Development Of A Slow Controls Alternate Data Acquisition Interface For The Solenoidal Tracker At RHIC (STAR)

BY

John Chester Meier

A THESIS

Submitted to the Faculty of the Graduate School of
Creighton University in Partial Fulfillment of
the Requirements for the Degree of Master
of Science in the Department of Physics

Omaha, 1996

Thesis Approved

by

____________________________________Major Advisor

____________________________________Dean

CU-PHY-NP 96/02
October 1996

DEPARTMENT OF PHYSICS
2500 CALIFORNIA PLAZA • OMAHA, NE 68178 • (402) 280-2835

Development Of A Slow Controls Alternate Data Acquisition Interface For The Solenoidal Tracker At RHIC (STAR)

John Chester Meier

Creighton University, Department of Physics, Omaha, Nebraska 68178, USA

A THESIS

Submitted to the Faculty of the Graduate School of Creighton University in Partial Fulfillment of the Requirements for the Degree of Master of Science in the Department of Physics, 1996

Development Of A Slow Controls Alternate Data Acquisition Interface For The Solenoidal Tracker At RHIC (STAR)

BY

John Chester Meier

Abstract

The Solenoidal Tracker At RHIC (STAR) will search for signatures of a Quark-Gluon Plasma formation. An overview of the capabilities of the STAR detector is presented. The VME-based Slow Controls system is described. The development of an alternate data acquisition path based on the High-level Data Link Control protocol and the Experimental Physics and Industrial Control System (EPICS) are discussed.

Acknowledgments

I would like to thank the people that have helped me while completing my education at Creighton University.

First I would like to thank my family. Without their love and support I would have never made it this far.

I would like to thank Dr. Michael G. Cherney for providing me with the opportunity to work on the STAR project. I am grateful for his support and encouragement; it has been a pleasure working with him. I would also like to thank Dr. Iwona Sakrejda for all her help during my stay at LBNL.

During the time I worked on the STAR project I have interacted with many individuals. It has been a privilege to work with such a diverse group of people.

Ultimately, I would like to thank my advisor Fr. Thomas S. McShane SJ.. Without him I would have never pursued an academic career in physics. He has taught me many, many things about life, people, and having fun. The difference he has made in my life is indescribable and I am honored to call him my friend.

This work was supported in part by the United States Department of Energy under contract numbers DE-FG02-91ER40652 and DE-FG03-96ER40991. I am also grateful for the assistance of the dean of the Creighton college of Arts and Sciences and the dean of the Graduate School.

There's more than one way to skin a cat.
-J.A. Meier

For Dad

Table of Contents

Chapter 2 Solenoidal Tracker At RHIC

Chapter 3 Slow Controls System

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 1. Particle Physics

1.1 Theory and History of Particle Physics

"Atom", derived from the Greek word atomos, means indivisible, and it was once thought that atoms were the indivisible constituents of matter. "Atom" is a misnomer in light of the achievements of the 20th century. When firing alpha particles at a thin piece of gold foil eleven years after the turn of this century, Ernest Rutherford found that while most of the alpha particles passed right through the foil, a few were deflected at large angles from their original trajectory. The result indicated that there was something very dense in the middle of the gold atoms in the foil. Rutherford had discovered the atomic nucleus.

With the discovery of the neutron by Chadwick in 1932 it was thought that the structure of the nucleus was complete, and that the proton, neutron and the electron were the fundamental constituents of matter. From the time of Chadwick's discovery through the 1960's many new particles were discovered. During this time many attempts were made to come up with new theories that would illustrate the similarities and differences of the escalating numbers of new "fundamental" particles.

Today two classes of particles are studied: leptons and hadrons. Particles classified as leptons are not affected by the strong force and are considered fundamental or elementary particles, that is, they are not observed to have internal structure. Six different types of leptons have been identified (see table 1.1).

FLAVOR (Symbol)	Electron (e)	Electron Neutrino (ν_e)	Muon (μ)	Muon Neutrino (ν_μ)	Tau (τ)	Tau Neutrino (ν_τ)
MASS (MeV/c^2)	0.511	About 0	106.6	About 0	1,784	Less Than 164
CHARGE	-1	0	-1	0	-1	0

Table 1.1
Lepton flavor, mass, and charge.

For each lepton there exists a corresponding anti-lepton. The anti-leptons have the same mass as their respective particles but carry the opposite charge.

The second class of particles that are studied are the hadrons. This class of particles includes the proton and neutron. Less familiar hadrons exist, but only as the product of high-energy collisions from which massive and extremely unstable particles materialize (lifetimes of about 10^{-23} seconds). Hadrons are not fundamental particles since they have been shown to have internal structure. In 1964 Murray Gell-Mann of the California Institute of Technology and George Zweig at the European Center for Particle Physics (CERN) independently proposed that hadrons were not fundamental, but are, in fact, made up of smaller particles. This proposal gave birth to the quark model. In this model all hadrons are made up of point-like spin-1/2 particles called quarks. Quarks can combine in doublets producing mesons (π's, K's, etc.) or in triplets forming baryons (proton, neutron, etc.). Quarks come in six flavors (see table 1.2), each of which is complimented by an anti-quark of opposite charge and flavor.

FLAVOR	Up (u)	Down (d)	Strange (s)	Charm (c)	Bottom (b)	Top (t)
MASS (MeV/c^2)	7	15	200	1.3 GeV/c^2	4.8 GeV/c^2	180 GeV/c^2
CHARGE	+ 2/3	- 1/3	- 1/3	+ 2/3	- 1/3	+ 2/3

Table 1.2
Quark mass, charge and flavor.

Even though the quarks have a fraction of the charge of the electron, they combine in such a way that the sum of their charge is integral and have never been observed to combine in such a way to produce a particle with fractional charge.

Experiments during the 1960's and 1970's in which high energy electrons were fired at protons and neutrons supported the quark theory. These experiments found the distribution in energy and angle of scattered electrons suggestive of electrons colliding with point-like, electrically charged objects within the protons and neutrons [CQ 85].

The quark model was not flawless from the start. Nature has shown that no two particles in the same quantum state can have the same quantum numbers (Fermi-Dirac principle), but at the same time nature produces baryons that are the bound state of three of the same quarks (i.e. Ω^- (sss) and Δ^- (ddd)). According to the Fermi-Dirac principle, three quarks, all of which are spin-1/2 particles, should not be allowed to combine in the same quantum state to form a baryon. The Fermi-Dirac principle has had success in other scientific areas (i.e. determining an atoms placement in the periodic table), and one would expect this principle to account for the configuration of quarks within a hadron as well.

To reconcile the discontinuity, the quark model was modified by the inclusion of a new quantum number arbitrarily termed color. Each quark can have one of three kinds of color charge: red, green, or blue. An anti-quark could have one of three kinds of color charge: anti-red, anti-green, or anti-blue (color as used here serves as a convenient label and is not connected with ordinary color). With the inclusion of color charge as a quantum number, we can now specify the rules under which quarks combine to form hadrons. In nature, the observed hadrons do not demonstrate color charge which means that the sum of the component quarks in a hadron must be color neutral or white. Therefore the allowable combinations of quarks are: a quark and its anti-quark, yielding the mesons, and a red, a green and a blue quark producing the baryons.

Experiments in the late sixties measuring the momentum of the quarks within a hadron, found another discrepancy between nature and the quark model. It was thought that if a proton moves with a momentum p, by conservation of momentum the constituent quarks as a whole should move with a momentum p. However, it was discovered that the quarks only accounted for about half the total momentum. Again the quark model was modified. A new particle, the gluon, was introduced into the model as the carrier of the strong force -- the exchange particle that governs the interactions between quarks.

There are eight types of gluons each of which is a superposition of color and anti-color states. Therefore, when a quark emits or absorbs a gluon, the quark changes color but not flavor. Within a hadron the quarks are constantly emitting and absorbing gluons and constantly changing color. This process describes the way gluons set up the strong field in which constituent quarks are held together within a hadron. The study of quark-gluon interactions is called Quantum-Chromodynamics (QCD).

The fact that gluons carry color charge accounts for the asymptotic freedom of quarks. Unlike the electromagnetic force, which weakens with distance, the strong force grows stronger with distance. We shall consider two of the effects that influence the behavior of the strong force: screening and camouflage.

Screening in QCD is analogous to screening in Quantum-Electrodynamics (QED). In QED when an electron enters a region of space populated by molecules that have positively and negatively charged ends, it will polarize the molecules. It, in effect, screens itself in positive charge. The result is a reduction of the effective charge of the electron that decreases with distance. One would think that the screening effect would not occur within a vacuum, where there are no molecules to polarize. This, however, is not true. According to the Heisenberg uncertainty principle, a vacuum is never empty; in fact, energy in the vacuum can increase as it is investigated on progressively shorter time scales. Because of this, particles can violate the conservation of energy for brief instants, materializing in and out of existence. These particles are referred to as virtual particles. Therefore, electric charge in a vacuum will be screened by virtual particles and effectively reduced at large distances. A QCD vacuum is populated by virtual quarks and anti-quarks, often called sea quarks. When a quark is introduced into the vacuum, the virtual quarks harboring contrasting color charge will be attracted to the quark while those harboring like color charge will be repelled. The result is the same as in QED: the quark's color charge is screened by virtual quarks.

The electromagnetic force between charges in QED is due to the exchange of neutral photons, while in QCD the strong force between colored quarks is due to the exchange of color-charged gluons. Because gluons carry color, it changes the nature of the force. Quarks are constantly changing color by emitting and absorbing gluons. The color-charged gluons spread to considerable distances, spreading the color-charge throughout space. In close proximity, the gluon propagation serves to camouflage the quark that is the source of the color-charge. Consequently the color-charge felt by a quark of another color will diminish as it gets closer to the first quark. In this way quarks are permanently confined and asymptotically free, or in other words, at large distances the force between quarks is great, but at small distances (<1 fm) the force is next to zero.

1.2 Quark Deconfinement and the Quark Gluon Plasma

In nuclear matter under normal conditions, quarks are bound within individual hadrons. These hadrons do not overlap; they occupy a volume to themselves. At higher densities, however, hadrons can be compressed in such a way that they overlap and the constituent quarks of one hadron can not be distinguished from that of another. In such a state the idea of individual hadrons does not apply. This new state of matter is called a quark-gluon plasma (QGP). In this state quarks and gluons, deconfined from their hadronic bounds, may freely move about in a volume that is several hadronic volumes in size. Currently it is believed that the QGP existed in the early stages of the development of the universe and may presently exist in the cores of dense neutron stars. In a high-energy particle physics experiment it is estimated that beam energies of 10-100 GeV per nucleon would be needed to recreate this phase of matter under terrestrial conditions.

A head-on collision of two nuclei in a colliding beam experiment appear as two Lorentz contracted disks traveling toward each other (see figure 1.1).

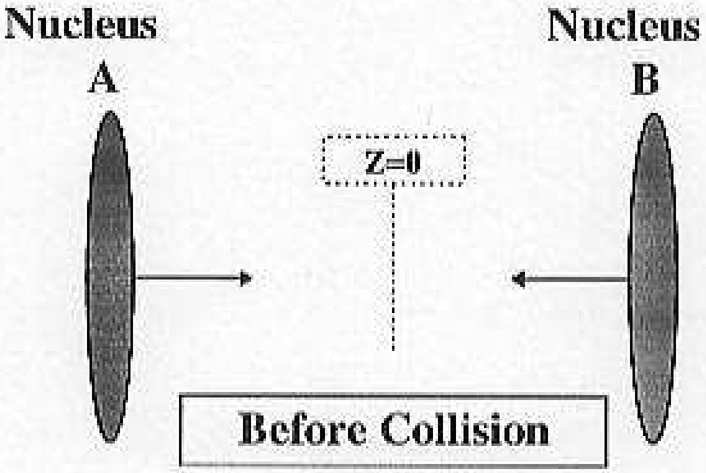


Figure 1.1
Configuration of two colliding nuclei A and B.

They meet at position z = 0 and at time t = 0. At this point (z,t) = (0,0), the particles overlap and a region of extremely high-energy density is created. This region of high-energy density is referred to as the interaction region (see figure 1.2). It is in the interaction region that the formation of a QGP is likely to occur.

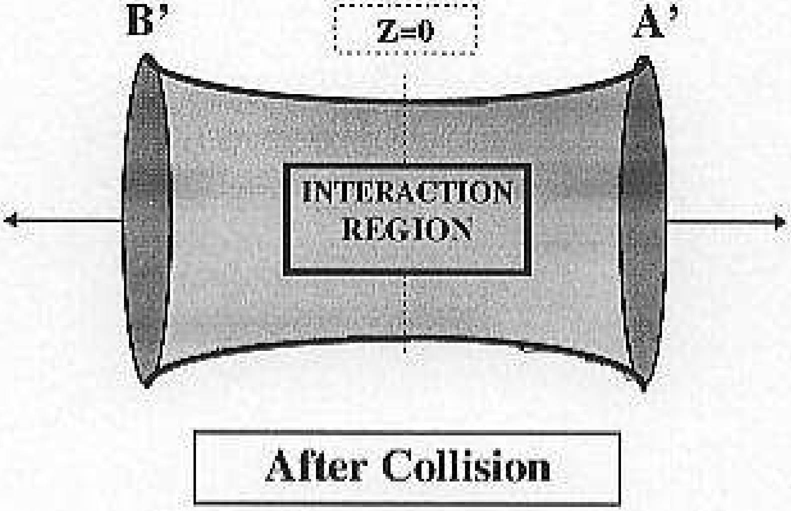


Figure 1.2
The configuration after collision. Energy is deposited at z = 0.

The evolution of a nucleus-nucleus collision in space-time is represented in figure 1.3.

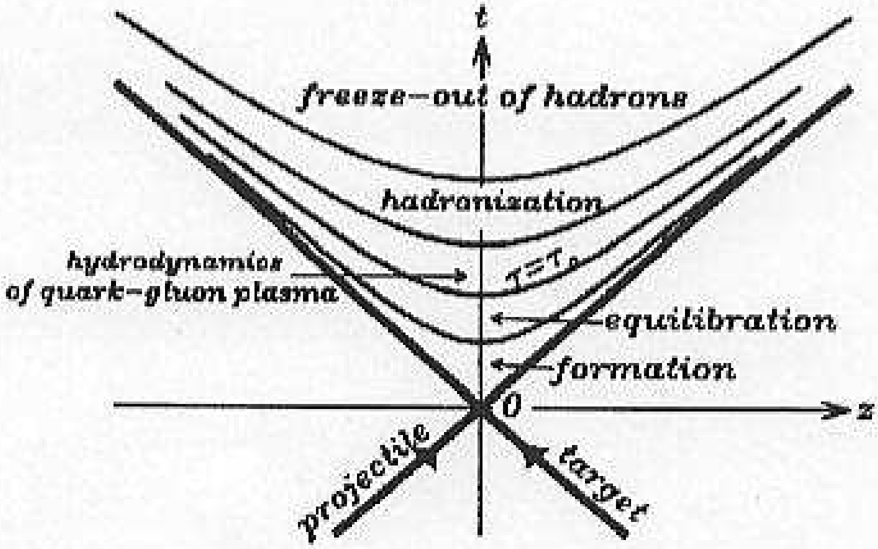


Figure 1.3
The evolution of the nucleus-nucleus collision on the space-time axis [WO 94].

There are two types of collisions in which a high energy-density region can occur: one in which the initial nuclear matter is stopped, called the "stopping" regime, the other in which the initial nuclear matter is slowed but not stopped, often called the “transparent” regime. In the first situation baryons in the two colliding beams are completely stopped by each other. The result is baryon-rich matter in the middle of the interaction region. If a QGP were to form this type of collision it would be hard to detect because of complicated scattering due to the high baryon content.

If the beams have sufficient energy, the initial baryons will not be completely slowed down, but recede from the interaction region leaving it baryon-free. High energy-density is still deposited in the interaction region and may form a baryon-free QGP. The theoretical model calculations are much more straightforward with this type of QGP, and hence it is of great interest.

The hope is that a QGP will be formed at the Relativistic Heavy Ion Collider (RHIC).

1.2.1 Signatures of the Quark Gluon Plasma

Relativistic heavy ion collisions present the possibility of producing the high energy densities needed for the formation of a QGP. However, the detection of this phase of matter is obscured by the number of particles produced in the collisions, about 4000 particles per event. Because of this and the short lifetime (~ 10^{-24}s) of the QGP, one must look for indirect evidence of the formation.

1.2.1.1 Photon and Dilepton Formation

In a QGP state, quark-gluon interactions can result in the production of real photons, while quark-anti-quark interactions can produce lepton pairs. These particles interact weakly and are expected to escape the plasma without rescattering. Unfortunately, the signals of these two mechanisms can be degraded by high background levels. The photon energy spectrum may provide a signal of the plasma formation, however large background sources (i.e. photons from π^0 and η decay) make it difficult to quantitatively predict photon energies originating from the interaction region. Likewise, dilepton signals can be obscured by background signals, such as pion annihilation which is followed by lepton pair production.

1.2.1.2 J/Ψ Suppression

A J/Ψ meson contains a bound charm quark-anti-quark pair ($c\bar{c}$). Theory predicts that in a QGP the quark-anti-quark bound state of the J/Ψ will dissolve due to Debye screening, resulting in a low density of $c\bar{c}$ pairs in the plasma. In this scenario, finding a charm quark-anti-quark pair in close proximity would be very unlikely, due to the comparatively large number of lighter quarks. Hence, upon hadronization, the charm would emerge from the QGP in the form of the D-mesons.

1.2.1.3 Momentum Distribution of Jets

The momentum distribution of particle jets in central nucleus-nucleus collisions is another possible indication of the formation of a QGP. In a central collision, a large fraction of the energy (in jets of particles) comes from the interaction region perpendicular to the direction of the initial beam travel. It is felt that quarks and gluons will lose less energy passing though a plasma state. This would result in a larger average transverse momentum of particle jets coming from a QGP compared to those from a hadron gas [DO 91].

