei.

In the detector, a wire is strung down the middle of the straw, the outer wall of which acts as a cathode. The wire then attracts the free electrons and forces the ionized gas atoms towards the cathode (the edge of the straw).

If the electrons are liberated close to the wire, the current in the wire from the electrons will be seen almost immediately. Electrons that are freed closer to the cathode will have a drift time that depends on the type of gas used - for a Xenon/CO_2 mixture, this drift time will be around 50ns. A particle passing through the straw will generally ionize some of the gas close to the edge of the straw, so the "trailing edge" of the bit pattern should be fixed with respect to the clock as shown in Figure 3.26.

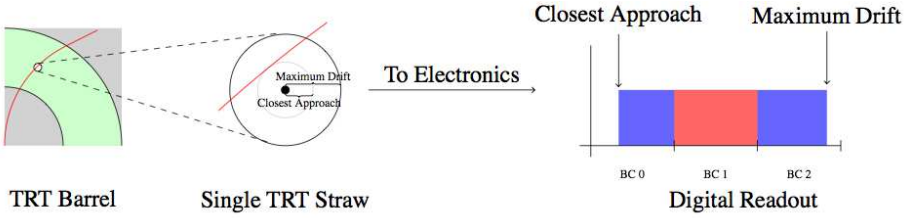


Figure 3.26: The passage of a charged particle through the straw ionizes gas, which causes an electron cascade into the anode. The resulting current is read out by the analog electronics and digitized by the DTMROC. The pulse at the right shows the result of that digitization. The blue regions are where the low threshold was crossed - the red region is where the low and the high threshold was crossed. The point of closest approach of the particle to the wire determines the leading edge of the digital pulse, while the size of the straw (and drift speed of the gas) determines the trailing edge.

In Figure 3.26, the time over threshold for the digital output is more-or-less fixed by the point of closest approach of the track to the wire. An example of the effect of moving the track further away from the wire is shown in Figure 3.27.

In order to properly reconstruct the path of the particle through the TRT, it's essential that both the leading and trailing edges are present in the readout. The maximum

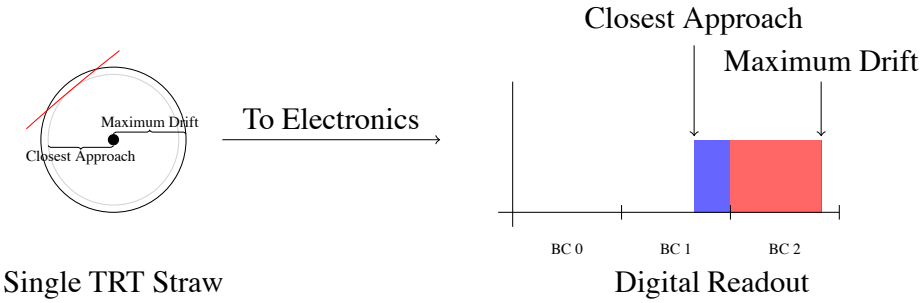


Figure 3.27: In this case, the straw is just barely hit by the charged particle, and the only gas ionized is near the cathode. The trailing edge of the distribution does not shift, as it is fixed by the size of the straw - the leading edge, however, moves later in time to reflect the drift radius of the track.

time over threshold, as seen by the digital electronics, is close to 70ns, while the full readout window for the digital chips is 75ns, so the margin for error is quite small. To properly align the data with respect to the clock, the trigger and clock will need to be shifted in time such that the passage of the particle comes at a fixed time relative to the DTMROC

To properly align the data with respect to the clock, the TRT-TTC must delay the trigger by the proper number of bunch crossings so that the data can be seen at all, and then must use the BX delay to align the data within the window of 75ns.

3.15.2 Trigger Delay

When the trigger is received by the DTMROC, it will look into its pipeline to retrieve data from a point in the buffer. The maximum depth of the pipeline is 255×25ns, and a programmable register in the DTMROC is what indicates where in that pipeline the DTMROC should look for data. The value of that register is called the *pipeline latency*, and should be equal to the number of bunch crossings elapsed between the collision to be read out and the time the trigger signal reaches the DTMROC.

In practice, the pipeline latency, which is programmable by DTMROC, is set to a uniform value for the entire detector, and a separate delay in the TRT-TTC is used to delay the transmission of the trigger by some number of bunch crossings. This delay in the TRT-TTC is applied at the level of a front-end board, which contains between 9 and 15 DTMROC's, resulting in fewer parameters to tune.

The coarse grained trigger delay in the TRT-TTC is called the TDM delay as shown in Figure 3.28.

Using the TDM delay, it is possible to tune the timing such that the majority of a track will lie within the time windows seen in Figure 3.26. However, the leading or

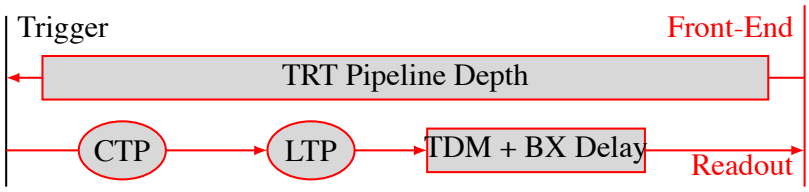


Figure 3.28: The contribution of various delays and cable lengths to finding the right timing for particles. A particle's passage is shown on the left, which generates a level 1 accept in the CTP. The trigger from the CTP travels to the subsystem LTP. In the TRT case, the trigger then goes to the TRT-TTC, where it is delayed by some number of bunch crossings such that when it arrives at the front-end, the total time between the passage of the particle and the arrival of the trigger at the front-end is equal to the pipeline latency set in the DTMROC.

trailing edge may still be missing. For this, a finer-grained delay is necessary.

3.15.3 BX Fine Delay

To tune the position of the digital pulse within the TRT readout window, steps of 25ns are too large. However, delaying the trigger in steps of less than 25ns will not shift the pulse within the time window, as the DTMROC is only aware of the trigger when it latches the trigger to the clock. The solution, then, is to change the phase of the clock such that the trigger is latched some number of nanoseconds earlier or later.

If one wants to shift the position of the digital pulse such that the leading edge comes later in time, one should reduce the size of the BX delay. To shift the pulse to an earlier time, one should increase the size of the BX delay. See Figure 3.29.

Shifting the phase of the clock allows for fine-tuning within the time window, but it will likely require some re-tuning of the TDM delay before the readout is correct. In Figure3.31, the diagonal dead band corresponds to the region in which the clock and the data (in this case, a trigger signal) are arriving at the DTMROC with exactly the same phase. Moving across that band effectively changes the clock tick in which the data are latched. On one side of the band, the data arrive just before the leading edge of the clock, and are latched almost immediately - on the other side, the data arrive just after the leading edge of the clock, and take almost 25ns to be latched.