1.2.1.4 Enhancement of Strange Particle Production

In a relativistic heavy ion collision the resulting fireball initially contains the up and down quarks of the colliding baryons. The premature fireball is said to contain little strangeness. Strangeness is a quantum number associated with the presence of one or more strange quarks in a system. As the fireball matures, strange quarks have to be created as quark-anti-quark pairs, in order to conserve strangeness. It is much easier to create a $s\bar{s}$ pair in a collision that produces a QGP than it is in a collision that produces a hadron gas of the same temperature. This is because in a QGP the quarks and gluons are deconfined, and it takes just the rest mass of the two strange quarks (i.e. $2m_s = 400$ MeV) to make a $s\bar{s}$ pair. In a collision that produces a hadron gas it takes 570 MeV or 2.23 GeV to produce a strange or anti-strange baryon respectively, and up to 2.64 GeV for the production of multi-strange baryons. Strangeness production is also enhanced in a QGP because of the different reaction mechanisms. In a hadron gas, $s\bar{s}$ pairs are produced by the interactions between hadrons. A QGP has gluon interactions as well as quark interactions. In a QGP two real gluons can produce a pair of strange quarks and a quark anti-quark pair can annihilate, making a virtual gluon which converts to a pair of strange quarks. Theoretical calculations have shown that the production rate of $s\bar{s}$ pairs in the QGP should be 10-30 times greater than in a hadron gas.

1.3 Relativistic Heavy Ion Collider (RHIC)

The Relativistic Heavy Ion Collider (RHIC), located at Brookhaven National Laboratory (BNL), is designed to accelerate heavy nuclei to an energy of 100 GeV per nucleon. Construction of the collider started in January 1991 and is expected to finish in early 1999. The collider itself consists of two concentric rings of superconducting magnets built in an existing ring tunnel of approximately 3.8 km in circumference located in the northwest section of the BNL site. The existing accelerator complex at BNL is made up of the Tandem Van de Graaff accelerators, the Booster synchrotron, and the Alternating Gradient Synchrotron (AGS), interconnected with beam transfer lines, and will serve as the injector for the RHIC. An aerial view of the RHIC complex is shown in figure 1.4.

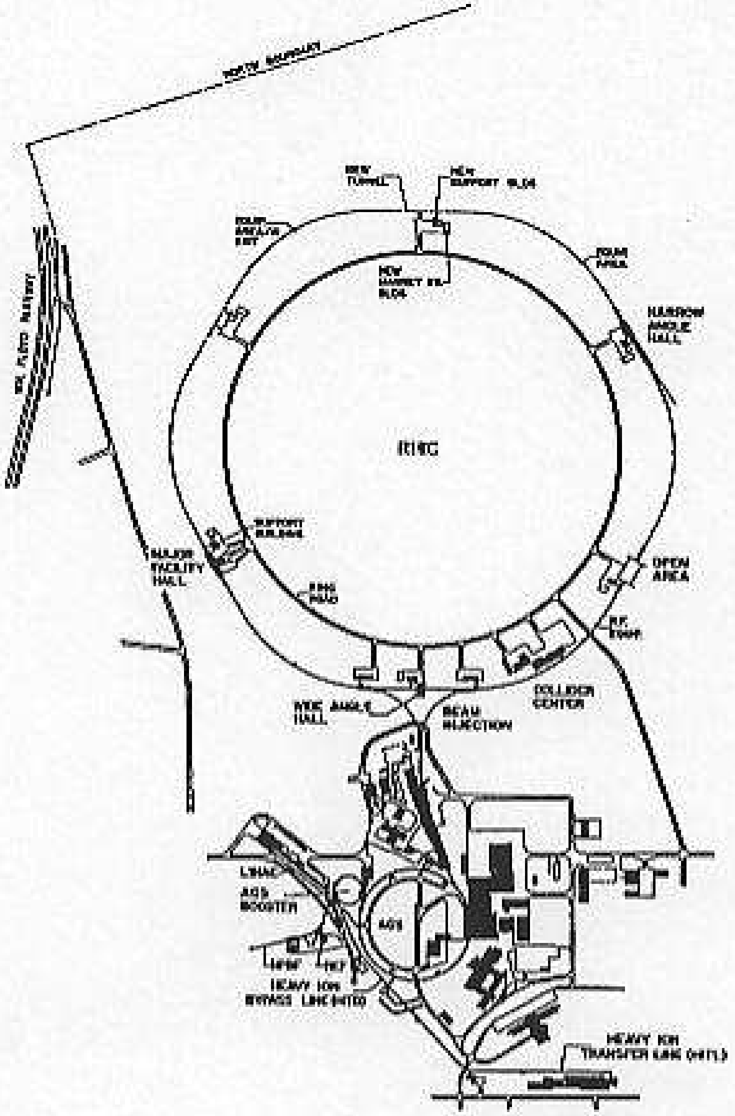


Figure 1.4
Layout of the RHIC complex [LU 93].

The Tandem Van de Graaff accelerator provides the initial acceleration. The gold ions, having been stripped of 12 electrons, exit the Tandem Van de Graaff accelerator at a kinetic energy of 1 MeV per nucleon. Upon exiting the Tandem Van de Graaff, the ions are further stripped to the +32 state. The ions then travel through the heavy-ion transfer line and are injected into the Booster synchrotron. The Booster synchrotron increases the kinetic energy of the ions to 95 MeV per nucleon. In the Booster-to-AGS transfer line the ions are stripped to the +77 state and then injected into the AGS. The AGS accelerates the ions to 10.8 GeV per nucleon and transfers the ions to the RHIC collider through the AGS-to-RHIC transfer line where the final stripping from +77 to +79 takes place. The RHIC accelerator will perform the final acceleration to 100 GeV per nucleon. The ion acceleration scenario is shown in figure 1.5.

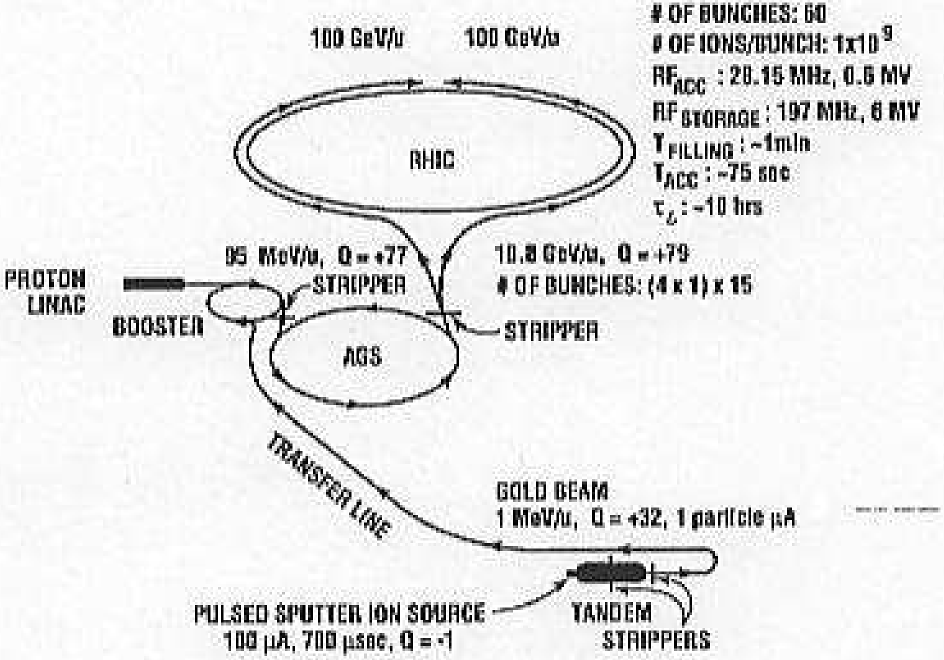


Figure 1.5
RHIC acceleration scenario for gold [LU 93].

1.4 Physics of the Solenoidal Tracker At RHIC

The Solenoidal Tracker At RHIC (STAR) will search for signatures of a quark-gluon plasma formation and investigate the behavior of strongly interacting matter in high energy density collisions. Emphasis will be on the correlation of many observables on an event-by-event basis [ST 92]. Measurement of global observables at an event-by-event level is made possible by the expected high density of charged particles (about 4000 particles per event) produced in the nucleus-nucleus collisions at the Relativistic Heavy Ion Collider (RHIC). The global observables -- such as average transverse momentum, temperature, flavor composition, collision geometry, energy- and entropy-density -- are expected to fluctuate from one event to another in the vicinity of the hadron gas-QGP phase transition. Thus the STAR experiment will be sensitive to the behavior of these observables as a function of energy density. In the absence of definitive signatures of a QGP, it is essential that such correlations be used to identify special events and possible signatures. This requires an elaborate and flexible detection system that can simultaneously measure many global observables.

STAR is capable of tracking and identifying several thousand particles in a typical nucleus-nucleus collision at RHIC. Tracking measurements coupled with electromagnetic calorimetry and high-momentum particle measurements, will allow the study of hard QCD processes such as jet, mini-jet and hard photon production.

The experiment plans to directly identify and analyze the momentum of charged particles (π^+, π^-, K^+, K^-, p, $\bar{p}$, d, $\bar{d}$), as well as neutral and charged strange particles (K^0, φ, Λ, $\bar{\Lambda}$, Ξ^-, Ω^-) via charged-particle decay modes.

1.4.1 Particle Spectra

At STAR the transverse momentum (p_t) distribution and the average transverse momentum $<p_t>$ for pions and kaons can be determined on an event-by-event basis. This allows individual events to be characterized by temperature. Events with high temperature may be a signature of QGP formation. Figure 1.6 displays two spectra generated by the Monte Carlo method from Maxwell-Boltzmann distributions with T = 150 and 250 MeV, each containing 1000 pions. The slopes (temperature) of the spectra (also shown) can easily be noticed at the single event level.

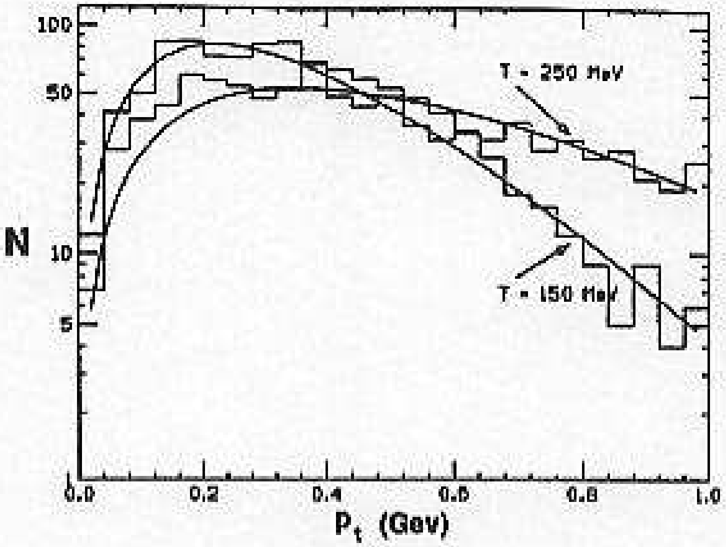


Figure 1.6
Simulation of the p_t spectrum for one event generated using a Boltzmann distribution of 1000 pions. The histograms correspond to single events generated with T = 150 MeV and 250 MeV. The curves are fits to the histogram using a Maxwell-Boltzmann distribution [ST 92].

1.4.2 Flavor Composition

It is believed that at RHIC energies the various stages of the collision, i.e. thermalization, expansion, and hadronization, will be reflected in the abundance and spectral distribution of particles containing heavy quarks. Measurement of the particle yields and spectra as a function of the reaction volume may lead to a decisive signature for the formation of the QGP [ST 92].

Information on the relative concentration of strange and non-strange quarks in an event will come from the measurement of the K/π (kaon to pion) ratio. This has been suggested as a diagnostic tool to identify hot, dense matter [ST 92]. Measurement of the production rate of the phi-meson, an $s\bar{s}$ pair, places a more stringent constraint on the origin of the observed flavor composition than the K/π ratio, and is expected to be more sensitive to the presence of a QGP . It is also thought that in the presence of a QGP there may be a shift in the phi-meson invariant mass [ST 92]. The K/π ratio and the phi-meson yield will be measured on an event-by-event basis with enough accuracy to classify events for correlations with other observables.

1.4.3 Global Observables

The appearance of any dynamical fluctuations in a narrow range of conditions has always been an indicator of a phase transition. Within the realm of high-energy particle physics however, such fluctuations can be seen in individual events if the statistics are large enough to overcome the uncertainties. The large transverse energies produced at RHIC will allow the STAR detector to do event-by-event measurements of fluctuations in particle ratios, energy-density, and entropy-density as a function of transverse momentum and rapidity. These fluctuations may indicate a phase change, i.e. the hadronization of a QGP [ST 92].

1.4.4 Electromagnetic and Hadronic Energy

Correlation and fluctuation analysis are considerably improved by including electromagnetic (EM) and charged particle data. One third of the energy is EM energy and is measured by calorimetry. The hadronic energy will be measured by charged-particle tracking. What is more important, a change in the amount of EM energy versus charged-particle energy is one of the signatures of the QGP and other new physics [ST 92].

1.4.5 Parton Physics

The study of hard QCD processes uses the propagation of quarks and gluons as a probe of nuclear matter, hot hadronic matter, and quark matter. The RHIC collider will be the first facility to provide an environment with sufficient energy to accurately measure the rates of detectable parton debris (jets, high p_t particles, and direct photons) from hard scattering and parton-parton interactions. Calculations have shown that the propagation of quarks and gluons through matter depends on the properties of the medium. For example, it has been suggested that there is a noticeable energy loss associated with propagating partons as the energy-density of the medium increases, particularly if the medium passes into a QGP.

Chapter 2. Solenoidal Tracker At RHIC

2.1 Implementation Plan

The initial phase of the STAR detector will include the following subsystems:

- Solenoidal magnet
- Time Projection Chamber (TPC)
- Silicon Vertex Tracker (SVT)
- Electromagnetic Calorimeter (EMC)
- Trigger
- Data Acquisition System (DAQ)

and the associated electronics for each of these subsystems. These subsystems are anticipated to be on-line at RHIC start-up in mid 1999. Upgrades, such as Time-of-Flight Detectors (TOF), Forward Time Projection Chambers (FTPC), and more sophisticated triggers, will take place as funds become available. The detector subsystems will be implemented as shown in figure 2.1.

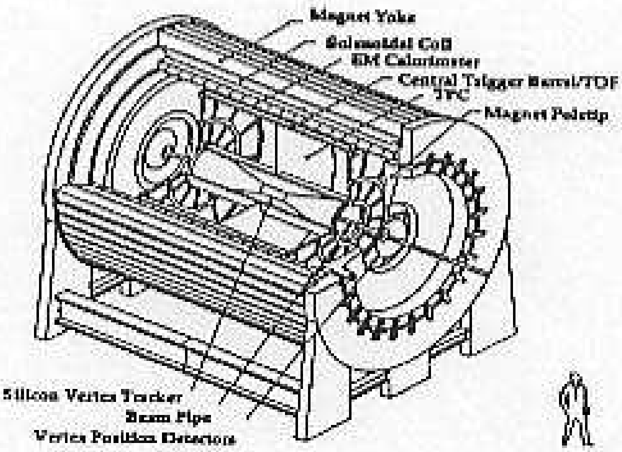


Figure 2.1
Configuration of the baseline STAR detector.

2.2 Experimental Setup

2.2.1 Time Projection Chamber

2.2.1.1 Introduction

Over the past twenty years the Time Projection Chamber (TPC) has become one of the most effective tracking devices. It has a large variety of possible applications ranging from searches for rare or exotic processes at very low energies to the study of strangeness production in the high multiplicity environment of high energy heavy ion collisions and to charged-particle tracking at the highest energy electron-positron colliders [LO 92].

As shown schematically in figure 2.2, the basic constituents of a TPC are a large gas drift space and a high voltage plane which sets up a uniform electric field terminating at Multi-Wire Proportional Chambers (MWPC).

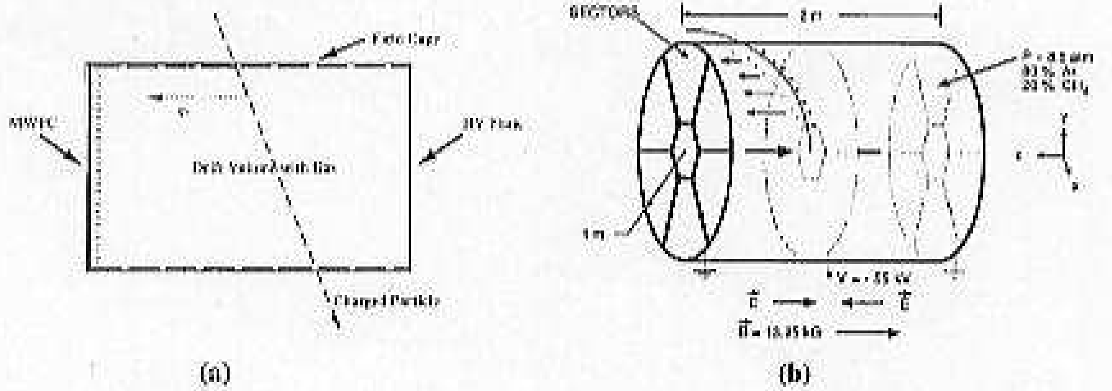


Figure 2.2
In both a rectangular (a) and a cylindrical (b) TPC the basic constituents are:
a large drift space - a high voltage plane - and a Multi-Wire Proportional Chamber [LO 92].

The shape and size of a TPC is application-specific. A rectangular TPC is most likely to be used in a fixed target experiment, placed downstream of the target. A cylindrical TPC enclosing the interaction region is best suited for beam-on-beam collisions.

When a charged particle passes through the gas drift space it leaves behind a string of electrons and ions formed by the ionization process. The electrons are accelerated by the electric field to the wire chambers where they produce a signal on the sense wires. The position of the signal is induced on a cathode pad plane which is positioned in back of the sense wires. Combining the position signals with the drift-time of an electron in the TPC gas, the TPC provides three-dimensional coordinate measurements of the particle as it traverses the drift volume (figure 2.3). The wire signals can also be used to measure the specific energy loss, dE/dx, of the ionizing particles, thus adding particle identification capabilities [LO 92].

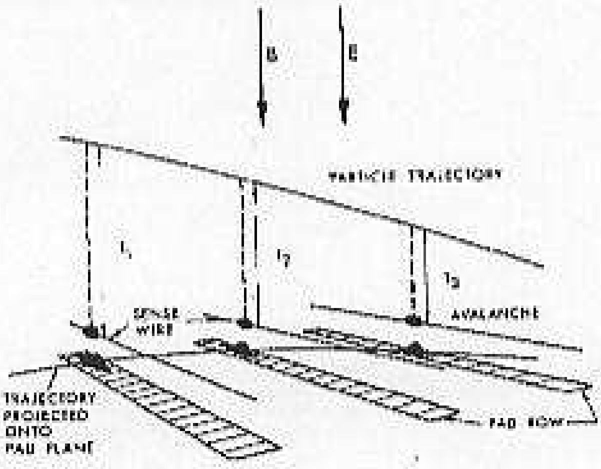


Figure 2.3
Schematic of the detection process of charges on the wires and pad plane of a TPC end-plate [LO 92].

The TPC is usually placed inside a uniform magnetic field which is directed parallel to the electric field. This allows measurement of particle momenta from track curvatures and improves the spatial resolution of the track signals by decreasing the transverse diffusion of the electrons during the drift process.

The TPC has many advantages over other types of tracking devices. The device can cover almost the full solid angle and measures truly three-dimensional coordinates with typical resolutions of about 200 mm orthogonal to the drift direction and often much better than 1 mm along the drift direction, leading to excellent momentum resolution [LO 92]. The performance of the TPC is that of a bubble chamber, but relying entirely on electronic information instead of the numerous pictures that are associated with bubble chamber data. Pattern recognition is therefore simple and efficient and allows experimenters to deal with high track densities [LO 92].

The most significant disadvantage of the TPC is its slow readout time due to large drift lengths. Also, high rates of particle collisions and accelerator background radiation can lead to space-charge pile-up which will distort the electric drift field and affect momentum resolution. The TPC three-dimensional coordinate data is provided by a highly complicated and sophisticated electronic readout system. Thousands of wires and tens of thousands of cathode pads have to be read out in time slices of typically 100 ns. All the information from the sense wires and pads have to be amplified and digitized with expensive fast circuitry. Furthermore, the readout of large numbers of channels, each with a large number of time slices, introduces huge amounts of data which must be passed to a data acquisition system.

2.2.1.2 Basic Time Projection Chamber Components

In this section the basic components required for a TPC are described. The TPC can be applied to many different experimental situations which ultimately influence the final design and implementation of the detector. Requirements change considerably from one application to the next, but certain components are found in most TPC's.

The largest part of a TPC is the drift region. The drift region is a volume of detector (radiator) gas enclosed by a field cage which creates an electric field. In the case of a rectangular TPC used in fixed-target experiments, the drift space is terminated on one side by a high voltage plane and the other side by a Multi-Wire Proportional Chamber (MWPC). For colliding beam experiments the TPC is cylindrical in shape, positioned so that the beam travels along the main axis of the cylinder. In a cylindrical TPC, the high voltage plane is usually placed in the middle of the TPC, orthogonal to the beam, and the drift volume is terminated with MWPC's on both ends of the detector. Charged particles cross though the drift volume creating tracks of electrons. These electrons drift to the MWPC under the influence of the electric field. Arriving at the MWPC, the electrons generate signals around the sense wires and induce a signal on the cathode plane behind the sense wires.

Multi-Wire Proportional Chamber

In a Multi-Wire Proportional Chamber (MWPC), the principle of a proportional counter is put to use in a detector with a large area. A basic MWPC is composed of a cathode plane, a sense/field wire plane, a cathode (or shielding) wire plane and a gating grid. A schematic of the arrangement of these planes is shown in figure 2.4.

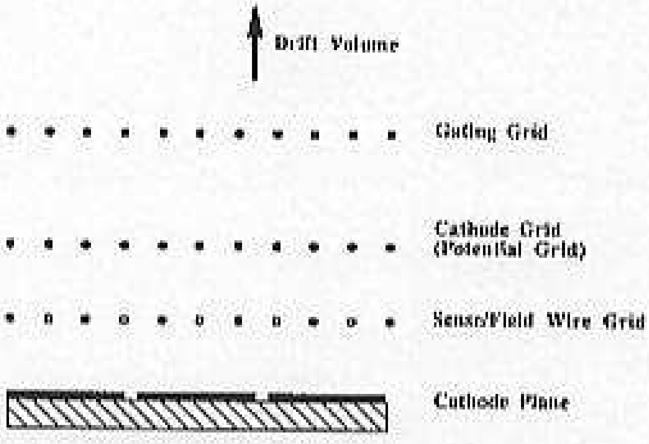


Figure 2.4
The wire arrangement of a Multi-wire Proportional Chamber [LO 92].

The cathode plane is a conductor-coated plate (usually copper clad onto a fiber glass-reinforced epoxy plate) that has been segmented into small pads. The segmentation is performed by means of a photochemical etching process, like used for printed circuit boards, or by milling grooves into the plate. The configuration of the pads, as well as the size of the pads, are application-specific. The cathode plane is where the signal from the ionizing particle is induced. Each pad is connected to a preamplifier. One preamplifier may read out several thousand pads and is the first step in the readout of the data.

The sense/field wire grid contains two kinds of wires, each having a different function. The grid is made by alternating anode (sense) wires and field wires. The anode wires are held at positive high voltage with respect to the surroundings and the field wires serve to electrically decouple the anode wires [LO 92]. The field wires can be used to provide test pulses to both the sense wires and the pads. When the electrons emerge from the drift region, they travel toward the nearest anode wire. The electric field near the anode wires is that of a cylindrical capacitor. If the primary electrons from the drift region gain enough energy in this electric field, secondary ionization occurs (gas amplification). The secondary electrons repeat the process leading to the formation of an avalanche around the anode wire as shown in figure 2.5. If the signal created by the avalanche on the wire is to be used (e.g. for dE/dx information), each of the sense wires is connected to a preamplifier [LO 92]. The cathode wire plane closes the wire chamber from the drift volume and is typically held at the same potential as the cathode plane.

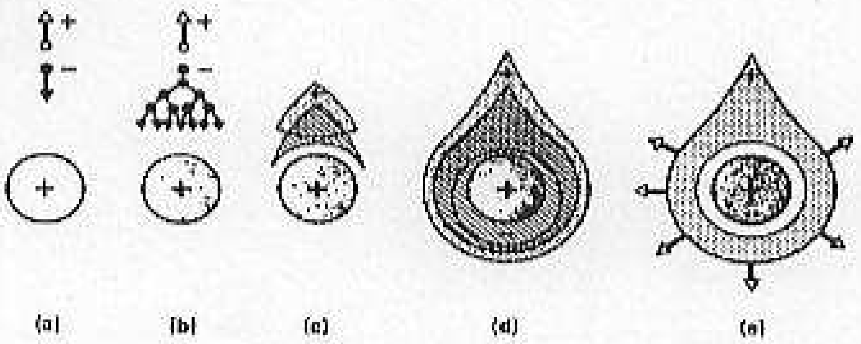


Figure 2.5
Time development of an avalanche near to an anode wire in a proportional chamber. (a) Primary electron moving towards the anode. (b) The electron gains kinetic energy in the electric field and ionizes further atoms; multiplication starts. (c) The electron and ion cloud drift apart. (d), (e) The electron cloud drifts towards the wire and surrounds it; the ion cloud withdraws radially from the wire [KL 86].

The gating grid, located between the MWPC and the drift volume, performs an essential duty. A uniform electric field is of great importance in the TPC. The uniformity of this field can be disrupted by the presence of space charge in the form of ions in the drift region. The positive ions created in conjunction with the electrons in the primary ionization process represent only a small electric charge and are of no consequence. In contrast, the number of positive ions formed during the avalanche process is on the order of 10^2-10^4. These positive ions can seriously affect the uniformity of the electric field. A large fraction of these positive ions are nullified by the cathode plane, field wires and the cathode wire grid, but some reach the drift volume building up space charge and therefore distorting the electric field. The gating grid overcomes the problem of space charge buildup. When the gating grid is in a "closed" state it prevents electric charges from traversing the grid. It is made up of a layer of wires that are alternatively connected together. When the gate is in the "open" state it is at a potential similar to its location and does not disturb the original field. In the closed state a potential difference is applied between adjacent wires such that a dipole field is created. A gating grid that is closed prevents the positive ions created in the avalanche process from entering the drift region as well as stopping the primary electrons arriving from the drift region before they enter the MWPC. Therefore the gating grid can be used to choose whether an event is readout (gate open -- signals formed in MWPC) or whether it is to be ignored (gate closed -- no signals formed in MWPC). Figure 2.6 shows the field configuration for both states of the gating grid.

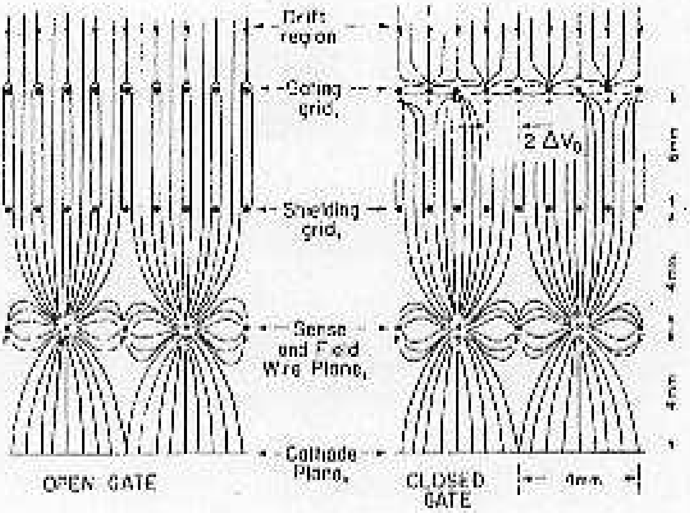


Figure 2.6
The electric field configuration of a Multi-wire Proportional Chamber with the gate in the open state and in the closed state [LO 92].

Field Cage

The primary purpose of the field cage is to provide a uniform electric drift field. In addition to this, the field cage also serves as a gas containment vessel. As mentioned before, the field cage is terminated on one side by a high-voltage plane and on the other by a MWPC (rectangular TPC), or terminated on both sides by MWPC with a high-voltage plane in the center (cylindrical TPC). To produce the uniform drift field, electrodes that are connected together by a precision resistor chain are mounted the walls of the field cage parallel to the direction of drift. These resistors create linearly spaced equipotential lines perpendicular to the drift direction, hence the uniform electric field.

The construction of a field cage involves resolving two contradictory requirements. The walls of the field cage should be as thin as possible to minimize the amount of material the particles have to traverse in order to reach other detectors. At the same time the field cage must be mechanically stable, self-supporting, and have enough insulator to cope with the high voltage. The electrodes covering the walls of the field cage should leave as little insulating surface as possible exposed to the drift region because surface charges building up there may distort the electric field [LO 92]. Figure 2.7 shows a partial cross-section of the STAR field (voltage gradient) cage. The construction of the voltage gradient cage consists of a double layer of electrodes (0.035-mm Cu) separated by 0.075-mm Kapton film as shown in the figure. In the STAR implementation the outer field cage is in close proximity to other subsystems held at ground potential, therefore a mylar insulator and a grounded aluminum skin are wrapped in layers around the outside of the voltage gradient cage, electrically shielding it from its surroundings.

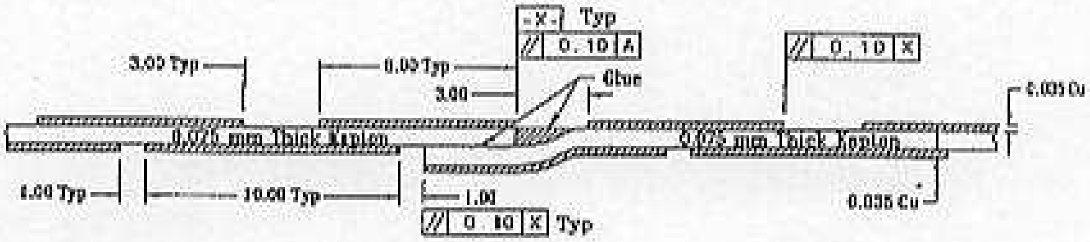


Figure 2.7
Details of the Outer-Field Cage construction [ST 92].

Electronics

Each pad on the cathode pad plane is connected to a chain of electronics. This chain consists of a preamplifier, a shaping amplifier and digitizing electronics which measure the amplitude and arrival time of the MWPC signal. In the STAR implementation, the charge from the cathode plane pads is amplified and integrated by a low noise, low capacitance preamplifier. The time history of the signals is stored on a 512-sample switched-capacitor array (SCA) analog memory. After the total event is stored on the SCA, it is clocked out to an analog-to-digital converter (ADC) for digitization. At this point the digitized data are transmitted to a readout board for data collection and zero suppression before being transmitted to the Data Acquisition (DAQ) subsystem.

To measure the response of the electronics, calibration pulses must be fed to the preamplifiers. This can be accomplished by pulsing the field wire grid in the MWPC. Pulsing the wire grid should induce identical pulses on all the pads. Knowing the input and induced pulse heights, one can correct for differences in electronics gain.

2.2.1.3 The STAR Time Projection Chamber

The TPC is the main detector of the STAR experiment and is capable of handling high-multiplicity events. It determines individual particle momenta by tracking them through the solenoidal magnetic field. By making multiple energy-loss measurements, particle identification is achieved. The TPC gas volume (approximately 50 cubic meters) is bound by coaxial field cage cylinders and is terminated with endcaps which house multi-wire proportional chambers (MWPC) at each end. A high-voltage central membrane creates an electric field so that ionization electrons, that were produced by a particle traversing the TPC, drift toward the endcaps. The ionization electrons are amplified by gas (avalanche) amplification at the anode wires of the MWPC's. Image charges are induced on the pad plane located behind the anode wires and are read out as a function of time. The drift time of the ionization electrons furnish one space coordinate, while the induced signals on the pad plane provide two-dimensional space coordinates of a particle's path through the TPC.

Each endcap is divided into twelve supersectors, each of which is divided into a inner and outer sector as shown in figure 2.8. Each sector will be equipped with electronics designed to read out the information produced in the TPC.

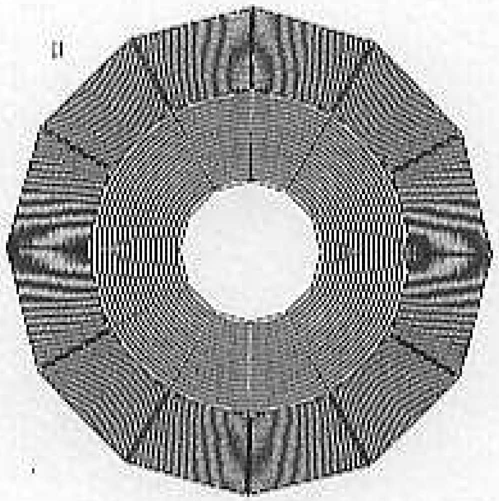


Figure 2.8
A diagram of the TPC end-cap indicating the locations for the pad plane sectors and pad rows [ST 92].

2.2.2 Solenoid

The TPC detector subsystem requires a uniform magnetic field over its entire volume to measure the momentum of charged particles produced in the collisions at

RHIC. At STAR a 0.5-Tesla solenoidal magnet will be used to facilitate these measurements. The iron return yoke and shaped pole pieces shown in figure 2.9, carry the magnetic flux generated by the solenoid and shape the magnetic field to the required uniformity at the ends of the detector.

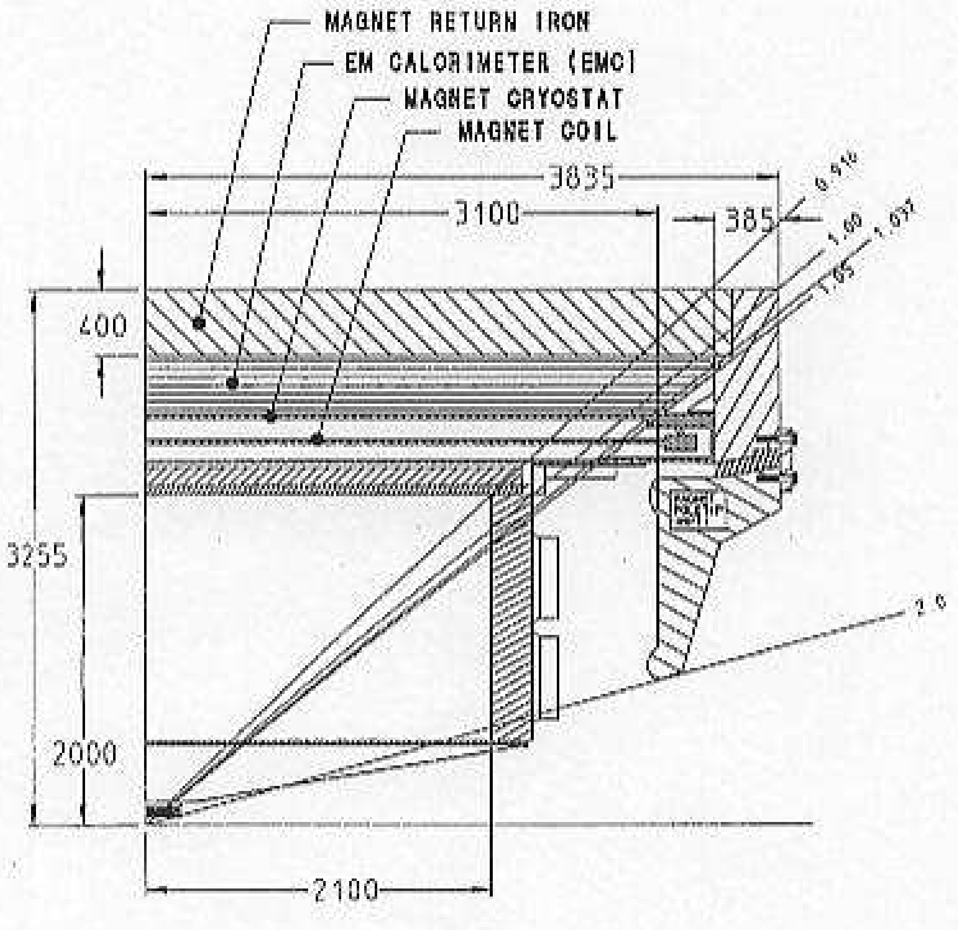


Figure 2.9
Magnet iron return yoke and shaped pole pieces [ST 92].

2.2.3 Silicon Vertex Tracker

The SVT greatly improves the tracking capabilities of the STAR detector below transverse momenta of 150 MeV/c and extends the particle detection scheme to short-lived neutral and charged strange particles. Since strange quarks will be the most frequently produced heavy quarks, strange-particle detection is one of the primary goals of STAR. The various types of strange particles are detected and identified by different methods. Charged long-lived kaons will be identified by measuring their energy loss (dE/dx) in both the TPC and the SVT. Short-lived particles (K^0, Λ, anti-Λ, Ξ^-, Ω^-) are detected via their weak decay into charged particles. Identification of strange-particle decays requires the separation of their decay tracks from those tracks that originate from the primary interaction point. The separation technique requires precise information on the track position close to the interaction point. The SVT is an excellent tool for extending the acceptance of the TPC to low transverse momentum (p_t) particles. Simulations of central Au+Au events show that the transverse momentum distribution of emitted particles at midrapidity will in general be peaked around 300 MeV/c [ST 92]. The lower TPC limit for fully efficient track reconstruction is 150 MeV/c. A considerable part of the event will be measured by inclusion of tracking in the SVT alone.