There is (at least) one other thing to note in Figure 3.31. The vertical band, at constant BX, corresponds to the region in which the data from the DTMROCs are not interpreted correctly by the TRT-TTC. This vertical band provides the spread in the signal arrival time within a front-end board. More precisely, it says that the spread in the timing distributions for chips from the same front-end board will have a minimum width of roughly half the width of this diagonal band. In the worst case, this means a spread

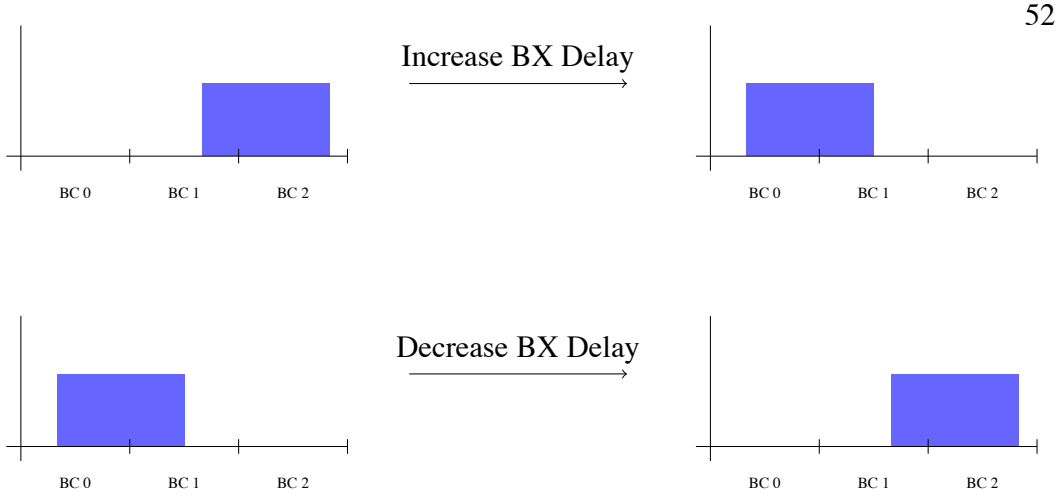


Figure 3.29: The effects of changing the clock (BX) delay on the time position of digitized pulse. Increasing (decreasing) the value of the delay makes the pulse appear to move left (right). In fact, it is the x-axis (time) which moves with respect to the pulse.

of almost 6ns, which means that the available window of 75ns is almost completely consumed by all of the constraints on the timing parameters.

3.16 Setting Fine Delays in Practice

Given all the constraints above, this section covers how the various fine delay values are actually chosen in practice. The software used to choose these values is located in the `DAQ/Integration` directory of the TRT DAQ CVS tree. The program `xtrt.exe` is used to perform delay scans; it is described in [20]. The program `fine` is used to choose delay values from those scans.

The process of choosing delays typically starts with a desired set of BX values, motivated by physics. These values were originally chosen based on cable length measurements, a process which was quite successful at aligning the detector reasonably well in time to within $\pm$5ns. As data has become available, these original values have been refined by offline analysis, with a timing goal of $\pm$1ns in the barrel and $\pm$2-3ns in the endcaps.

In any event, once the BX values have been chosen, the next step is to find corresponding DX and strobe values which guarantee stable readback. Finally, GOL delay values are chosen to ensure stable readout along the data path to the RODs. The problem of selecting TTC fine delays and GOL delays typically factorizes, so that one can simply pick the former and then the latter, without any need for iteration. There is a special case where this does not work, described in section 3.16.3. The steps for choosing delays are described in more detail here.

3.16.1 TTC **Fine Delays**

Given a BX value, the choices of DX and strobe to maximize stability are typically straightforward. These parameters are chosen to put the operating delay point as far as possible from the dead regions in delay space, measuring with a flat metric in sc bx-dx space. However, for a small number of boards, there may be no sufficiently stable region for some choices of BX because the chosen BX value is always too close to the vertical band. This can happen because of the large number of chips on some boards creating a wide vertical band, or it may happen that due to a hardware problem, not all strobe choices are available. In these cases, a new BX value is chosen to improve stability, where the new value is taken to be within $\pm$2ns of the original value. See Fig. 3.30.

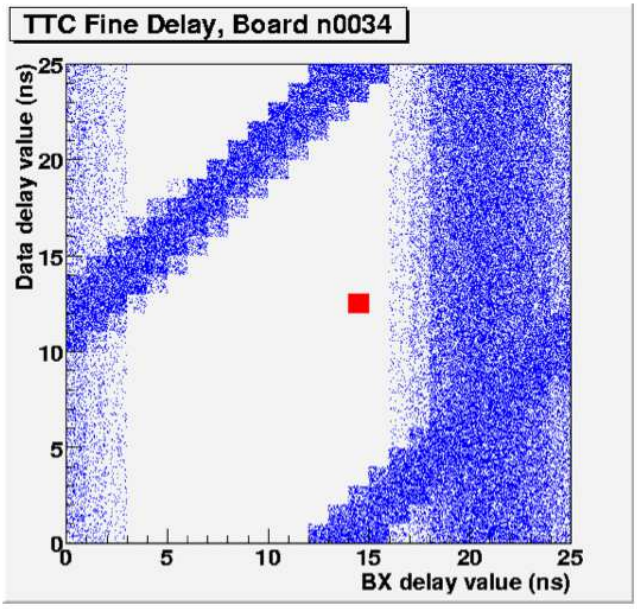


Figure 3.30: A TTC delay scan, with the delay point chosen by `fine` in red. For this value of BX, this was the best delay point possible, but it is still close to the vertical band. For greater stability, it should be shifted to the left.

In Figure 3.31 four ttc fine delay scans for a barrel front-end board, one for each ttc strobe (or edge) is shown. The diagonal band is caused by clock/data phase mismatches in the dtmroc, while the vertical band is due to poor interpretation of the dtmroc register readback at the trt-ttc. The vertical band can be shifted by adjusting the ttc strobe, while the position of the diagonal band is determined by cable and via lengths between the trt-ttc and the front-end board, and should remain fixed over all scans of that board.