The STAR SVT is based on Silicon Drift Detector (SDD) technology. SDD's use an n-type silicon wafer as a substrate. Field-shaping electrode strips made of p-type material are symmetrically implanted on both sides of the substrate. A high voltage gradient is applied to the electrodes, depleting the silicon and thus creating a deep potential well that attracts free electrons produced by the passage of charged particles through the detector. The electrodes are connected to a voltage divider network that produces an electric field parallel to the surface of the wafer. This field transports the electrons toward the edge of the detector where the signals are read out on a segmented anode.

The SVT will consist of 216 SDD's mounted in three concentric barrels. Figure 2.10 shows a drawing of the SVT layout.

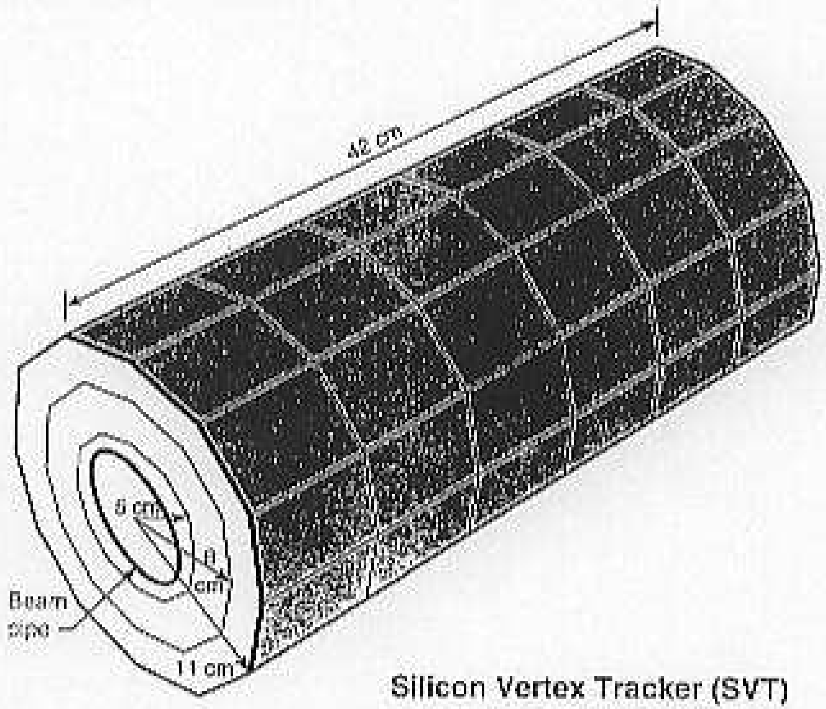


Figure 2.10
Conceptual view of the STAR SVT [ST 92].

2.2.4 Electromagnetic Calorimeter

A calorimeter is a block of matter which intercepts a particle, and is of sufficient thickness to cause the particle to interact and deposit all its energy inside the detector volume. Electrons and photons interact with the calorimeter material principally through bremsstrahlung and pair-production respectively. These processes produce secondary photons and electrons which interact in a similar manner producing a cascade or "shower" of secondary particles. Calorimeters use scintillators to convert the excitation of a transparent material caused by an ionizing particle into visible light. The scintillator also transports this visible light to the photocathode of a photomultiplier tube. When the light from the excitation strikes the photocathode of the tube it liberates electrons. An electrode is used to collect, focus and accelerate the photo-electrons from the cathode to the first dynode of the phototube. The dynode is an electrode made from a material with a high coefficient of secondary electron emission. The emission of three to five secondary electrons can be achieved for one incident electron with a kinetic energy of 100 to 200 eV. Subsequent dynodes are used to amplify the number of secondary electrons to around 10^{8}. This whole process takes about 40 ns.

The STAR Electromagnetic Calorimeter will measure the total and local E_t using a lead-scintillator sampling calorimeter. This measurement is essential in determining the degree of nuclear stopping which occurs in the collisions at RHIC. Analysis of the transverse energy leads to estimates of the energy-density of the reaction, which may allow further determination of the possibility of QGP formation. It covers the full azimuthal range and a psuedorapidity range of $-1.05\leq\eta\leq 1.05$, and consists of 60 wedge segments of 6 degrees each in ϕ as shown in figure 2.11.

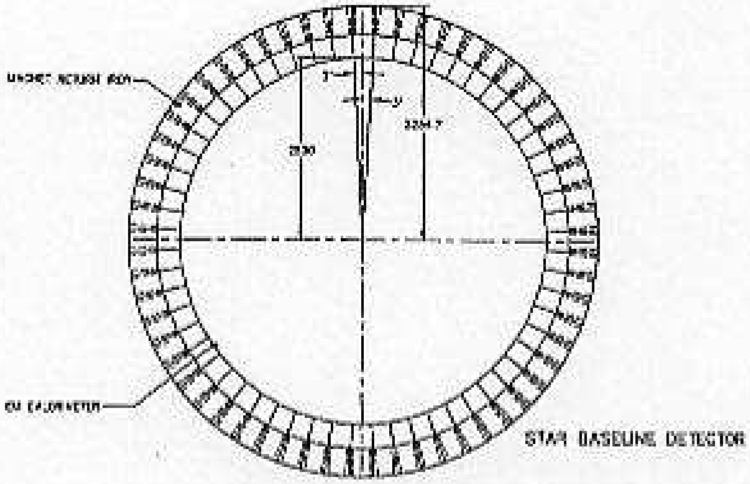


Figure 2.11
An end view of the STAR EMC barrel [ST 92].

The wedges are also broken into 40 towers along the length of the detector. Each tower has 21 layers of 5-mm lead absorber and 3-mm plastic scintillator. There are thus 50,400 pieces of scintillator (60 wedges x 40 towers x 21 layers) of 420 different shapes. Each tile will be read out with two optical fibers that go to photomultiplier tubes.

2.2.5 Time-of-Flight

The purpose of the TOF detector is to extend the particle identification capabilities of the TPC and SVT to high transverse momenta and to provide a flexible trigger for event selection. The system must measure the flight time of as many charged particles as possible in the central region of the STAR detector with a timing resolution of at least 100 picoseconds. The TOF is a highly segmented cylindrical scintillator/counter mounted on the outside of the TPC.

The system design is a shingle layout, consisting of tiling the cylinder with 7776 single-ended scintillators arranged in 216 trays of 36 scintillators each. A drawing of 108 trays that house the TOF shingle counters covering one half of the length of the experiment is shown in figure 2.12. Each tray positions and supports 36 scintillators and covers 3.3 degrees in ϕ. Each shingle consists of a proximity-mesh dynode phototube optically coupled to a scintillator with a Plexiglas guide. Because of budget constraints the complete TOF detector will be installed only when additional funds become available [ST 92].

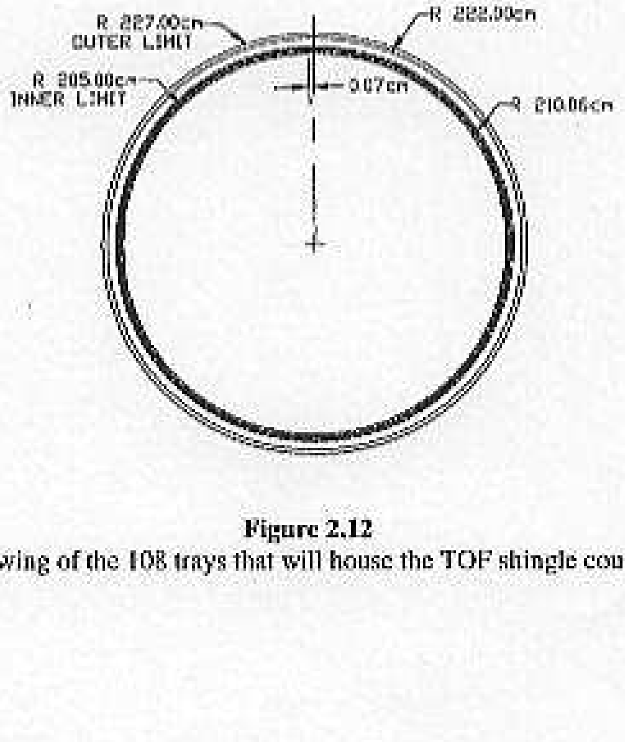


Figure 2.12
A drawing of the 108 trays that will house the TOF shingle counters.

2.2.6 Forward Time Projection Chambers

The purpose of the FTPC's is to extend the kinematic range covered by the experiment. In order to determine primary particle distributions, it is necessary to be able to detect all particles and reject those coming from secondary interactions. These secondary particles are the result of interactions in the beam pipe, the SVT, the TPC and the decay of short-lived particles.

The FTPC is a high-resolution radial TPC with Micro-strip Gas Chamber readout. The FTPC is located inside the STAR magnet, filling the space between the RHIC beam pipe and the inner field cage of the main TPC. Figure 2.13 shows a perspective view of the FTPC.

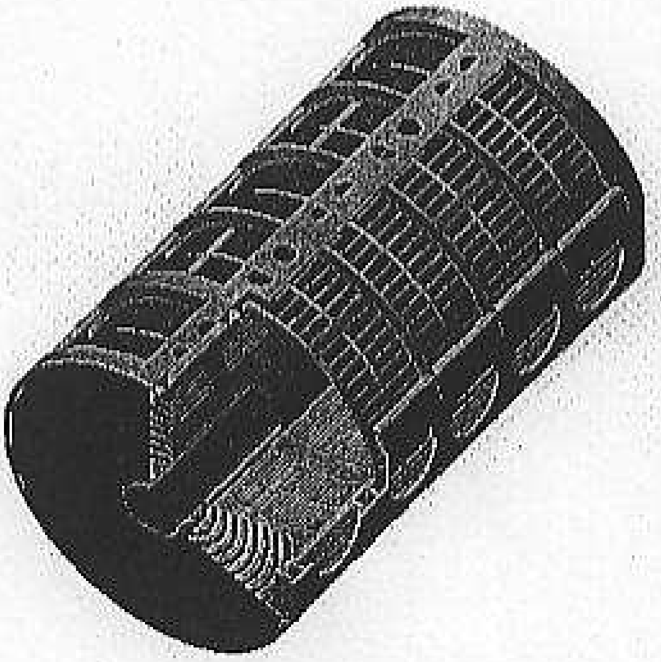


Figure 2.13
Perspective view of the FTPC.

2.2.7 Trigger

Since a triggerable signature of quark-gluon plasma events has not been identified, the STAR trigger is being designed to be flexible enough to allow for continual upgrades as the understanding of such signatures increases. The baseline trigger system will operate with high deadtime until sufficient information is obtained to be able to safely increase the selectivity of the trigger and thus decrease the overall deadtime [ST 92].

The goals of the STAR trigger system are to [ST 92]:

- Detect Au-Au interactions with >95% efficiency
- Detect p-p interactions with >50% efficiency
- Locate an Au-Au collision vertex within ±6 cm within 200 ns
- Detect multiple events occurring within 40 μs
- Select highest 2.5% of multiplicity distribution in < 200 ns
- Select high transverse electromagnetic energy
- Select on basis of the geometry of hit patterns
- Allow simple integration of upgrades to the trigger system

Figure 2.14 is a quadrant view of the STAR detector showing the trigger detectors.

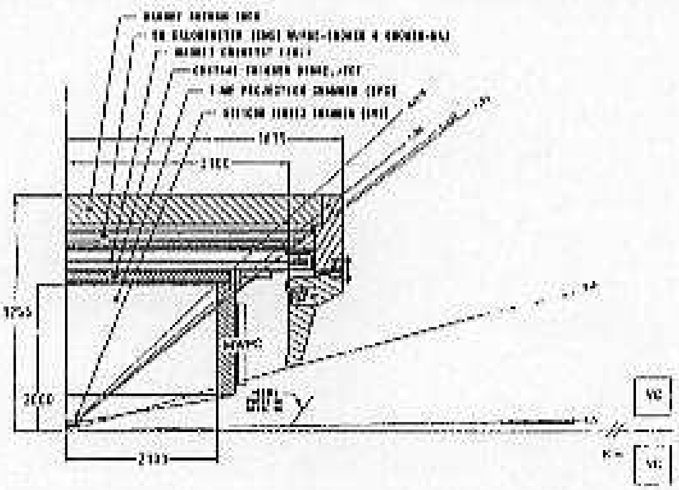


Figure 2.14
STAR trigger detectors [ST 92].

The trigger system is divided into two components. The first comprises the trigger detectors, consisting of a Central Trigger Barrel (CTB), Vertex Position Detectors (VPD), TPC endcap Multi-Wire Proportional Chambers (MWPC), Veto Calorimeters (VTC) and output from the EMC. The second component consists of data processing and the interfaces to the detector subsystems and the Data Acquisition subsystem.

The CTB is a barrel of 200 slats of scintillator on the TPC outer surface that subtends $|\eta| < 1$. These slats are radiators optically coupled to two phototubes.

The VPD contains two scintillators, each placed on either end of the TPC, covering 3.3-3.8 in $|\eta|$. The light signals from the scintillators are read out by photomultiplier tubes and are sent to specially designed fast circuitry to determine the vertex location. This circuit uses the time difference in the signals to determine the location of the primary vertex in z to within ±6 cm in less than 200 ns.

The EMC, covering a region with pseudorapidity less than one, measures a fraction of the total transverse energy which comes in the form of photons and electrons. This energy is primarily the result of π^0 decays. The EMC is used to select events on the basis of the centrality of the collision and fluctuations in E_t. The EMC, together with the CTB, can be used to select events with unusual ratios of E_t-to-charge multiplicity.

One purpose of the TPC MWPC's is to trigger on electron avalanche at the anode wires. The direct readout of this information allows triggering on multiplicity fluctuations between different regions of psuedorapidity. The MWPC's cover 1-2 in $|\eta|$.

The VC's are placed beyond the first dipole magnets in the beam (about seventeen meters from the interaction point) and cover $|\eta| > 5$. These provide information on the impact parameter independent of the E_t values. By looking at the well-separated proton and neutron spectators from the interaction point, the spectator energy can be detected. The particle density in psuedorapidity is shown in figure 2.15.

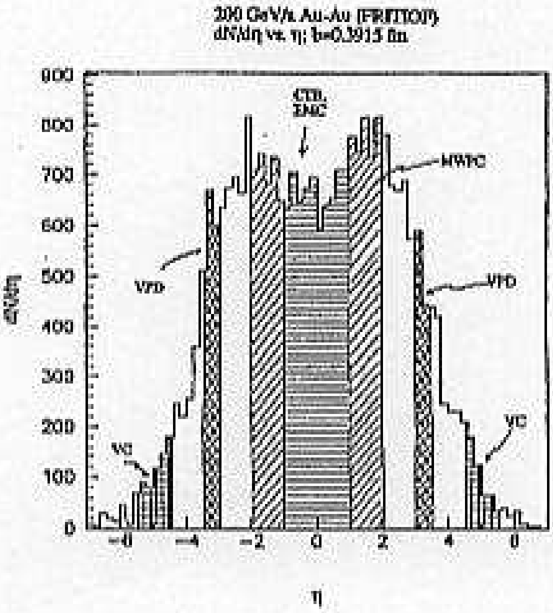


Figure 2.15
Typical particle density in rapidity. Shaded regions indicate the coverage of the trigger detectors [ST 92].

Trigger decisions are made according to a three step process. At each step (level), the event (interaction) may be rejected or retained for additional processing.

Level 1

The first-level trigger provides a veto decision within 200 ns. A determination of the total transverse electromagnetic energy is done with the information from the EMC.

Location of the collision within ±6 cm to keep the collision inside the optimum SVT acceptance is done using the VPD. The CTB is used to trigger on multiplicities. The TPC gating grid may be opened within approximately 1 μs of the original collision on the basis of the level one data.

Level 2

The second-level trigger will provide a decision within 50 μs. This time is set by the maximum drift time in the TPC, during which time data is passed onto a switched-capacitor array where it can be stored for a few hundred μs without degrading the signal. Simultaneously, the data is analyzed to decide whether to accept or reject an event. If the event is rejected, it takes 10 μs for the TPC FEE to reset itself. If the event is accepted, the data is passed on to the next level of analysis.

Level 3

The third-level trigger provides a decision on the order of a ms. This time scale is based on extensive processing of track and calorimeter data to look for high-p_t tracks, energy clusters correlated with jets, and for large-scale fluctuations in multiplicity and E_t [ST 92].

Chapter 3. Slow Controls System

3.1 Slow Controls System Overview

The main purpose of the STAR Controls System is to ensure the validity and consistency of the recorded data so that physics may be extracted. To this end the STAR Controls System:

- Sets the system parameters according to a pre-defined sequence
- Saves and restores the detector configuration
- Provides an alternate data acquisition path
- Verifies proper functioning of the detector by monitoring the subsystem parameters
- Warns about possible subsystem malfunctions
- Handles alarms
- Controls normal operations
- Provides fault diagnostics
- Allows for simultaneous testing of different subsystems
- Archives information about the subsystem parameters in an easily accessible way
- Provides logging capability

In addition to these system features, the development of the Controls System has been guided by other system needs. A large and complex detector system like STAR will see a large turnover of users and control system operators. Because of this, the system needs to be well-documented and, perhaps more importantly, robust and suitably supported. The system will employ the use of graphical user interfaces, on-line help and suggestions, and a color-coded "traffic light" model of representing alarms. As well as having to manage tens of thousands of channels, the system must allow for the addition of new subsystems and upgrades.

The systems included in the baseline STAR Controls System are the time projection chamber, magnet, data acquisition, trigger and beam.

3.2 EPICS

The Experimental Physics and Industrial Control System (EPICS) was selected as the foundation for the STAR Controls software environment because it incorporates all of the system features outlined above. EPICS was designed by Los Alamos National Laboratory and the Advanced Photon Source at Argonne National Laboratory. At STAR, EPICS is run on a UNIX-based workstation connected to Versa Module Europa (VME) crates operating under VxWorks, a real-time operating system [JG 94].

EPICS was designed as a development toolkit and common run-time environment that allows users to build and execute real-time control and data acquisition systems for experimental facilities [RC 92].

3.2.1 EPICS Database Configuration

A principal component of EPICS is its database. The database is an ASCII file that is configured off-line using a Database Configuration Tool and is loaded into the memory of an Input Output Controller (IOC) at boot time. The IOC consists of a VME crate, a CPU, and any application-specific hardware interfaces. The database consists of specific records written for alarm checking and setting, and for monitoring and controlling parameters within each subsystem. The database configuration tool in current use is the Graphical Database Configuration Tool (GDCT). GDCT is part of the EPICS software release and provides application developers with a Computer-Aided Drafting (CAD)-type drawing of the database as it is made (see figure 3.1a). The great advantage of GDCT is that it allows one to graphically see the links between database records, and process and logic flow within a database. By "clicking" on a record in a database, the developer can see and manipulate the attributes (fields) of the record. For these reasons, GDCT is favored over the Database Configuration Tool (DCT) which is a menu driven tool that shows only one record at a time in a given database (see figure 3.1b).

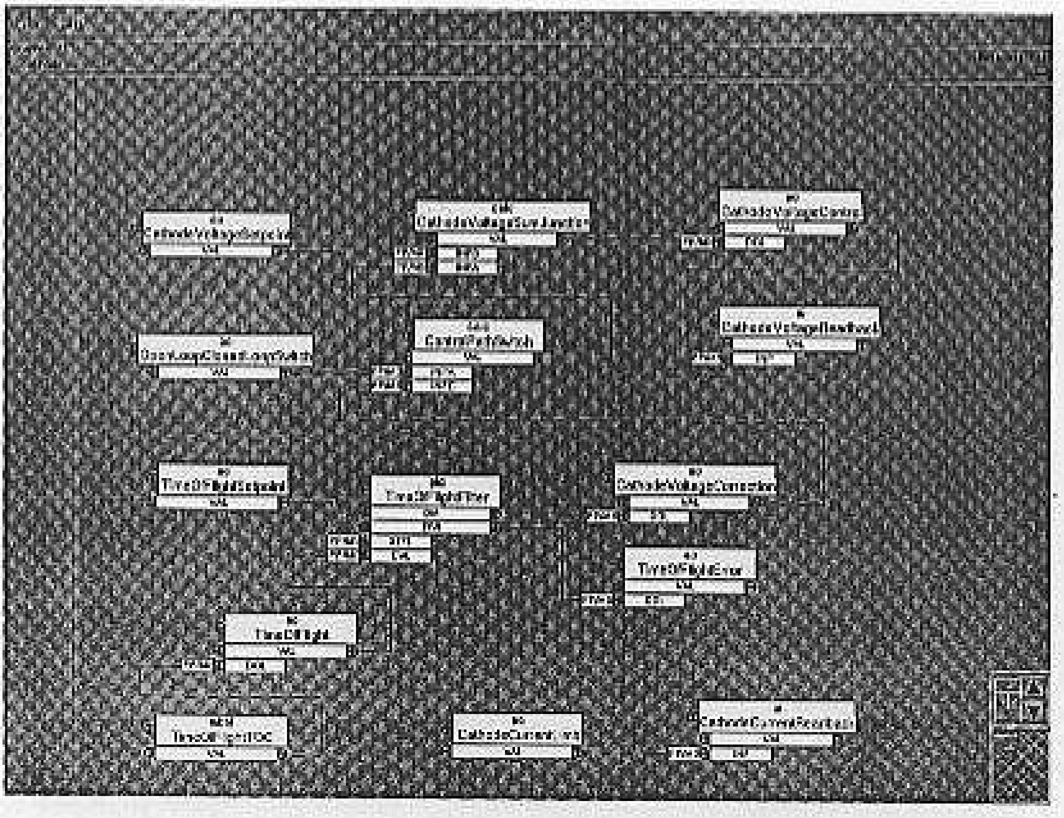
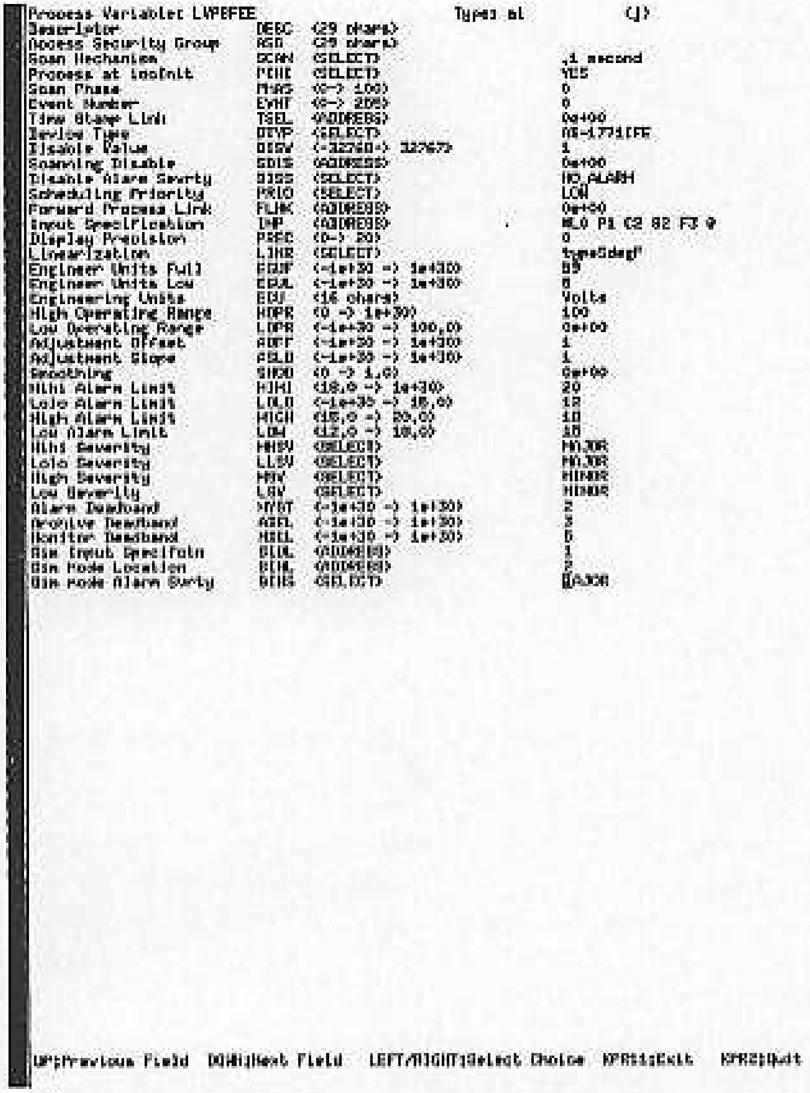


Figure 3.1a
A GDCT screen showing the database that controls the TPC Drift Velocity High Voltage Control [JG 95a].



```
Process Variable: LVPEFEE                         Type: ai          (J)
Descriptor              DESC   <29 chars>
Access Security Group   ASG    <29 chars>
Scan Mechanism          SCAN   <SELECT>                          .1 second
Process at iocInit      PINI   <SELECT>                          YES
Scan Phase              PHAS   <0-> 100>                         0
Event Number            EVNT   <0-> 255>                         0
Time Stamp Link         TSEL   <ADDRESS>                         0e+00
Device Type             DTYP   <SELECT>                          AB-1771IFE
Disable Value           DISV   <-32768-> 32767>                  1
Scanning Disable        SDIS   <ADDRESS>                         0e+00
Disable Alarm Severty   DISS   <SELECT>                          NO_ALARM
Scheduling Priority     PRIO   <SELECT>                          LOW
Forward Process Link    FLNK   <ADDRESS>                         0e+00
Input Specification     INP    <ADDRESS>               .         #L0 P1 C2 S2 F3 @
Display Precision       PREC   <0-> 20>                          0
Linearization           LINR   <SELECT>                          typeSdegF
Engineer Units Full     EGUF   <-1e+30 -> 1e+30>                 89
Engineer Units Low      EGUL   <-1e+30 -> 1e+30>                 0
Engineering Units       EGU    <16 chars>                        Volts
High Operating Range    HOPR   <0 -> 1e+30>                      100
Low Operating Range     LOPR   <-1e+30 -> 100.0>                 0e+00
Adjustment Offset       AOFF   <-1e+30 -> 1e+30>                 1
Adjustment Slope        ASLO   <-1e+30 -> 1e+30>                 1
Smoothing               SMOO   <0 -> 1.0>                        0e+00
Hihi Alarm Limit        HIHI   <18.0 -> 1e+30>                   20
Lolo Alarm Limit        LOLO   <-1e+30 -> 15.0>                  12
High Alarm Limit        HIGH   <15.0 -> 20.0>                    18
Low Alarm Limit         LOW    <12.0 -> 18.0>                    15
Hihi Severity           HHSV   <SELECT>                          MAJOR
Lolo Severity           LLSV   <SELECT>                          MAJOR
High Severity           HSV    <SELECT>                          MINOR
Low Severity            LSV    <SELECT>                          MINOR
Alarm Deadband          HYST   <-1e+30 -> 1e+30>                 2
Archive Deadband        ADEL   <-1e+30 -> 1e+30>                 3
Monitor Deadband        MDEL   <-1e+30 -> 1e+30>                 5
Sim Input Specifctn     SIOL   <ADDRESS>                         1
Sim Mode Location       SIML   <ADDRESS>                         2
Sim Mode Alarm Svrty    SIMS   <SELECT>                          MAJOR




UP:Previous Field  DOWN:Next Field  LEFT/RIGHT:Select Choice  KPF1:Exit   KPF2:Quit
```

Figure 3.1b
A DCT screen used to create a database. Only one record can be viewed at a time [JG 95a].

Another tool for configuring databases is CAPFAST. CAPFAST is a CAD program which has been reprogrammed by software engineers at Jefferson National Laboratory to produce EPICS-database ASCII files. The most important feature of CAPFAST is that it can generate several databases from one drawing. Within a controls system that uses EPICS, each instance of a record must have a unique name. For example, it might be that one needs to develop a database to control a cluster of power supplies. This is an easy task for GDCT. Now say that the experiment has 300 instances of identical power supply clusters. Using GDCT a developer would have to copy the original database 300 times and change all the record names in the databases so that each one was unique. This is a very time consuming task. Using CAPFAST, however, the task requires little more that making the original database, telling CAPFAST to follow a naming scheme and how many instances of the database are needed.

3.2.2 Operator Interfaces

The current STAR operator interfaces (OPI) are SUN IPX workstations. A display editor, Motif Editor and Display Manager (MEDM), is used to create and modify the OPI displays. MEDM allows a developer to design monitoring and control interfaces to suit a specific application. A developer using MEDM has the capability to display graphs and histograms as well as set up monitoring and control dials, switches, alarm conditions, and color intensity. Figure 3.2 shows a MEDM screen developed for the TPC drift velocity high voltage control. A more extensive MEDM screen which controls the LINAC at the Advanced Photon Source at Argonne National Laboratory is shown in figure 3.3.

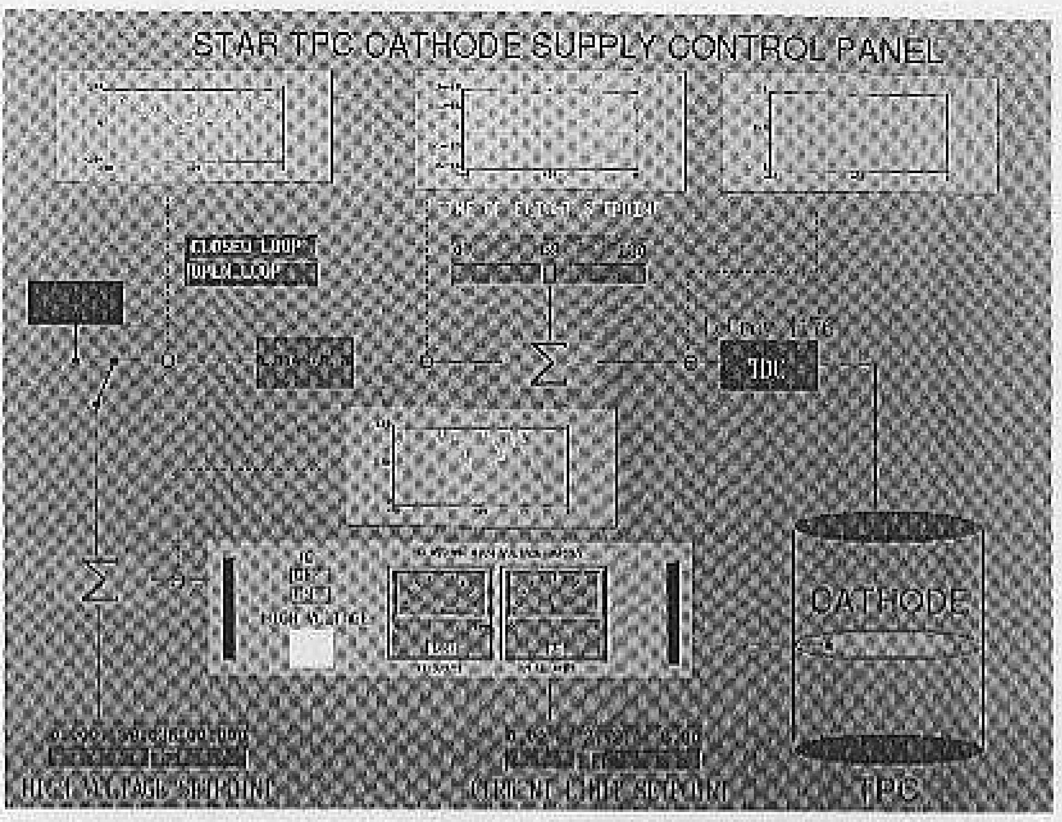


Figure 3.2
An MEDM screen used to control the TPC drift velocity [JG 95a].

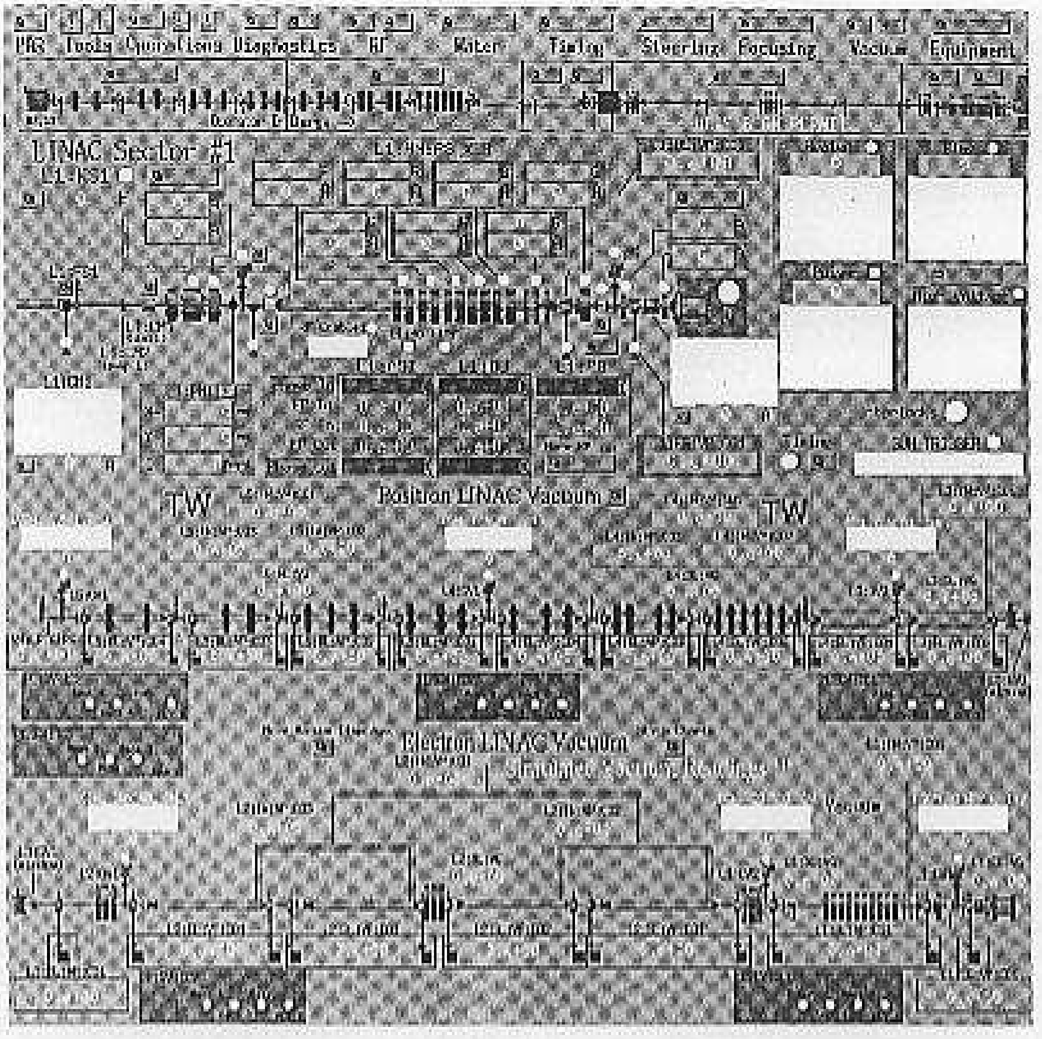


Figure 3.3
A top level MEDM screen used to control the LINAC at Argonne National Laboratory [JG 95a].

3.2.3 EPICS Features

The STAR controls system is a real-time fully distributed controls system. The system hardware consists of a VME crate for each subsystem using commercially available industrial interfaces and programmable controllers wherever possible. The monitoring tasks and system control are distributed on computer workstations. The workstations and VME crates are networked with ethernet. If a single-input output controller becomes saturated, the processing can be shared with other IOC's. If an operator interface were to become saturated, the processing could be distributed to other OPI's [JG 94]. The organization of EPICS is shown in figure 3.4.

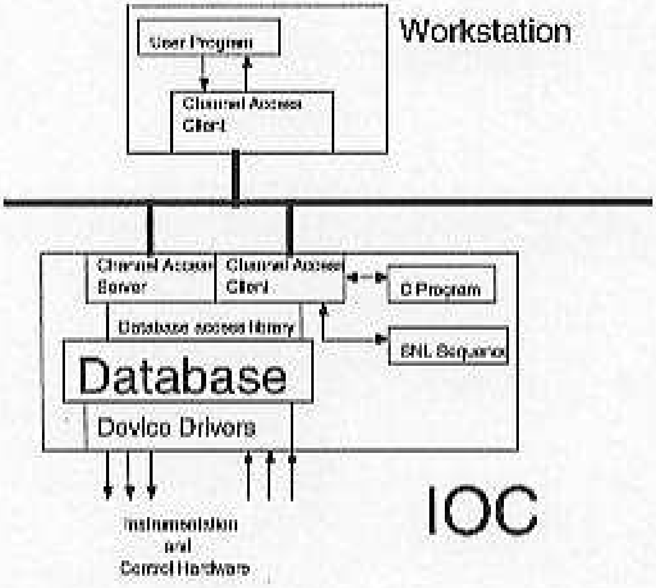


Fig. 3.4
Configuration of EPICS.