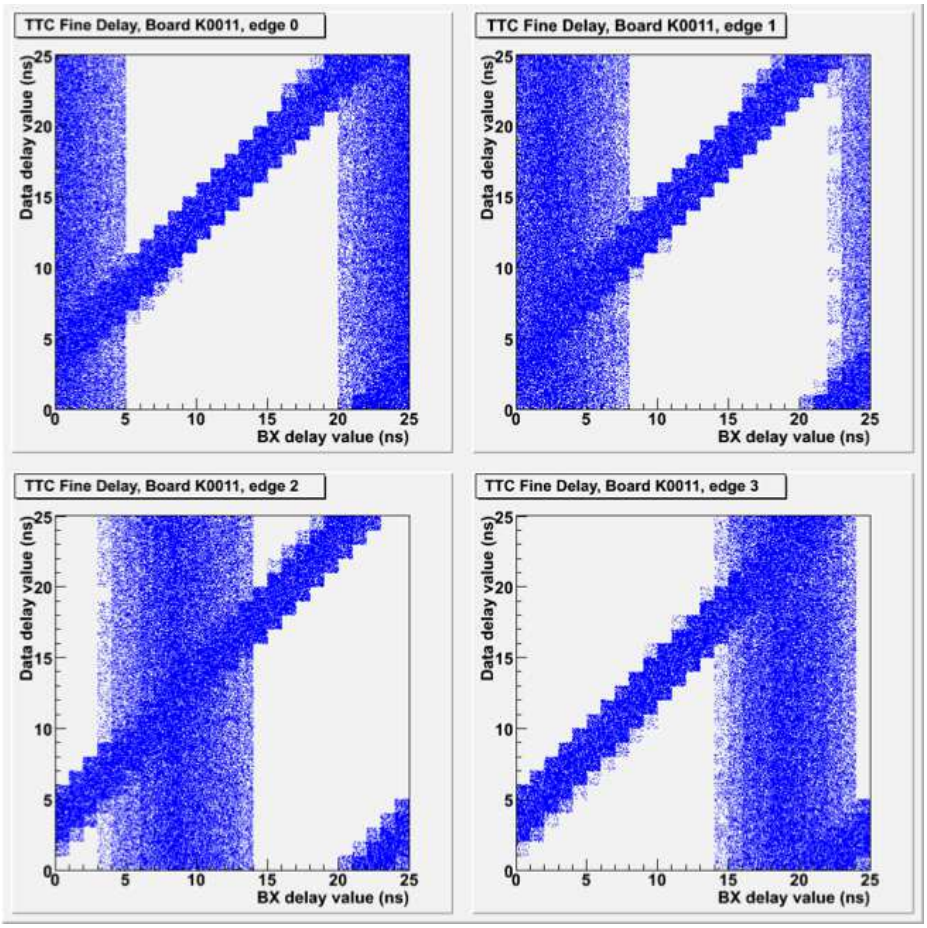


Figure 3.31: *TTC fine delay scans*

3.16.2 ROD **Fine Delays**

Given the choice of BX, DX and strobe, GOL delay values are chosen to obtain stable readout of the data from the DTMROC's. These GOL delay values only directly depend on the BX delays, but if DX and strobe are not set correctly, communication with the front end will fail and there will be no way of choosing correct GOL delays. The granularity for setting GOL delays differs from BX, however, because DTMROCs from multiple front end boards feed into each GOL chip.

So to choose GOL delays, the scans of BX versus GOL delay for each board reporting to a given GOL are effectively overlaid, taking into account the BX offsets between the boards, to create one plot of BX versus GOL delay for the entire GOL. Then a GOL delay can be chosen which maximizes stability of the readout from all the DTMROCs. However, this becomes difficult in the endcap, as described in the next section.

3.16.3 Overconstrained Timing

One example of overconstrained timing was found in section 3.16.1, where, for some values of BX, a stable set of DX and strobe cannot be found, and so physics timing must give way to readback stability. A more complicated scenario arises in the endcap for some boards. As described in the previous section, to choose GOL delays, one must take into account the BX delays of multiple boards. However, there is no way to adjust the relative timing of signals at the input of the GOL chip, only at the output. It occasionally happens that the dead regions in the front end board-level GOL delay scans all fail in different places. Then when these scans are overlaid to for the complete scan for a given GOL, the total dead region is so large that no stable operating point can be found.

In these cases, it becomes necessary to go back and adjust the choice of BX for each board until a stable GOL delay value can be found. So this requires an iterative process, and it also prohibits one from choosing the optimal delays based on physics alone. Thankfully, this doesn't occur very often (on the order of 1 or 2 GOLs for the endcaps), but it is, unfortunately, an inherent limitation in our system. This is expected to be less of an issue for collisions timing than for cosmics timing.

3.16.4 Timing from Cable Lengths

One of the primary reasons that timing delays need to be set differently throughout the TRT is signal propagation time. So by measuring the lengths of the cables, along which these signals propagate, one can get a good estimate for what delays should be set. This study was undertaken for the TRT, and the results were used to determine the BX and TDM delay values for all boards before any particle data was available. This method turned out to be quite successful, and most boards were aligned within $\pm$5ns of each other in time. For detailed information on this study, see [21].

3.17 Detector Configuration Database

The Object Kernel Source (OKS) library was developed by PNPI (Saint Petersburg) to create a configuration database for setup and diagnostics systems. It was later modified to suit the configuration needs of the ATLAS TDAQ project. Detector in test stand is using OKS for configuration and DAQ settings.

It was designed using an object-oriented model, with class definitions and objects stored in XML files. The "objects" can software applications, pieces of custom or commercial hardware, environment variables, or any number of other containers for information needed at run time. The subsections below describe some of the most common classes and objects.

3.17.1 Structure: Partitions and Segments

An OKS database is designed as a relational database, usually using a tree-like structure with a *Partition* object at its base, *Segment* objects which define the main branches, and then other objects which belong to either the segments or to the partition directly. In many ways, a partition is only a special kind of segment, one which provides a handle with which to access subset of objects defined in the XML files that is self-contained and sufficient for configuring the DAQ.

Each segment has an associated *RunControlApplication*, a process that responds to state transition commands from the segment's parent, and transmits those commands to the segment's children. The *Partition* object, being special, is associated with the *RootController*, which is a special kind of *RunControlApplication*.

A segment (or partition) can have as children:

- Other segments
- Applications
- Sets of hardware/software resources
- Anything else which inherits from:
    - Segment
    - Resource
    - Application

3.17.2 Applications

The *Application* class is the parent class for a large subset of OKS objects. It defines some basic attributes and relationships that many processes need to initialize and run properly. Most processes inherit from the application class, and are defined as, for instance, *RunControlApplication*s, *InfrastructureApplication*s, *ReadoutApplication*s, etc. The basic attributes include the name and location of the binary which is run, the computer on which the application should be started, and what should be done in the event that the application dies or throws an error.

An application is typically launched by another application - either by a *RunControlApplication* or by the *RootController* directly. Process manager (PMG) servers on each machine facilitate the launching and monitoring of each application in a partition.

3.17.3 Computers

Each computer in the system is defined by a *Computer* object. Many of the computers are handled centrally by the TDAQ group, so some of those objects are defined in a file provided centrally. Most of the TRT-specific objects are defined in an XML file in *DAQ/RCD/databases/hw/computers.data.xml*.

Most applications will have a *RunsOn* relationship defined, which points to a computer. If the *RunsOn* field is not defined, then the application will be launched on the machine which runs the partition.

3.17.4 Software Repositories

The configuration database is also used to provide a pointer to compiled libraries and binaries that the DAQ needs to run. It does this via objects called *SoftwareRepositories*. They are simply pointers to areas where executables, libraries and plugins are stored in the filesystem. There is one repository for the TRT, and some others for the global TDAQ system. (Every other subsystem will also have its own repository.) Higher level objects, such as parititions and segments, will often include these software repository objects so the applications the segment is responsible for launching can be found.

3.17.5 Resources

The TDAQ configuration and control structure uses the *Resource* base class to define objects which can be enabled or disabled without removing them from the configuration entirely. A *Resource* is typically associated with a piece of hardware, such as a ROD, TTC, or busy module input.

On top of the *Resource* base class, another type of object was defined which allows *Resource* objects to be bundled together and respond to the states of other objects in the same bundle. These objects are called *ResourceSet*s, and take three forms:

- *ResourceSet*: Allows many *Resources* to be grouped together, so that by disabling the set, all *Resource*s in the set are also disabled.
- *ResourceSetOR*: If any *Resource* contained in this object is disabled, then the *ResourceSetOR* object becomes disabled (thereby disabling all objects it contains as well.)
- *ResourceSetAND*: If all *Resources* contained in this object are disabled, then the *ResourceSetAND* will become disabled.