One advantage of using EPICS is that it comes with procedures that have been written to support analog and binary inputs and outputs, calculations, fanouts, multi-bit binary inputs and outputs, timers, and a score of other functions. EPICS uses a protocol called Channel Access to communicate with distributed IOC's and workstations. Channel Access hides the details of TCP/IP networking to both clients and servers in a given implementation. The modularity of EPICS eases the development of software to control specialized hardware.

Each subsystem in STAR has an EPICS-based hardware Controls System which is used by Experiment Control as a tool to control and monitor the performance of the detector hardware. Experiment Control does not execute detailed control of subsystem hardware directly; it issues commands to the appropriate subsystem control processes. This method of implementation serves two purposes. First, it preserves the "chain of command" so that it is clear who or what is controlling a piece of hardware. Second, it provides an interface between the experimentalists working on subsystem control and those involved in Experiment Control.

3.3 The STAR System Test

3.3.1 System Test Setup

For the STAR system test, one TPC outer sector has been instrumented with 16 FEE cards serviced by one readout board. The TPC sector has been fitted with a small drift chamber (drift distance ≈ 10 cm) through which flows a gas mixture of 90% argon - 10% methane. The configuration of the setup is shown in figure 3.5.

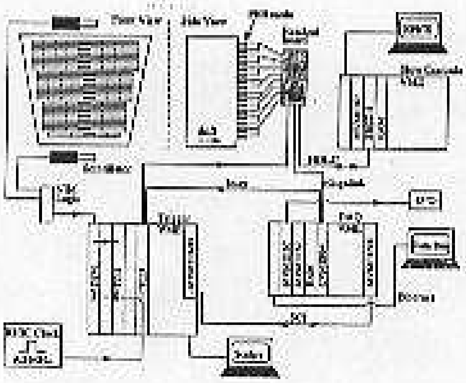


Figure 3.5
The STAR system test hardware setup [JC 96].

3.3.1.1 Data Acquisition System

A data acquisition system (miniDAQ) has been specifically designed for the system test. MiniDAQ consists of [JC 96]:

- two Motorola VME167 modules with 68040 processors that run the miniDAQ software;
- one Cyclone VME964 module, housing an I960 processor, that holds the "Rosie" mezzanine card, a temporary receiver for STAR;
- a Motorola MVME1600-001 (Power PC) that acts as a trigger-miniDAQ interface;
- and an additional 32 megabyte memory board for interproscessor communication and event buffering.

MiniDAQ receives data from the readout board over a 1.5 Gbit/s optical fiber link. The Rosie board takes this data and packs it into 20-bit quantities, representing two 10-bit ADC values, and then pads the data into a two-byte word.

When the Trigger system notifies miniDAQ of an upcoming event, miniDAQ starts the transfer of the data from the Rosie card to the memory board in the miniDAQ VME crate. During the transfer time a busy signal is sent to the Trigger system to prevent further incoming triggers.

3.3.1.2 Trigger

The system-test trigger is a VME-based prototype of the trigger that will be used in STAR. The Data Storage and Manipulation (DSM) board receives signals from scintillators connected to photomultiplier tubes, positioned above and below the TPC sector. If these signals are in coincidence with a master clock, which replicates the RHIC strobe, a 16-bit word is passed to the Trigger Control Unit (TCU). The TCU examines the 16 bits and translates them to a trigger command word that will indicate the type of event that has been identified (i.e. high multiplicity event, low multiplicity event etc.). In the current setup the only type of trigger type is a cosmic ray. The TCU then assigns a 12-bit token to the trigger command word. This token prevents the mixing of events from different sub-detectors in the event building stage. The command word and token are sent from the Trigger and Clock Distribution (TCD) board to the TPC readout board and the event data is sent to the Rosie board in the Data Acquisition system. The information from the TCU and DSM board is used in the trigger Level 1 accept/reject decision, and notifies mini-DAQ of its decision. Once the mini-DAQ system has been notified by the trigger of a Level 1 accept, it takes the event data from the Rosie board and puts it into the VME-event memory buffer. The event data is now ready to be written to tape or used for on-line analysis.

3.3.1.3 Slow Controls

In the actual implementation of the STAR detector the TPC readout boards require a link to the Slow Controls system for a number of testing, control and monitoring tasks as well as to provide STAR with an alternate data-acquisition path. Since each readout board stores over one megabyte of data per event, the Slow Controls link speed is required to be in excess of 10 Kbytes/sec to be of use as a alternate data-acquisition path. The link must operate between the electronics platform and the TPC endcaps, a distance of 30 meters, as well as in the 5-kG magnetic field present in STAR. Due to space considerations, the link must also accommodate a multi-drop topology to reduce cabling [PB 95],[SK 95]. To accomplish this, a bi-directional data transmission and control system is required that interfaces easily with the EPICS software environment.

High-Level Data Link Control (HDLC) was selected as the protocol for the TPC/Slow Controls link because it incorporates all the requirements outlined above. The link throughput is about 1 Mbit/sec in the STAR implementation. The major disadvantage of the generic HDLC is that it requires some software development and more on-board hardware than some of the more highly integrated choices[PB 95],[SK 95]. The HDLC interface accepts function-specific parameters passed from an EPICS database to execute a desired data transmission.

The TPC Front End Electronics (FEE) boards receive event signals from the chamber, and then perform amplification and digitization of the signals. The TPC readout boards receive input from the FEE boards and store the data in event memory or send the information to the Data Acquisition (DAQ) subsystem through a fiber optics link. By pulsing the readout board with known signals, the Slow Controls link can be used to verify that the signals were properly amplified, digitized and stored in memory. The Slow Controls link is also used to write random data to the readout board memory. This data can be read from the readout board by DAQ and also by Slow Controls. Comparisons can be made between the original data and the data sets that are read by DAQ and Slow Controls. This method is used to test data transmission on the fiber optics link as well as on the Slow Controls link.

The TPC has two endcaps each of which is divided up into 12 supersectors. Each supersector network is made up of 6 readout-board slave nodes and one commercially built master node in a VME crate. The slave nodes are based on a 68302 microcontroller running at 20 MHz, and are built on a 73-mm by 84-mm printed circuit "daughter" board which is designed to plug into a readout board. The board communicates with the readout board via memory mapping. The commercially built master node is a Radstone SBCC-1 which is based on a 68360 processor, and supports four HDLC channels. A TPC sector is fitted with readout boards and slave nodes which are then connected to one HDLC channel on a Radstone master node. Timing signals are sent from the Radstone board and reflected back by the readout boards. This is done because the 68302 serial ports cannot self-clock. The hardware configuration for a typical TPC sector is shown in figure 3.6.

The HDLC-EPICS interface consists of subroutines which are loaded into the IOC at boot time and perform read and write transactions to and from a TPC readout board memory in a transparent manner via the Radstone board.

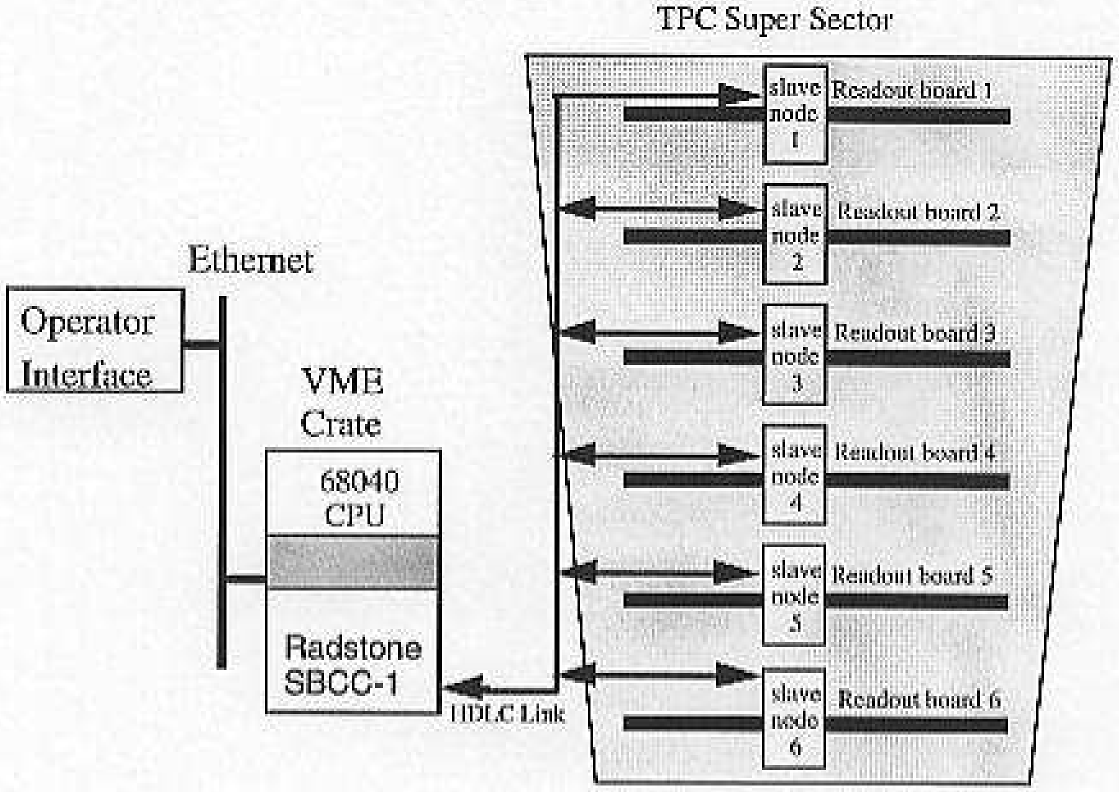


Figure 3.6
Hardware configuration for a typical TPC sector. There are six readout boards per super sector, each of which is slave-node daisy-chained by an HDLC cable to a Radstone board. (Each Radstone board can service up to 4 HDLC links; only one is shown.)

The Slow Controls function in the system test is to record the "housekeeping" parameters of the TPC front-end electronics chain. These parameters include the values of the low-voltage power supplies that power the readout board and the FEE cards and the temperatures of the electronics. Slow Controls can also be used in the geometric configuration (mapping) of the FEE cards on the TPC, test the FEE and pad aliveness, as well as read event data from the readout board over the HDLC link.

3.3.1.4 State Manager

A prototype State Manager has been developed to regulate the operations of each of the system test subsystems. MiniDAQ, Trigger, and Slow Controls have been interfaced to the prototype State Manager. The State Manager incorporates three state models: a sequence state model, a command state model, and an alarm state model. The sequence state model brings each subsystem on-line by stepping through a series of states (i.e. initialize, configure, run). The command model and alarm model provide a command and error messaging interface between the subsystem and operator.

The State Manager does not "know" the detailed operation of each subsystem; it issues commands and requests to each subsystem. The individual subsystems are required to "know" what to do when a certain command or request is issued from the State Manager.

3.3.2 Analysis of TPC Data

The data set discussed here was obtained during the period between July and October 1996. Five different types of data were collected: geometry events, pedestal events, pulser events, cosmic-ray muon events and laser-induced events.

3.3.2.1 Geometry Events

Geometry events were triggered whenever a new arrangement of FEE cards was connected to the TPC sector. Associated with each of the FEE connectors on the TPC pad plane is a "hard-wired" address, which specifies the location of that connector on the TPC. When A FEE card is plugged into a connector on the pad plane it assumes the geographical address of that connector. The readout of these addresses is used to map the FEE card locations on a readout board. This geographical map of the card locations is used in data analysis as well as to identify cabling errors.

3.3.2.2 Pedestal Events

Pedestal events determine the average ADC count for a given channel when no tracks are present. Typically one hundred events were taken for each pedestal determination run. Each run was taken under normal data-taking conditions with the exception of the anode-wire voltage which was disengaged to avoid contamination from cosmic rays. Because the SCA response is slightly time dependent, the average of the one hundred ADC values were averaged for each time division (bucket) of each channel. Figure 3.7 shows the pedestals of four channels as a function of time bucket.

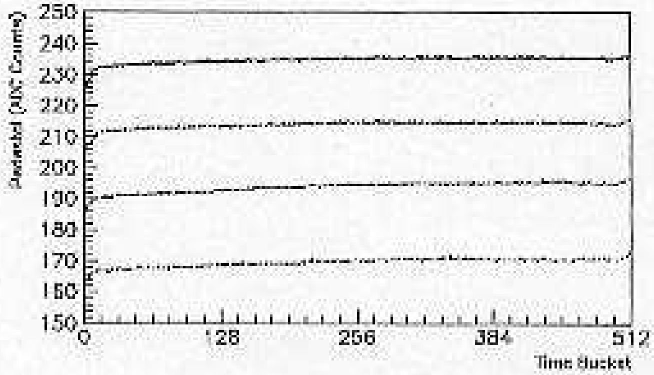


Figure 3.7
Pedestals as a function of time bucket [JC 96].

The RMS of a given channel's ADC values gives a measure of the noise level -- the variation in pedestal ADC value. Figure 3.8 shows the noise level for an average channel. In order to monitor the stability of the pedestals with time, continual pedestal event runs were taken over the period of several weeks, and the mean value of a given channel averaged over all time buckets was plotted as a function of time (figure 3.9).

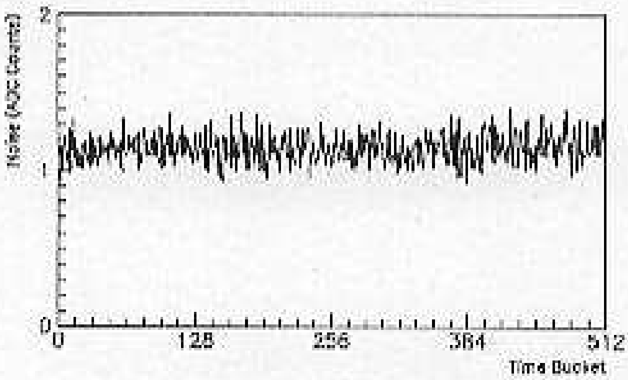


Figure 3.8
Noise as a function of time bucket [JC 96].

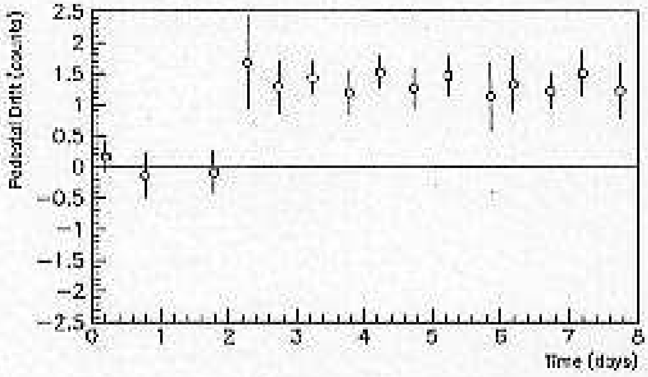


Figure 3.9
Pedestal stability [JC 96].

3.3.2.3 Pulser Events

Pulser events are used to measure the electronic gain of each channel. A current pulse is applied to the TPC ground plane which induces signals on the pad plane. Analysis of the pulsed ADC counts provides information on the electronic gain of each channel. Information on the stability of the gain over the period of several days is shown in figure 3.10. The FEE cards also have the ability to pulse themselves via an on-chip calibration system. The self-pulsing of the FEE cards is used to measure the absolute electronic gain.

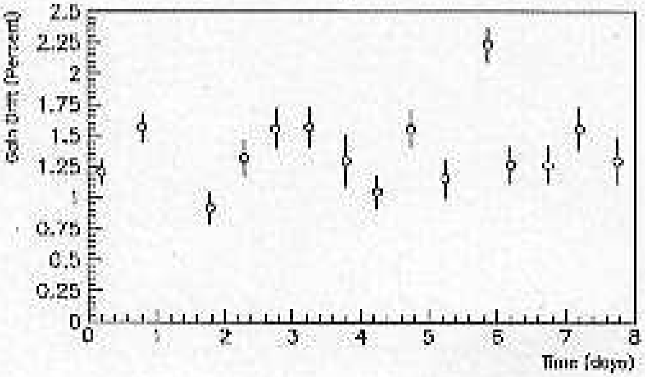


Figure 3.10
Gain stability [JC 96].

3.3.2.4 Cosmic Ray Events

Cosmic ray events (figure 3.11) were triggered by the use of two scintillators, one above the TPC sector and one below. Events of this type allowed the track reconstruction software to be used on real TPC data which is used to determine the hit resolution. The hit resolution is a determination of the accuracy with which the position of a charged track though the TPC can be measured. The position resolution directly influences the momentum resolution, and therefore is of great importance in a high-multiplicity environment such as STAR. The track reconstruction software takes into account the pedestal ADC values and gain corrections for each channel before it determines the position resolution. Initial measurements from cosmic ray tracks give a hit resolution of 410 μm [JC 96].

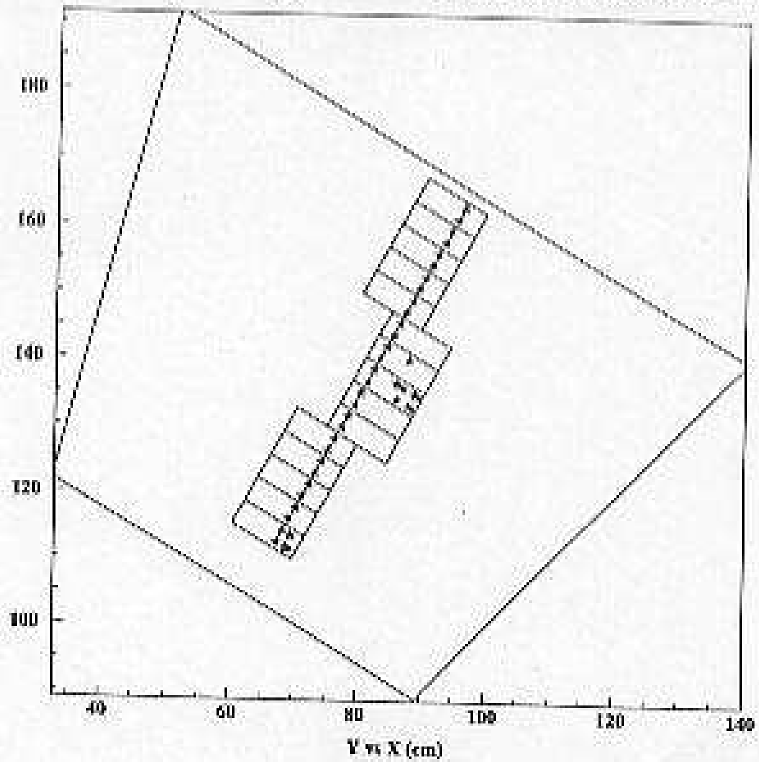


Figure 3.11
Hits from a cosmic ray track.