These three classes, and especially the last two, make it possible to define complex interdependencies between different pieces of hardware in the system. For example, in

the TRT, we have TTC objects (which are *Resources*) which can be disabled in the main IGUI panel. If one of them is disabled, we want both to be disabled. Thus, the following object is defined in the database:

```
<obj class="ResourceSetOR" id="ttc350108_ResourceSet">
  <rel name="Contains" num="2">
   "TRTTTC06Module" "ttc350108"
   "ResourceSetAND" "ttc350108_RODs"
  </rel>
</obj>


<obj class="ResourceSetAND" id="ttc350108_RODs">
  <rel name="Contains" num="2">
  "ResourceSetOR" "rod311800_ResourceSet"
  "ResourceSetOR" "rod311900_ResourceSet"
  </rel>
</obj>
```

This is, of course, only for one TTC - a similar object is defined for every TTC in the system.

A TTC is also connected to several ROD objects. If we have a situation where all RODs connected to a TTC are disabled, then we want the corresponding TTC to be disabled as well.

```
<obj class="ResourceSetAND" id="ttc350108_RODs">
  <rel name="Contains" num="2">
  "ResourceSetOR" "rod311800_ResourceSet"
  "ResourceSetOR" "rod311900_ResourceSet"
  </rel>
</obj>

<obj class="ResourceSetOR" id="ttc350108_ResourceSet">
  <rel name="Contains" num="2">
   "ResourceSetOR"  "ttc350108_TTC_and_ChannelDelay"
   "ResourceSetAND" "ttc350108_RODs"
  </rel>
</obj>
```

We now have a situation where if a TTC is disabled, then everything that depends on that TTC is also disabled. Likewise, if everything that depends on a TTC is disabled, then the TTC itself, now not needed, is also disabled.

Similar *ResourceSet* structures are used for other purposes:

- Mapping of ROD outputs to ROS inputs
- Mapping of TTC busy signal outputs to RODBusyModule inputs
- Linking Event Builder inputs to SBCs and/or ROS PCs

3.17.6 Variables

The *Variable* object in OKS serves as the equivalent of an environment variable in a *nix shell. Applications can link to these *Variable* objects in their *ProcessEnvironment* relationship. The name and value of the *Variable* are then passed in as part of the environment for the process as it is launched by a process manager.

In the detector, *Variables* are used to define the type of database being used to configure the front end (OKS or CORAL), whether the RODs readout via VME or not, and for many other purposes. The value of a variable is typically found by a *getenv()* call in a plugin or application which has that variable in its process environment.

3.17.7 Memory Allocation

For some parts of the DAQ, specifically those parts which are involved with processing data buffers from readout electronics, the amount and type of memory available in which to store those data buffers is defined in the OKS database. These *MemoryPool* objects are then linked to applications, which keeps a runaway process from hogging all of the available memory on a computer.

```
<obj class="MemoryPool" id="TRTBarrelA_RCDMemoryPool">
 <attr name="Type" type="enum">"CMEM_BPA"</attr>
 <attr name="NumberOfPages" type="u32">500</attr>
 <attr name="PageSize" type="u32">4096</attr>
</obj>
```

The *Type* of memory can be either *Malloc* or *CMEM_BPA*. *Malloc* uses the *C* standard *malloc()* call to allocate memory. It's not always clear, though, that this is what actually happens - there's some evidence that certain applications ignore the *Malloc* tag and default to *CMEM_BPA* always.

CMEM_BPA memory is reserved space in the contiguous memory in the *bigphysarea* defined in the kernel boot parameters. The contiguous memory is handled by the *cmem_rcc* driver, a part of the TDAQ release. The ATLAS VME driver, called *vme_rcc*, relies heavily on a large block of contiguous memory to manage the VME address mapping. The *ReadoutApplications* also use the contiguous memory, though it's not clear that it's really necessary.

3.18 Partitions

The default partition for the detector is the *TRTSR1* partition, which includes all the pieces needed to run the detector in standalone, data-driven event building mode. At the partition level, there are two segments included: one for the detector, including everything that is needed to run the detector in either standalone or combined mode;

and one for the standalone event builder, which is only ever used in standalone mode. Besides the top level segments, the partition also defines some other useful things:

- The default computer on which applications will be launched (if another computer is not specified)
- The default location of log files
- Various tags and run type definitions which can be chosen in the IGUI panels
- The source (i.e. an IS object) which provides event counter information for the IGUI
- The pieces of the partition which are disabled

The last item above, the pieces of the partition which are disabled, is particularly important. The *Partition* object, and this relationship in particular, is the only part of the OKS database which can be directly changed by the IGUI (as opposed to being changed by the *oks_data_editor* or another dedicated tool). The "Segments and Resources" panel, described elsewhere, is the customary way to enable or disable various parts of the system.

Depending on how the partition was generated, it's possible that the database will become inconsistent by referencing objects in the disabled relationship that no longer exist in the partition tree. It's important to remember, whenever using *gendb* or other related tools, to clean out the disabled relationship of the *Partition* object before trying to run.

3.19 Segments

The different segments of the TRT *Partition* are described in this section. Segments exist at various levels - there are "top" level segments, and segments which are children of those top level segments. In principle, there's no limit to the number of generations of segments inside of a partition, but in the case of the TRT there are only three levels of segments - one at the top level, one just below that, and a third that is specific to applications related to DAQ-DCS communications (DDC).

3.19.1 TRT Segment

The main top-level segment is the TRT Segment. This is the segment included by both standalone partitions and by the ATLAS combined partition. It includes between four and six segments of its own - one segment for each TTC partition, one segment for monitoring applications, and the last segment for DAQ scanning software.

The TRT segment also includes links to other TRT-specific applications. All of the infrastructure applications, IS servers, and an application which records run-time data into a database are launched by the TRT segment.

Connected with the TRT segment is a *RunControlApplication* which manages the launching and monitoring of the child segments and applications descended from the segment. This *RunControlApplication* also pulls in many of the environment variables needed by the different TRT applications.

3.19.2 Event Builder Segment

The only other segment included in the TRT partition directly, besides the TRT Segment, is the Event Builder Segment. The event builder is separated from the TRT segment to allow the TRT segment's structure to be used without modification in both standalone and combined running.

The event builder segment is composed of a segment (and its associated controller) and a readout application, which takes as inputs the event fragments from either the VME crates or the ROS PCs. The resources associated with the inputs to the event builder readout application are tied, via resource sets, to the corresponding objects in the TRT segment.

3.20 Readout Applications

Readoutapplications are used for both data readout and module configuration. The readout part is only really used when the RODs are read out over VME, or when we're running in standalone mode and reading out the ROSs. Most of the time, the *ReadoutApplication* is only responsible for facilitating the state transitions of the resources it controls. The term *ReadoutApplication* actually refers to the binary executable which is called and which does the work - the OKS object itself is usually called an *RCD* or *ROS* object.

Each ROD crate has an *RCD* object defined which controls all of the TTCs and RODs installed in that crate.

```
<obj class="RCD" id="TRTBarrelA_B-01">
 <attr name="ActionTimeout" type="s32">120</attr>
 <attr name="ShortTimeout" type="s32">5</attr>
 <attr name="ProbeInterval" type="s32">5</attr>
 <attr name="FullStatisticsInterval" type="s32">63</attr>
 <attr name="IfError" type="enum">"Error"</attr>
 <attr name="Parameters" type="string">""</attr>
 <attr name="RestartParameters" type="string">""</attr>
 <attr name="IfDies" type="enum">"Error"</attr>
 <attr name="IfFailed" type="enum">"Error"</attr>
```

```
 <attr name="StartAt" type="enum">"Boot"</attr>
 <attr name="StopAt" type="enum">"Shutdown"</attr>
 <attr name="InitTimeout" type="u32">60</attr>
 <attr name="InputDevice" type="string">""</attr>
 <attr name="Logging" type="bool">1</attr>
 <rel name="Trigger" num="1">
  "DataDrivenTriggerIn" "TRTRODDataDrivenTriggerIn_B-01"
 </rel>

 <rel name="Output">"" ""</rel>
 <rel name="Contains" num="4">
  "TRTTTC06Module" "ttc350108"
  "TRTROD05Module" "rod311800"
  "TRTROD05Module" "rod311900"
  "TRTCrateMonitor" "CrateMonitor_BarrelA_B-01"
 </rel>

 <rel name="MemoryConfiguration">"MemoryPool" "TRTRCDMemoryPool_B-01"</rel>
 <rel name="Detector">"Detector" "TRT-BC"</rel>
 <rel name="RunsOn">"Computer" "vmetrt15.cern.ch"</rel>
 <rel name="InitializationDependsFrom" num="0"></rel>
 <rel name="ShutdownDependsFrom" num="0"></rel>
 <rel name="Program">"Binary" "ReadoutApplication"</rel>
 <rel name="ExplicitTag">"" ""</rel>
 <rel name="Uses" num="1">
  "SW_Repository" "trt_sw_repository"
 </rel>
 <rel name="ProcessEnvironment" num="1">
   "VariableSet" "TRT_ROD_CRATE_VARIABLES"
 </rel>
 <rel name="Configuration">"ReadoutConfiguration" "RCD_Config"</rel>
</obj>
```

There are several important things to note in this configuration:

- The *Contains* relationship is where all of the TTCs, RODs, and ChannelDelays end up.
- The *ProcessEnvironment* relationship has pointers to all of the variables needed by the process and the modules it contains to go through all of the state transitions.
- The *RunsOn* relationship points to the computer on which the process should be run - in the case of an Event Builder, for instance, this can be almost any PC, but for *RCD* objects which control VME electronics, it's clearly important to get that correct.
- The *Detector* relationship is a pointer to an object which defines the TTC partition: in the case of the TRT, it can be
  - TRT Barrel A

    * TRT Barrel C
    * TRT Endcap A
    * TRT Endcap C

  The *Detector* object is then used to determine things like ROB IDs in the ROS or in an event builder.

- The *Configuration* object is something which holds parameters for *ReadoutApplications* which are presumably common to many different instances. It is rarely changed for TRT runs.

Each TTC partition also has a similar RCD object. Show below is only the *Contain* relationship - everything else is almost identical:

```
<rel name="Contains" num="3">
 "LTPModule" "TRTBarrelA_LTP"
 "RODBusyModule" "TRTBarrelA_RODBusy"
 "TTCviModule" "TRTBarrelA_TTCvi"
</rel>
```

The *ROS* objects are almost identical. In their case, they contain *RobinReadoutModules* instead of TTCs and RODs. The configuration of the ROSs is typically provided for us by the TDAQ group. There are typically two files used for ROS configurations, with only one of them being used at any given time: *ROS-TRT-robin-datadriven-reb.data.xml* and *ROS-TRT-robin-dc.data.xml*. The first is used for standalone running, while the second is used in "Data Collection" mode, i.e. when we're in a combined run.

3.21 TRT Specific Objects

To make the DAQ aware of TRT hardware, a set of TRT specific objects has been defined in the OKS framework. The class definitions are held in a schema file, and the objects themselves are defined within the normal configuration tree (as parts of segments, applications, etc.).

There's an important caveat to the OKS information about the modules described below. In the case when the CORAL database is used, many of the parameters contained in the OKS objects are ignored. In that case, the only thing used from the OKS objects is the *DetID* (or, in the case of the DTMROCs, *ChipID*) attribute, which is used to retrieve the configuration parameters from CORAL.

3.21.1 Schema

The schema file, located in *DAQ/RCD/databases/schema/TRTConfig.schema.xml* defines the following object classes:

- TRTTTC06Module
- TRTTTC06Group
- TRTROD05Module
- TRTDTMROCModule
- TRTDTMROCParameters
- TRTChannelDelayModule

The classes are described in more detail below. The class definitions set the objects attributes and relationships, as well as the base classes from which they should inherit.

3.21.2 RODs

The ROD object class definition looks like this:

```
<class name="TRTROD05Module" description="Parameters for TRT ROD05 module">
 <superclass name="ResourceSetAND"/>
 <superclass name="SequentialInputModule"/>
 <attribute name="DetID" description=" detector ID" type="u32" format="hex"
 init-value="0x3100" is-not-null="yes"/>
 <attribute name="VMEslot" description="VME slot no of ROD" type="u32"
 format="hex" init-value="0x210000" is-not-null="yes"/>
 <attribute name="RODhalf" description="Half of the ROD being used"
 type="enum" range="bottom,top" init-value="bottom" is-not-null="yes" />
 <attribute name="EdgeSelect0" description="Edge select 0" type="u32"
 format="hex" init-value="0" />
 <attribute name="EdgeSelect1" description="Edge select 1" type="u32"
 format="hex" init-value="0" />
 <attribute name="EdgeSelect2" description="Edge select 2" type="u32"
 format="hex" init-value="0" />
 <attribute name="EdgeSelect3" description="Edge select 3" type="u32"
 format="hex" init-value="0" />
 <attribute name="GolDelay0" description="GOL delay 0" type="u8" init-value="0" />
 <attribute name="GolDelay1" description="GOL delay 1" type="u8" init-value="0" />
 <attribute name="GolDelay2" description="GOL delay 2" type="u8" init-value="0" />
 <attribute name="GolDelay3" description="GOL delay 3" type="u8" init-value="0" />
 <attribute name="RODstatus" description="Status of this ROD - enabled or disabled"
 type="bool" init-value="1" is-not-null="yes"/>
 <attribute name="Compression" description="compression (0 - uncomp, 1 - zero-suppr,
  2+ - full comp)" type="u32" init-value="0" is-not-null="yes"/>
 <attribute name="Comment" description="Comments on this module" type="string"/>
 <relationship name="dtmroc_params" description="DTMROC parameters"
 class-type="TRTDTMROCModule" low-cc="zero" high-cc="many" is-composite="no"
 is-exclusive="no" is-dependent="no"/>
 <relationship name="ttc_params" description="TTC modules"
  class-type="TRTTTC06Module" low-cc="one" high-cc="one" is-composite="no"
  is-exclusive="no" is-dependent="no"/>
</class>
```

It inherits from two classes: *ResourceSetAND* and *SequentialInputModule*. The latter gives the ROD the functionality it needs to collect data during the *Running* state of the DAQ.

Each object has all of the configuration parameters needed to determine which ROD is being referenced, to set GOL delays and edges, and to enable the compression (if necessary).

3.21.3 TTCs

The TTCs are set up in such a way that their configuration is broken into two pieces: the main TRTTTC06Module, and a special TRTTTC06Group class for containing the configuration data for 10 TTC lines.