3.3.2.5 Laser-Induced Events

Laser-induced events were used to determine the two-track resolution of the TPC. The two-track resolution is the measurement of the distance at which two hits from separate tracks appear as one hit in the TPC pad plane. To measure this distance, a laser beam was directed from the bottom of the TPC test sector to the top where it was reflected back by a mirror at a small angle relative to the incident beam. This method produces two ionization tracks in the TPC drift volume in close spatial proximity (see figure 3.12). Several events

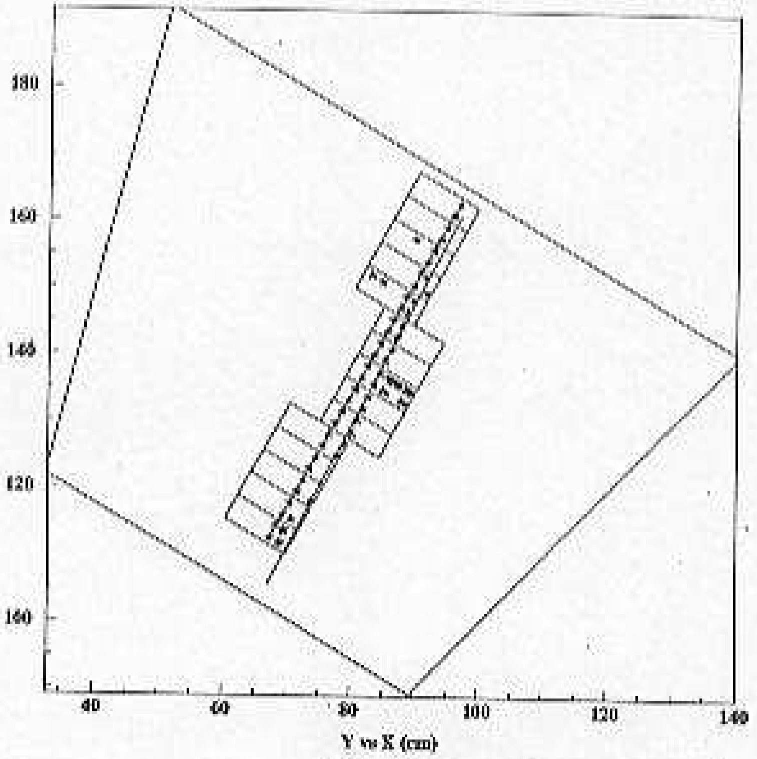


Figure 3.12
Laser induced two track event in TPC test sector [JC 96].

were taken at different angles, and the initial results show that two hits from different tracks blend into one hit when the track separation is less than 1 cm in the pad plane (x-y) plane as shown in figure 3.13.

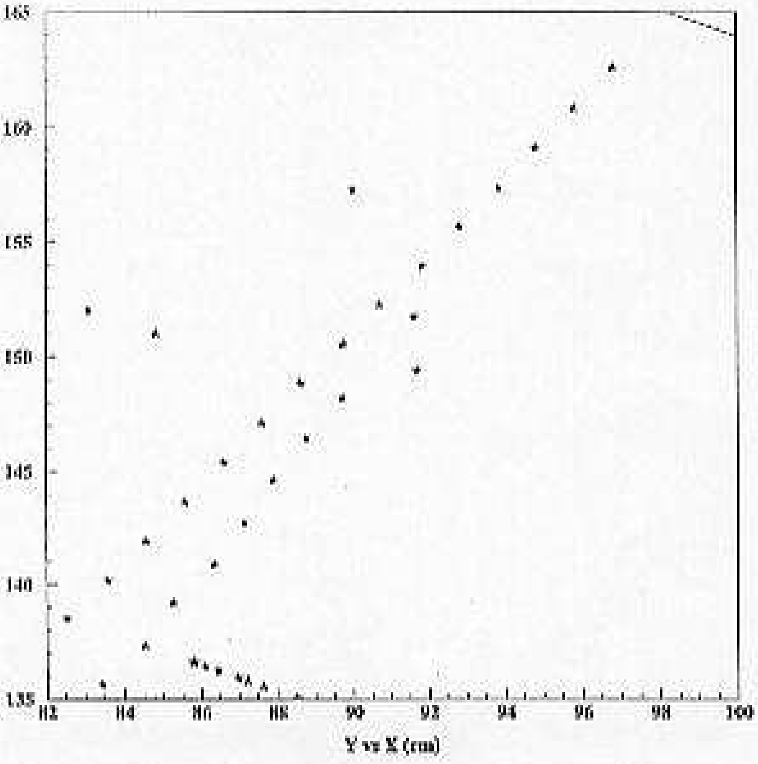


Figure 3.13
A zoom-in on figure 3.12 showing two track resolution of about 1 cm.

3.4 Author's Contribution

The development of the EPICS-to-HDLC interface to be used by STAR, as well as the testing and support of new EPICS implementations were accomplished. Along with this, several introductory EPICS training sessions for individuals within the STAR collaboration were setup and carried out by the author. At the time of this writing the author has been given charge of the development, construction and testing of two new scintillator triggers to be used in the system test environment.

3.4.1 EPICS - HDLC Interface

The HDLC link was designed to read data words from and write data words to memory locations on the TPC readout board. The interface required the passing of command data (memory address, network and node specifications, specific command words) from an EPICS database to the HDLC software.

The development of the EPICS-HDLC interface occurred in three steps. The decision to use the HDLC protocol was not yet made when work on the interface began, yet it was known that the Slow Controls system would have to read from and write to memory locations. It was decided that the first step in constructing the interface was to use EPICS to read and write from and to any VME memory location. To accomplish this, a detailed look into the configuration of the EPICS software was performed. As was mentioned before, EPICS uses records to perform general operations (i.e. analog inputs, binary outputs etc.). However, the record support routines do not contain information on how to communicate with specific hardware, but rather, rely on a device support software layer to take care of all hardware specific transactions. Hence, device support was written to perform the VME memory transactions. The existing Long-in and Long-out EPICS records use this device support to write or read one long data word to or from VME memory. The next step was to use the new device support to read and write large blocks (100's of Kbytes) of data to and from data files on a user workstation. Upon doing this it was found that the time duration for the transfer of 100's of kilobytes of data was much too large. The problem was that EPICS had to make individual network connections from the workstation to the VME crate for each data word transaction. What was needed was an EPICS record specifically written to move a large data array within a single network transaction. At the time no such record existed, but it was discovered that a generic VME record (allowing access to VME memory via data array transfers) was being developed at the University of Chicago Synchrotron Community. The development of this record guided the development of the EPICS - HDLC interface into the third and final stage. By the time the generic VME record was made available to the EPICS community, the HDLC software was near completion. The generic VME record was modified by the author (appendix B) to pass the appropriate parameters to the HDLC software.

The EPICS - HDLC link was the first data path to and from the STAR TPC and was used to facilitate the initial debugging of the TPC electronics. In the summer of 1997, mini-DAQ will be moved out of the system test setup. The EPICS -HDLC will be the only data path to the TPC test sector and will be used in the testing of the on-line analysis software.

It should be mentioned that STAR as well as BRAHMS, another collaborative detector experiment on the RHIC ring, plan on using the technology discussed here to monitor and control a large portion of their experimental hardware throughout the lifetimes of the experiments.

3.4.2 Trigger Scintillators

Two new scintillator triggers have been constructed by the author to facilitate the testing of the pad plane response to cosmic ray tracks which cross the TPC sector at an angle with respect to the vertical. For example, say that an amount of charge n is deposited on a pad with the passage of a cosmic ray straight down through the TPC test sector. If a cosmic ray traversed the test sector at an angle of 45° with respect to the vertical, one would expect an amount of $n\sqrt{2}$ charge deposited on the pad. A determination of the pad response to this situation must be made if the amount of charge deposited on the pads is to be used to track the energy loss (dE/dx) of a particle passing through the TPC, as it will be in STAR.

The scintillators were designed to cover the entire drift region of the TPC test sector (see figure 3.14a and 3.14b) and will be used in conjunction with present system test setup. A block of twenty FEE cards will instrument the test sector and read out the cosmic ray data.

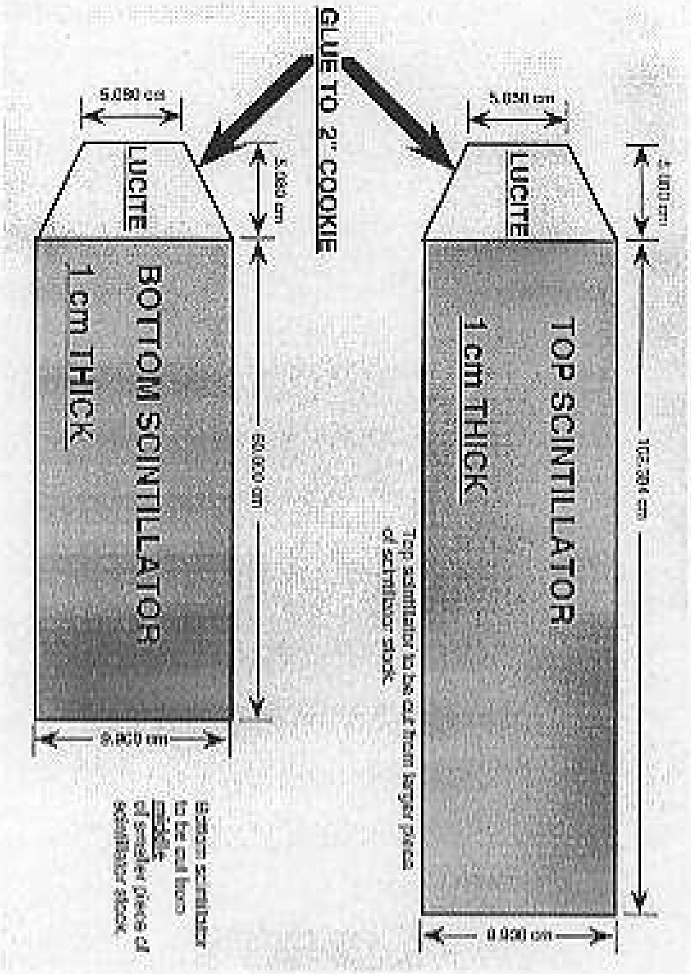

GLUE TO 2" COOKIE
5.080 cm
LUCITE
5.080 cm
BOTTOM SCINTILLATOR
1cm THICK
60.000 cm
9.900 cm
Bottom scintillator to be cut from middle of smaller piece of scintillator stock
5.080 cm
LUCITE
5.080 cm
TOP SCINTILLATOR
1cm THICK
Top scintillator to be cut from larger piece of scintillator stock
9.900 cm

Figure 3.14a
Dimensions of trigger scintillators.

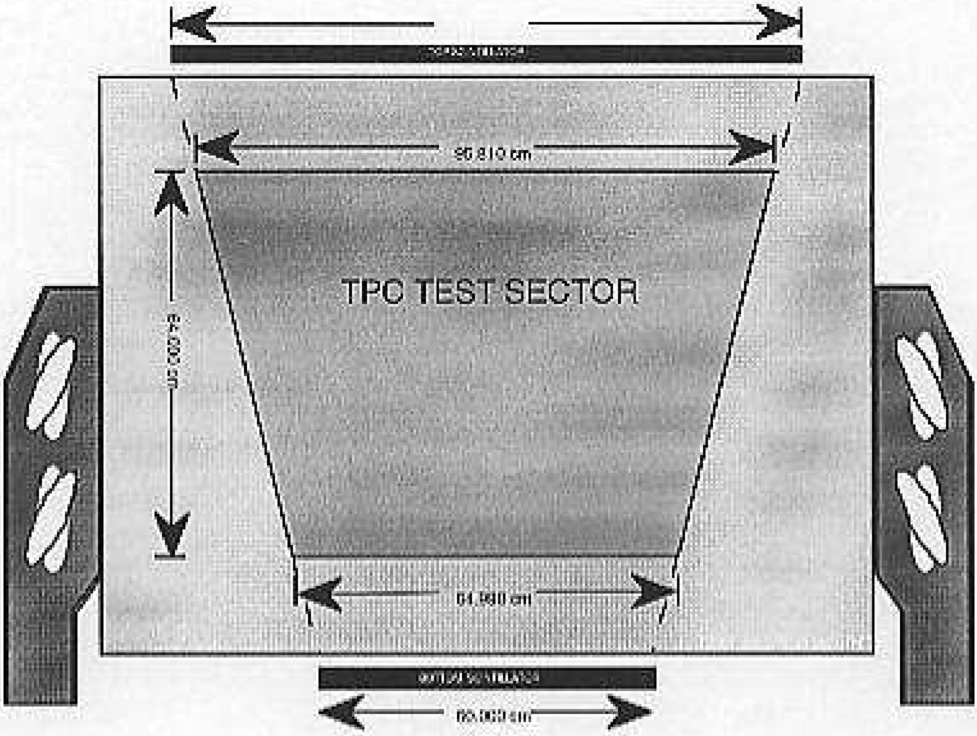


Figure 3.14b
Schematic of trigger scintillator placement.

Conclusions

EPICS has shown itself to be useful as a slow controls system in a high-energy physics detector application. The EPICS - HDLC interface has proven to be a reliable, inexpensive alternate data acquisition path. EPICS will be used in the low-level control and monitoring of the STAR subsystems, while Experiment Controls will take charge in the high-level operation of the experiment.

Appendix A

Rapidity

Rapidity, *y*, is used to describe the kinematic condition of a particle. It is a dimensionless quantity which is related to the ratio of a particles forward momentum to its backward momentum. Rapidity defined in terms of a particle's energy-momentum components, p_0 and p_z, by,

$$y = \frac{1}{2}\ln\left(\frac{p_0 + p_z}{p_0 - p_z}\right) \qquad (1)$$

The rapidity of a particle in one reference frame is related to the rapidity of the same particle in another reference frame by an additive constant (following [WO 94]):

> Consider a particle in reference frame *F* with rapidity *y* and rapidity *y′* in a Lorentz frame *F′* which moves with a velocity β in the z-direction. The rapidity *y′* of the particle in the *F′* frame is defined as:
>
> $$y' = \frac{1}{2}\ln\left(\frac{p'_0 + p'_z}{p'_0 - p'_z}\right) \qquad (2)$$
>
> Under the Lorentz transformation, the energy p'_0 and the longitudinal momentum p'_z in the frame *F′* are related to the energy p_0 and the longitudinal momentum p_z in the frame *F* by
>
> $$p'_0 = \gamma\,(p_0 - \beta p_z) \qquad (3)$$
>
> $$p'_z = \gamma\,(p_z - \beta p_0) \qquad (4)$$

where β is the velocity of F' relative to F. If we substitute equations (3) and (4) into equation (2) we get

$$y' = \frac{1}{2}\ln\left(\frac{\gamma(p_0 - \beta p_z) + \gamma(p_z - \beta p_0)}{\gamma(p_0 - \beta p_z) - \gamma(p_z - \beta p_0)}\right) \qquad (5)$$

or

$$y' = y + \frac{1}{2}\ln\left(\frac{\gamma(1-\beta)}{\gamma(1+\beta)}\right)$$

where y is the rapidity of the particle in the F frame.

Pseudorapidity

To characterize the rapidity of a particle, it is essential to measure two quantities of the particle, the longitudinal momentum and the energy. In many experiments it is possible to measure only the angle of the particle's trajectory with respect to the beam axis. In this case, it is convenient to use this information by employing the pseudorapidity variable η. The pseudorapidity is defined as:

$$\eta = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right] \qquad (6)$$

where θ is the angle between the particle 4-momentum **p** and the beam axis. The pseudorapidity can be written in terms of the momentum as:

$$\eta = \frac{1}{2}\ln\left(\frac{|\mathbf{p}| + p_z}{|\mathbf{p}| - p_z}\right) \qquad (7)$$

A close examination of equations (1) and (7) show that the pseudorapidity coincides with the rapidity when the momentum is large, i.e. $|\mathbf{p}| \approx p_0$, the particle energy. The pseudorapidity is used to define locations of an experiment, i.e. "The STAR TPC covers -2 to 2 in pseudorapidity."

Appendix B

VME-HDLC Code

The first five code listings in this appendix belong to the EPIC - HDLC interface.

A brief description of each file is as follows:

RecVmeHDLC.c This is the main body of code that makes up the VMEHDLC record. This record serves as the interface between EPICS and HDLC.

vmehdlcRecord.h This is the header file which supports the main code. It contains calls to include other header file that are needed to compile the main code as well as declarations of structures used in the main code.

vmehdlcRecord.ascii This file contains ASCII definitions of many of the main record fields.

choiceVMEHDLC.h This is a header file that contains the record structure choice values.

choiceRec.ascii This file contains the ASCII strings for each of the record choices.

The last file, named ***modelr_rb.c***, was written by Dr. Jan Chrin (Creighton University). This file and variations of it are responsible for reading entire events out of the event memory on the TPC readout boards and it can also be used to write fake events to the TPC readout boards. This code utilizes the VMEHDLC record for the transferring of the data.