TRTTTC06Module

```
<class name="TRTTTC06Module" description="Parameters for TRT TTC06 module">
 <superclass name="ResourceSetAND"/>
 <superclass name="ReadoutModule"/>
 <attribute name="DetID"         description="ATLAS-wide detector ID number"
  type="u32"    format="hex"   init-value="0x3000" is-not-null="yes"/>
 <attribute name="VMEslot"       description="VME slot number for this TTC"
 type="u32"    format="hex"   init-value="0x2300000" is-not-null="yes"/>
 <attribute name="Delay"         description="Time delay"
  type="u8"     init-value="0" is-not-null="yes"/>
 <attribute name="TTCrxDelay" description="Time delay" type="float"
 init-value="0" is-not-null="yes"/>
 <attribute name="ArShaping"     description="Ar Shaping Bit"

 type="bool"   init-value="false" is-not-null="yes"/>
 <attribute name="SetSendID"     description="Send ID toggle"
 type="bool"   init-value="false" is-not-null="yes"/>
 <attribute name="ClockSource"   description="Clock source: ext,int,ttcrx"
 type="string" range="ext,int,ttcrx" init-value="ext" is-not-null="yes"/>
 <attribute name="Comment" description="Comments on this module" type="string"/>
 <relationship name="groups"     description="Collection of 10 lines"
 class-type="TRTTTC06Group" low-cc="zero" high-cc="many"/>
 <relationship name="rod_params" description="ROD parameters"
 class-type="TRTROD05Module"  low-cc="zero" high-cc="many" is-composite="no"
 is-exclusive="no" is-dependent="no"/>
 <relationship name="dtmroc_params" description="DTMROC mapping"
 class-type="TRTDTMROCModule" low-cc="zero" high-cc="many" is-composite="no"
  is-exclusive="no" is-dependent="no"/>
</class>
```

That part of OKS is used to configure TTC module of the DAQ system.

TRTTTC06Group

```
<class name="TRTTTC06Group" description="Collections of 10 TTC lines">
 <attribute name="Group" description="Group on TTC (0-3)" type="u8"
  init-value="0" />
 <attribute name="DutyCycle" description="Duty Cycle Array" type="float"
```

```
  init-value="0" />
  <attribute name="EdgeSelect" description="Edge select" type="u8"
  is-multi-value="yes"/>
  <attribute name="finebx" description="BX delays Array" type="u8"
   is-multi-value="yes"/>
  <attribute name="finedx" description="Dout delays Array" type="u8"
  is-multi-value="yes"/>
  <attribute name="TDMdelay" description="TDM Phase Delay Array" type="u8"
  is-multi-value="yes"/>
 </class>
```

Here, the *Group* corresponds to the connector on the front-panel of the TTC, or to the *(int)(ttc line)/10*. Each of the following attributes has 10 values, one for each line. For example:

```
<obj class="TRTTTC06Group" id="ttc350108-0">
 <attr name="Group" type="u8"> 0 </attr>
 <attr name="DutyCycle" type="float"> 50 </attr>
 <attr name="EdgeSelect" type="u8" num="10"> 0 0 0 0 0 0 0 0 0 0 </attr>
 <attr name="finebx" type="u8" num="10"> 49 49 49 49 49 49 49 49 49 0 </attr>
 <attr name="finedx" type="u8" num="10"> 33 34 23 11 17 11 12 20 24 0 </attr>
 <attr name="TDMdelay" type="u8" num="10"> 14 14 15 15 15 15 15 15 15 0 </attr>
</obj>
```

In this case, the values of the delays and the edges are invalid (the largest fine delay is 49, the largest TDM delay is 255, and the largest edge is 3) - this means that the object has not yet been initialized. When using the CORAL database for configuration, these values are ignored completely.

3.21.4 DTMROCs

The DTMROC objects are (at the time of this writing) almost always configured using the CORAL database. Thus, there was a decision to split most of the information in the DTMROC modules into two pieces - one which contains only what's needed to retrieve the CORAL information, and one which has everything the front-end chips will need in case the CORAL database is not used.

TRTDTMROCModule

```
<class name="TRTDTMROCModule" description="TRT DTMROC module ID
and link to parameters">
 <superclass name="ReadoutModule"/>
 <attribute name="ChipID" description="Chip ID number" type="u32"
 init-value="1" is-not-null="yes"/>
 <relationship name="dtm_params" description="default DTMROC parameters"
  class-type="TRTDTMROCParameters" low-cc="zero" high-cc="one" is-composite="no"
  is-exclusive="no" is-dependent="no"/>
</class>
```

Since the *ChipID* is the only thing needed to retrieve the configuration from CORAL, this is the only attribute left behind.

TRTDTMROCParameters

```
<class name="TRTDTMROCParameters" description="Default DTMROC
parameters, for use with CORAL DB.">
 <attribute name="Chip_Valid" description="Is this chip is valid"
 type="bool" init-value="1" is-not-null="yes"/>
 <attribute name="RODgroup" description="ROD group from 1 to 4(5)"
  type="u8" init-value="1" is-not-null="yes"/>
 <attribute name="RODinput" format="hex" description="Which line of
 the RODgroup is taken by this DTMROC (1..26(20))" type="u32"
 init-value="0x00000001" is-not-null="yes"/>
 <attribute name="HW_Addr_FE" description="Hardware address of chip"
 type="u8" init-value="1" is-not-null="yes"/>
 <attribute name="TTC_Line_FE" description="TTC line this FE is on (1..18)"
 type="u8" init-value="1" is-not-null="yes"/>
 <attribute name="Thresh0_Low_FE" description="Low threshold of 0"
 type="u8" init-value="124" is-not-null="yes"/>
 <attribute name="Thresh0_High_FE" description="High threshold 1"
 type="u8" init-value="109" is-not-null="yes"/>
 <attribute name="Thresh1_Low_FE" description="Low threshold 2"
 type="u8" init-value="124" is-not-null="yes"/>
 <attribute name="Thresh1_High_FE" description="High threshold 2"
 type="u8" init-value="109" is-not-null="yes"/>
 <attribute name="VT_DAC0_FE" description="Voltage and temperature DAC0"
 type="u8" init-value="0" is-not-null="yes"/>
 <attribute name="VT_DAC1_FE" description="Voltage and temperature DAC1"
 type="u8" init-value="0" is-not-null="yes"/>
 <attribute name="Mask_FE" description="Channel mask" type="u32"
 format="hex" init-value="0xFFFF" is-not-null="yes"/>
 <attribute name="Pipe_Latency_FE" description="Pipe latency"
 type="u8" init-value="130"/>
 <attribute name="Clock_FE" description="external/internal clock"
 type="string" range="ext,int" init-value="ext"/>
 <attribute name="Comment" description="Comments on this module"
  type="string"/>
 <relationship name="rod_params" description="Read-out status"
 class-type="TRTROD05Module" low-cc="zero" high-cc="one"
 is-composite="no" is-exclusive="no" is-dependent="no"/>
 <relationship name="ttc_params" description="TTC modules"
 class-type="TRTTTC06Module" low-cc="zero" high-cc="one"
 is-composite="no" is-exclusive="no" is-dependent="no"/>
</class>
```

Everything here will be ignored if the CORAL database is used.

3.21.5 Channel Delay Module

The Channel Delay Module requires minimal configuration, and is special compared to the objects above - it does not have any configuration stored in CORAL, so what's in OKS is what is used. In fact, the Channel Delay Module as implemented in the plugin library represents a "hole" in our configuration scheme - it uses the ROD and TTC information stored in the OKS database directly, even if the CORAL database is used to configure those RODs and TTCs. Fortunately, little more than the VME slot and other geographical information is used by the Channel Delay Module, so it's probably

not a severe problem.

The schema for this module looks like this:

```
<class name="TRTChannelDelayModule" description="TRT Channel Delay module">
 <superclass name="ResourceSetAND"/>
 <superclass name="ReadoutModule"/>
 <attribute name="DetID"   description="ATLAS-wide detector ID number" type="u32"
 format="hex" init-value="0x3000" is-not-null="yes"/>
 <attribute name="VMEslot" description="VME slot number for this TTC"  type="u32"
 format="hex"   init-value="0x2300000" is-not-null="yes"/>
 <attribute name="TTC_UID" description="UID of the TTC belonging to the
 ChannelDelay module" type="string" is-not-null="yes"/>
</class>
```

It only needs the TTC information, and infers the RODs from the location of the TTC. Again, this module **does not** use CORAL at all.

CHAPTER 4

RESULTS

Most part of the construction of the TRT replica has been completed. Both Barrel and Endcap of the detector mechanically assembled. In the setup with my contribution:

- Single board computers, and other server systems which are used for data taking set up, configured and necessary tools are installed.
- Connections of electronic components and their tests are done.
- Detector cooling system and active gas system are constructed, their leakage tests has been done.
- Both scintillator trigger and Fast-OR trigger logic are built and tested.
- Low voltage system connections, calibration, power supply cabling are done and software which is needed to control the system is prepared.
- Logical design of the High voltage system is set as the real.
- TRT detector to prevent complications on software side and tested.
- The thresholds obviously need to be tuned on the FE boards.
- OKS xml database is ready for two stacks of the barrel.
- DAQ setup finished.
- Some software improvements and changes made.

TTC delay scan shows that FE boards works well and they succesfully comunicate with TTC. TTC delay scans and ROD scans taken for 1BS board can be seen in Figure 4.1 and Figure 4.2.

TTC/ROD timing scans, threshold scans, and test pulse scans do several things:

- Establishes a history of the setup.

– It gives us some confidence that things are stable.

– It will alert us to any developing problems or sudden changes.

– It gives us a baseline set of tools to fall back on for debugging purposes that we're sure will work.

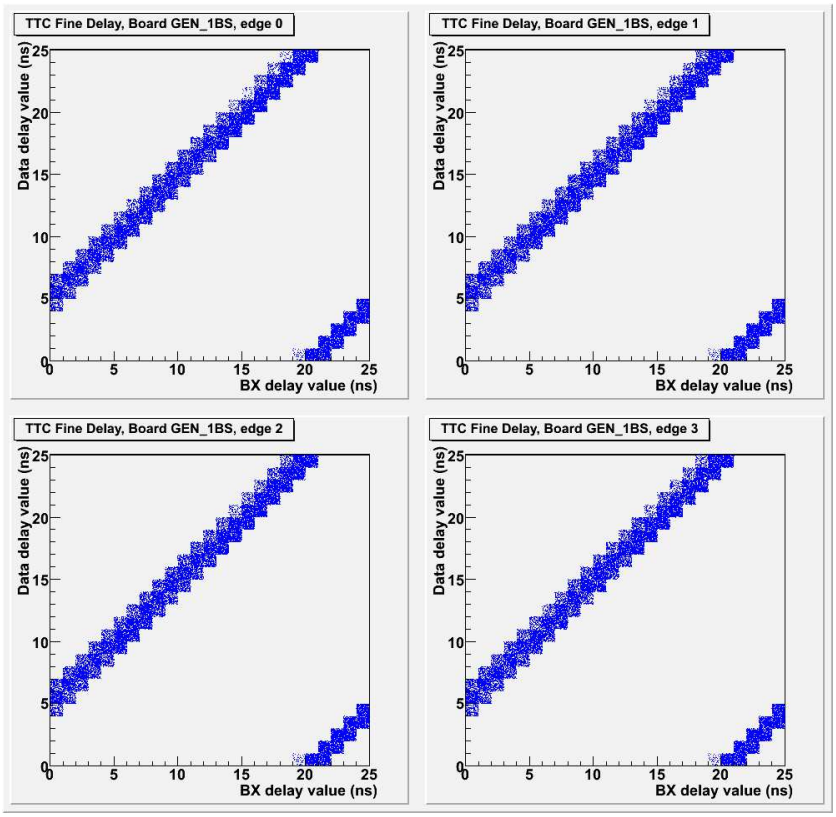


Figure 4.1: *TTC delay scan*.

The noise data which is shown in the Figure 4.3 points that the DAQ system works smoothly. Noise data is taken using random trigger. Using random trigger data taken is only used to validate that DAQ chain is working. This data is also used for equalization of threshold levels.

After having subsystems of the detector functional it is used to observe the cosmic ray. The cosmic ray candidate of Barrel A is shown in the Figure 4.4. Detector was operating at $\sim$1.5 kV voltage and using mixture of Ar, CO_2 gases. The trigger used for the data taking of candidate was scintillator trigger. The offline software has been able to provide a prompt reconstruction of these events.

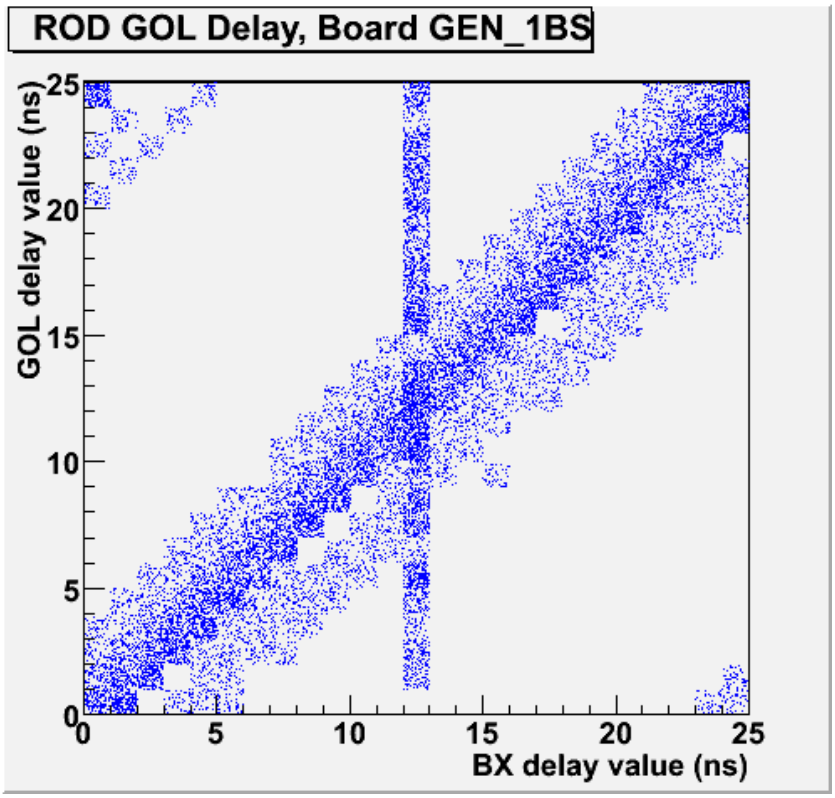


Figure 4.2: *ROD delay scan*.

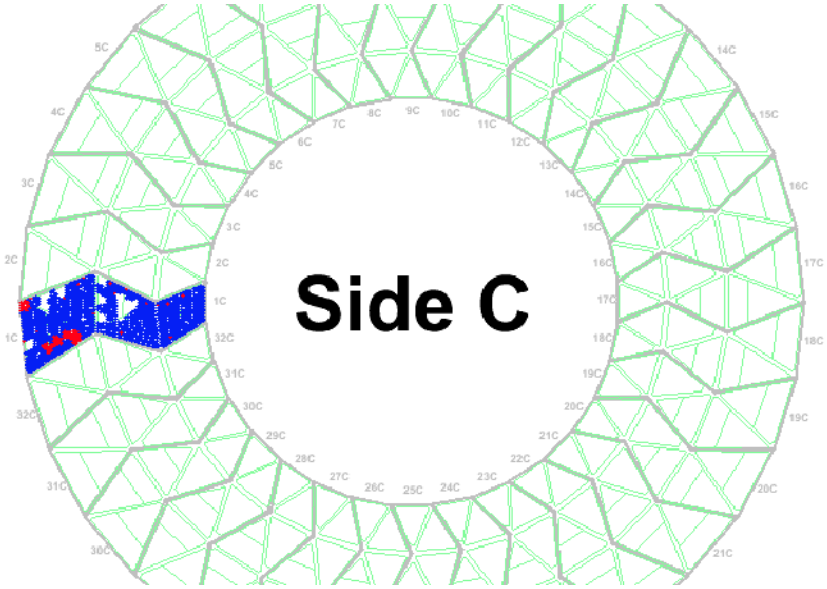


Figure 4.3: *Noise data taken*.

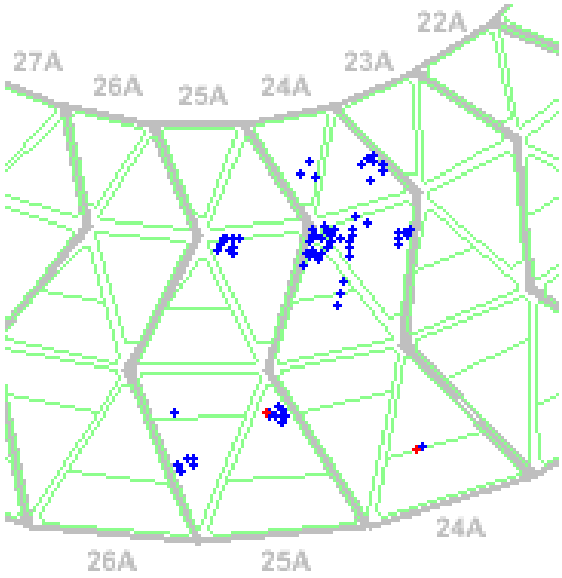


Figure 4.4: *Cosmic ray candidate*.

CHAPTER 5

SUMMARY AND CONCLUSION

In conclusion, the replica of the TRT detector is switched on and SR1 is set to work to observe the cosmic ray. The test setup has both TRT barrel and endcap sectors in SR1, and the test stand was built from spare Barrel and Endcap components of the detector in the ATLAS detector.

Major milestone for the TRT SR1 project has been accomplished as cosmic rays going through both the barrel replica detector have been successfully recorded. The candidate is the result of the effort of many people and long period of research and development, production, integration, and software preparations. Excellent collaboration among detector, DAQ, HV, HWI, DCS and software experts have made this progress possible.

The next steps for the setup should be;

- understanding time delays for the Fast-OR trigger. This is tricky to understand, before you go hunting for cosmics, you should at least know the relationship between the TDM delay in the TTC, the pipeline latency on the front-end, and the signal propagation times between the various components.
- finishing the endcap part of the detector which involves electronic testing, connections and setting for DAQ.
- after completion of endcap part, whole system must be tested together.
- completing the documentation of the system. This is very important because that will be a guide for the newcomers even for the experts to understand how the system works.

LIST OF REFERENCES

[1] Kunihiro Nagano.(2009).*ATLAS Upgrade for the Super LHC : Meeting the challenges of a ten-fold increase in collisions*, ATL-UPGRADE-PROC-2009-002.

[2] Peter Cwetanski.(2006).*Straw Performance Studies and Quality Assurance for the ATLAS Transition Radiation Tracker*, CERN-THESIS-2006-025.

[3] HVSys company. SC508 MULTI-CHANNEL HIGH VOLTAGE SYSTEM CONTROLLER USER MANUAL: HVSys company.

[4] E.Banas, P.Farthouat, Z.Hajduk, B.Kisielewski, P.Lichard, J.Olszowska, V.Ryjov, L.Cardiel Sas.(2005).*Distributed Low Voltage Power Supply System for Front End Electronics of the TRT Detector in ATLAS Experiment*.

[5] The ATLAS collaboration.(2009).*The TRT Fast-OR Trigger.*, ATL-INDET-PUB-2009-002.

[6] ATLAS Collaboration.(2008).*The ATLAS Experiment at the CERN Large Hadron Collider.*

[7] ATLAS Collaboration.(1997).*ATLAS inner detector: Technical design report. Vol. 1*.

[8] ATLAS Collaboration.(1997).*ATLAS muon spectrometer: Technical design report*.

[9] E. Abat *et al*..(2008).*The ATLAS TRT electronics*.

[10] A. dos Anjos, H.P. Beck, B. Gorini.(2009).*The raw event format in the ATLAS Trigger*, ATL-D-ES-0019.

[11] D.A. Huffman.(September 1952).*A Method for the Construction of Minimum-Redundancy Codes, Proceedings of the I.R.E.*

[12] TRT Collaboration.(2003).*TRT DTMROC Specification*.

[13] S.Baron, P.Lichard.(2005).*TRT TTC06 Specification*.

[14] M. Hance.(2006).*TRT Detector Granularity Note*.

[15] TRT Collaboration.(2003).*TRT DTMROC Specification*.

[16] H. Correia, A. Marchioro, P. Moreira and J. Schrader.(2005).*Delay25*.

[17] The ATLAS TRT Collaboration.(2008).*The ATLAS TRT Electronics*.

[18] P. Moreira, T. Toifl, A. Kluge, G. Cervelli, A. Marchioro and J. Christiansen.(October 2005).*GOL Reference Manual, Gigabit Optical Link Transmitter manual*.

[19] M. Hance.(2008).*TRT Detector Synchronization Note*.

[20] B. LeGeyt.(2006).*XTRT User's Manual*.

[21] S. Fratina, B. LeGeyt and J. Saxon.(2007).*A Method for Determining First-Pass Fine Delays for the ATLAS TRT Detector.*

PUBLICATIONS

[1] Loginov A., Diblen F., Adelman J., Banas E., Cwetanski P., Egorov K., Hajduk Z., Hance M., Kayumov F., Mouraviev S., Olivito D., Olszowska J., Penwell J., Wagner P., Zhukov K.(2010).*TRT Test Stand : description of the TRT test stand in the SR1*. ATL-COM-INDET-2010-035