RecVmeHDLC.c

```
/* recVmeHDLC.c - Record Support Routines for  VME HDLC Link*/
/*
 *     Author:        John Meier
 *     Date:          8/21/95
 *
 *
 *     Based on recVme.c by Mark Rivers
 *
 *
 *
 * Modification Log:
 * -----------------
 *                                  */


#include       <vxWorks.h>
#include       <types.h>
#include       <stdioLib.h>
#include       <lstLib.h>
#include       <string.h>

#include       <alarm.h>
#include       <dbDefs.h>
#include       <dbAccess.h>
#include       <dbFldTypes.h>
#include       <devSup.h>
#include       <errMdef.h>
#include       <recSup.h>
#include       <vme.h>
#include       <vmehdlcRecord.h>
#include  "readoutBdLib.h"

/* Create RSET - Record Support Entry Table*/
#define report NULL
#define initialize NULL
static long init_record();
static long process();
#define special NULL
static long get_value();
static long cvt_dbaddr();
static long get_array_info();
static long put_array_info();
#define get_units NULL
#define get_precision NULL
#define get_enum_str NULL
#define get_enum_strs NULL
#define put_enum_str NULL
#define get_graphic_double NULL
#define get_control_double NULL
#define get_alarm_double NULL

 #define READ_RO                    2
#define WRITE_RO           3
#define WRITE_RO_NR        4
#define NODES              5
#define RO_RESET           6


struct rset vmehdlcRSET={
  RSETNUMBER,
  report,
  initialize,
  init_record,
  process,
  special,
  get_value,
  cvt_dbaddr,
  get_array_info,
  put_array_info,
  get_units,
  get_precision,
  get_enum_str,
  get_enum_strs,
  put_enum_str,
  get_graphic_double,
  get_control_double,
  get_alarm_double };

static void monitor();
static long doVmeIo();


static long init_record(pvmehdle, pass)
  struct vmehdleRecord  *pvmehdle;
  int pass;
{
  if (pass==0) {
    if (pvmehdle->nmax <= 0) pvmehdle->nmax=1;
    pvmehdle->bptr = (char *)calloc(pvmehdle->nmax, sizeof(long));
    pvmehdle->sptr = (char *)calloc(pvmehdle->nmax, sizeof(char));
    return(0);
  }
  return(0);
}

static long process(pvmehdle)
  struct vmehdleRecord *pvmehdle;
{
  long status;

  status=doVmeIo(pvmehdle); /* Do VME I/O operation */

  recGblGetTimeStamp(pvmehdle);

  /* check event list */
  monitor(pvmehdle);
  /* process the forward scan link record */
  recGblFwdLink(pvmehdle);

  pvmehdle->pact=FALSE;
  return(status);
}

static long get_value(pvmehdle,pvdes)
   struct vmehdleRecord *pvmehdle;

   struct valueDes *pvdes;
{
  pvdes->no_elements=pvmehdle->nuse;
  pvdes->pvalue = pvmehdle->bptr;
  pvdes->field_type = DBF_LONG;
  return(0);
}

static long cvt_dbaddr(paddr)
struct dbAddr *paddr;
{
  struct vmehdlcRecord *pvmehdlc=(struct vmehdlcRecord *)paddr->precord;

  if (paddr->pfield == &(pvmehdlc->val)) {
    paddr->pfield = (void *)(pvmehdlc->bptr);
    paddr->no_elements = pvmehdlc->nmax;
    paddr->field_type = DBF_LONG;
    paddr->field_size = sizeof(long);
    paddr->dbr_field_type = DBF_LONG;
  } else if (paddr->pfield == &(pvmehdlc->sarr)) {
    paddr->pfield = (unsigned char *)(pvmehdlc->sptr);
    paddr->no_elements = pvmehdlc->nmax;
    paddr->field_type = DBF_CHAR;
    paddr->field_size = sizeof(char);
    paddr->dbr_field_type = DBF_CHAR;
  }
  return(0);
}

static long get_array_info(paddr,no_elements,offset)
struct dbAddr *paddr;
long *no_elements;
long *offset;
{
  struct vmehdlcRecord *pvmehdlc=(struct vmehdlcRecord *)paddr->precord;

  *no_elements =  pvmehdlc->nuse;
  *offset = 0;
  return(0);
}

static long put_array_info(paddr,nNew)
struct dbAddr *paddr;
long nNew;
{
struct vmehdlcRecord *pvmehdlc=(struct vmehdlcRecord *)paddr->precord;

  pvmehdlc->nuse = nNew;
  if (pvmehdlc->nuse > pvmehdlc->nmax) pvmehdlc->nuse = pvmehdlc->nmax;
  return(0);
}


static void monitor(pvmehdlc)
   struct vmehdlcRecord   *pvmehdlc;
{
  unsigned short monitor_mask;

  /* get previous stat and sevr  and new stat and sevr*/
  monitor_mask = recGblResetAlarms(pvmehdlc);
  /* check for value change */
  if (pvmehdlc->val != pvmehdlc->mlst) {
    /* post events for value change */
    monitor_mask |= DBE_VALUE;
    /* update last value monitored */
    pvmehdlc->mlst = pvmehdlc->val;
  }

  /* send out monitors connected to the value field */
  if (monitor_mask){
    db_post_events(pvmehdlc,&pvmehdlc->val,monitor_mask);
  }
  return;
}

static long doVmeIo(pvmehdlc)
  struct vmehdlcRecord   *pvmehdlc;
{
  int amod, length, mode, status=ERROR, value, i, nem, node;
  int *data_array = (int *) pvmehdlc->bptr;
  unsigned char *status_array = pvmehdlc->sptr;
  char *vmehdlc_ptr;
  short word_value;
  char char_value;
  int all_amods[] = {
    VME_AM_SUP_SHORT_IO,
    VME_AM_STD_SUP_DATA,
    VME_AM_EXT_USR_DATA,
    VME_AM_EXT_SUP_DATA};

  int all_lengths[] = {1, 2, 4};
  int all_modes[] = {VX_READ, VX_WRITE, READ_RO, WRITE_RO, WRITE_RO_NR, NODES, RO_RESET};

  amod = all_amods[pvmehdlc->amod];
  length = all_lengths[pvmehdlc->dsiz];
  mode = all_modes[pvmehdlc->rdwt];
  sysBusToLocalAdrs(amod, pvmehdlc->addr, &vmehdlc_ptr);
  if (mode == VX_READ) {
    for (i=0; i<pvmehdlc->nuse; i++) {
      switch(length) {
      case 1: status = vxMemProbe( vmehdlc_ptr, VX_READ, 1, (char *) &char_value);
            value = char_value;
            break;
      case 2: status = vxMemProbe( vmehdlc_ptr, VX_READ, 2, (char *) &word_value);
            value = word_value;
            break;
      case 4: status = vxMemProbe( vmehdlc_ptr, VX_READ, 4, (char *) &value);
            break;
       }
      data_array[i] = value;
      status_array[i] = status;
      vmehdlc_ptr += pvmehdlc->ainc;
    }
  }

  if (mode == VX_WRITE) {
    for (i=0; i<pvmehdlc->nuse; i++) {
      switch(length) {
      case 1: char_value = data_array[i];
            status = vxMemProbe( vmehdlc_ptr, VX_WRITE, 1, (char *) &char_value);
            break;
      case 2: word_value = data_array[i];
            status = vxMemProbe( vmehdlc_ptr, VX_WRITE, 2, (char *) &word_value);
            break;
      case 4: value = data_array[i];
            status = vxMemProbe( vmehdlc_ptr, VX_WRITE, 4, (char *) &value);
            break;
       }
      vmehdlc_ptr += pvmehdlc->ainc;
      status_array[i] = status;
    }
  }


  if (mode == READ_RO)  {
          HDLC_HANDLE   readhandle;
          readhandle = readoutBdCreateHandle();

          readoutBdReadMem (readhandle, pvmehdle->netn, pvmehdle->node, pvmehdle->addr, pvmehdle->nuse,
pvmehdle->aine, pvmehdle->bptr);
          readoutBdDeleteHandle(readhandle);
  }

  if (mode == WRITE_RO)  {
          HDLC_HANDLE   writehandle;
          writehandle = readoutBdCreateHandle();

          readoutBdWriteMem (writehandle, pvmehdle->netn, pvmehdle->node, pvmehdle->addr, pvmehdle->nuse,
pvmehdle->aine, pvmehdle->bptr);
          readoutBdDeleteHandle(writehandle);
  }

  if (mode == WRITE_RO_NR)  {
          HDLC_HANDLE   write_nrhandle;
          write_nrhandle = readoutBdCreateHandle();

          readoutBdReadMem (write_nrhandle, pvmehdle->netn, pvmehdle->node, pvmehdle->addr, pvmehdle-
>nuse, pvmehdle->aine, pvmehdle->bptr);
          readoutBdDeleteHandle(write_nrhandle);
  }

  /*if (mode == NODES)  {
*         HDLC_HANDLE   nodehandle;
*         nodehandle = readoutBdCreateHandle();
*
*         readoutBdNodesPresent(nodehandle, pvmehdle->netn, putnode list here); }*/

  if (mode == RO_RESET)  {
          HDLC_HANDLE   resethandle;
          resethandle = readoutBdCreateHandle();

          readoutBdResetNode(resethandle, pvmehdle->netn, pvmehdle->node);
  }


   return OK;
}
```

```
        void              *dpvt;               /* Device Private */
        unsigned short    prio;                /* Scheduling Priority */
        unsigned char     tpro;                /* Trace Processing */
        char bkpt;                 /* Break Point */
        unsigned char     udf;                 /* Undefined */
        char              p5  [3];             /* Created Pad  */
        TS_STAMP          time;                /* Time */
        struct link       flnk;                /* Forward Process Link */
        /* start of vmeIdle specific fields */
        long              val;                 /* Current value */
        unsigned char     sarr;                /* Status array */
        char              p127[3];             /* Created Pad  */
        long              mlst;                /* Last Val Monitored */
        void *bptr;                /* Buffer Pointer */
        void *sptr;                /* Status Pointer */
        long              addr;                /* VME address (hex) */
        unsigned short    amod;                /* VME address mode */
        unsigned short    dsiz;                /* VME data size */
        unsigned short    rdwt;                /* Read/write */
        char              p128[2];             /* Created Pad  */
        long              nmax;                /* Max. number of values */
        long              nuse;                /* Number of values to R/W */
        long              ainc;                /* Address increment (1-4) */
        unsigned short    netn;                /* Network Number */
        unsigned short    node;                /* Node Number */
};
typedef struct vmeIdleRecord vmeIdleRecord;
#endif
```

vmehdlcRecord.ascii

```
! vmehdlcRecord.ascii
!
!     Original Author: John Meier        Creighton University
!     Date:          8/21/95
!
!          Based on vmeRecord.ascii by Mark Rivers
!
!  Modification Log:
!  ----------------
!
!

#include <dbDefs.h>
#include <special.h>
#include <choiceGbl.h>
#include <dbFldTypes.h>
#define INCdbRecDesh
#include <dbRecDes.h>
#include <guigroup.h>
#include <choiceVMEHDLC.h>
RECTYP "vmehdlc"
! special is one of {NULL, SPC_NOMOD, SPC_DBADDR, SPC_SCAN, SPC_MOD, SPC_RESET,
SPC_LINCONV, SPC_CALC}
! ASL1 DBF_int    means any of {DBF_UCHAR, DBF_SHORT, DBF_LONG, DBF_ULONG}
! ASL1 DBF_float  means any of {DBF_FLOAT, DBF_DOUBLE}
! ASL1 DBF_link   means any of {DBF_INLINK, DBF_OUTLINK, DBF_FWDLINK}
!
! Prompt                    name    special          ASL1 DBF_STRING       pp  interest        size
          pmt
! Prompt                    name    special          ASL1 DBF_int          pp  interest        init
          pmt    range1     range2  cvt_type
! Prompt                    name    special          ASL1 DBF_float   pp  interest       init        pmt
          range1    range2
! Prompt                    name    special          ASL1 DBF_ENUM         pp  interest        init
          pmt        range1  range2
! Prompt                    name    special          ASL1 DBF_GBLCHOICE    pp  interest        init
          choice_set        pmt
! Prompt                    name    special          ASL1 DBF_RECCHOICE    pp  interest        init
          choice_set        pmt
! Prompt                    name    special          ASL1 DBF_CVTCHOICE    pp  interest        init
          pmt
! Prompt                    name    special          ASL1 DBF_link    pp  interest       pmt         range1
          range2    conversion
! Prompt                    name    special          ASL1 DBF_NOACCESS     pp  interest        size
          code

"Current value"          VAL  SPC_DBADDR ASL0 DBF_LONG       YES  0    0  0
"Status array"          SARR SPC_DBADDR ASL0 DBF_UCHAR       YES  0    0  0
"Last Val Monitored"       MLST SPC_NOMOD  ASL1 DBF_LONG       NO   3    0  0
"Buffer Pointer"         BPTR SPC_NOMOD  ASL1 DBF_NOACCESS   NO   4    4  void *bptr;
"Status Pointer"         SPTR SPC_NOMOD  ASL1 DBF_NOACCESS   NO   4    4  void *sptr;
"VME address (hex)"         ADDR 0       ASL1 DBF_LONG      NO   1    0  GUI_DISPLAY    CON 0
CON 1E9       CT_HEX
"VME address mode"          AMOD 0       ASL1 DBF_RECCHOICE  NO   1    0  REC_VMEHDLC_AMOD
1
"VME data size"          DSIZ 0      ASL1 DBF_RECCHOICE  NO   1    1  REC_VMEHDLC_DSIZ   1
"Read/write"            RDWT 0      ASL1 DBF_RECCHOICE  NO   1    0  REC_VMEHDLC_RDWT   1
```

```
"Max. number of values"      NMAX 0      ASL1 DBF_LONG      NO   1    32  GUI_DISPLAY    CON 0
CON 8192  CT_DECIMAL
"Number of values to R/W"     NUSE 0      ASL1 DBF_LONG      NO   1    1  GUI_DISPLAY    CON 0
VAR NMAX  CT_DECIMAL
"Address increment (1-4)"    AINC 0      ASL1 DBF_LONG      NO   1    2  GUI_DISPLAY    CON 1
CON  4    CT_DECIMAL
"Network Number"             NETN 0          ASL1 DBF_RECCHOICE  NO   1    0  REC_VMEHDLC_NETN
1
"Node Number"                        NODE 0          ASL1 DBF_RECCHOICE  NO   1    0
REC_VMEHDLC_NODE   1
```

choiceVMEHDLC.h

```
/*  choiceVMEHDLC.h */

/*
 *     Author: John Meier      Creighton University
 *     Date: 8-21-95
 *
 *    Based on choiceVME.h by Mark Rivers
 *
 * Modification Log:
 * -----------------
 *
 */

#ifndef INCchoiceVMEHDLCh
#define INCchoiceVMEHDLCh 1

/* Define RECCHOICE values.  See choiceRec.ascii for meanings */
#define REC_VMEHDLC_AMOD      0
#define REC_VMEHDLC_DSIZ      1
#define REC_VMEHDLC_RDWT      2
#define REC_VMEHDLC_NETN      3
#define REC_VMEHDLC_NODE      4
/* for record-support use */
#endif
```

choiceRec.ascii

```
! choiceRec.ascii
! cat_ascii @(#)choiceRec.ascii          1.2                    3/14/94
!
!
! Modification Log:
! ----------------
! 5-26-95 JCM took out motor table men etc., kept VME stuff
! 8-21-95 JCM added vmehdlc choices




! VME
#include <choiceVME.h>
"vme"     REC_VME_DSIZ  "D8"
"vme"     REC_VME_DSIZ  "D16"
"vme"     REC_VME_DSIZ  "D32"
"vme"     REC_VME_RDWT           "Read"
"vme"     REC_VME_RDWT           "Write"



"vme"   REC_VME_AMOD    "A16"
"vme"   REC_VME_AMOD    "A24"
"vme"   REC_VME_AMOD    "A32"
"vme"   REC_VME_AMOD    "A09"


! VMEHDLC

#include <choiceVMEHDLC.h>
"vmehdlc"          REC_VMEHDLC_DSIZ      "D8"
"vmehdlc"          REC_VMEHDLC_DSIZ      "D16"
"vmehdlc"          REC_VMEHDLC_DSIZ      "D32"

"vmehdlc"          REC_VMEHDLC_RDWT   "VME Read"
"vmehdlc"          REC_VMEHDLC_RDWT   "VME Write"
"vmehdlc"          REC_VMEHDLC_RDWT   "Read RO_Board"
"vmehdlc"          REC_VMEHDLC_RDWT   "Write RO_Board"
"vmehdlc"          REC_VMEHDLC_RDWT   "Write RO_Board No Reply"
"vmehdlc"          REC_VMEHDLC_RDWT   "Nodes Present"
"vmehdlc"          REC_VMEHDLC_RDWT   "Reset RO_Board"

"vmehdlc"          REC_VMEHDLC_AMOD   "A16"
"vmehdlc"          REC_VMEHDLC_AMOD   "A24"
"vmehdlc"          REC_VMEHDLC_AMOD   "A32"
"vmehdlc"          REC_VMEHDLC_AMOD   "A09"

"vmehdlc"          REC_VMEHDLC_NETN   "0"
"vmehdlc"          REC_VMEHDLC_NETN   "1"
"vmehdlc"          REC_VMEHDLC_NETN   "2"
"vmehdlc"          REC_VMEHDLC_NETN   "3"
"vmehdlc"          REC_VMEHDLC_NETN   "4"
"vmehdlc"          REC_VMEHDLC_NETN   "5"
"vmehdlc"          REC_VMEHDLC_NETN   "6"
```

```
"vmehdlc"        REC_VMEHDLC_NETN   "7"
"vmehdlc"        REC_VMEHDLC_NETN   "8"
"vmehdlc"        REC_VMEHDLC_NETN   "9"
"vmehdlc"        REC_VMEHDLC_NETN   "10"
"vmehdlc"        REC_VMEHDLC_NETN   "11"
"vmehdlc"        REC_VMEHDLC_NETN   "12"
"vmehdlc"        REC_VMEHDLC_NETN   "13"
"vmehdlc"        REC_VMEHDLC_NETN   "14"
"vmehdlc"        REC_VMEHDLC_NETN   "15"
"vmehdlc"        REC_VMEHDLC_NETN   "16"
"vmehdlc"        REC_VMEHDLC_NETN   "17"
"vmehdlc"        REC_VMEHDLC_NETN   "18"
"vmehdlc"        REC_VMEHDLC_NETN   "19"
"vmehdlc"        REC_VMEHDLC_NETN   "20"
"vmehdlc"        REC_VMEHDLC_NETN   "21"
"vmehdlc"        REC_VMEHDLC_NETN   "22"
"vmehdlc"        REC_VMEHDLC_NETN   "23"

"vmehdlc"        REC_VMEHDLC_NODE   "1"
"vmehdlc"        REC_VMEHDLC_NODE   "2"
"vmehdlc"        REC_VMEHDLC_NODE   "3"
"vmehdlc"        REC_VMEHDLC_NODE   "4"
"vmehdlc"        REC_VMEHDLC_NODE   "5"
"vmehdlc"        REC_VMEHDLC_NODE   "6"
"vmehdlc"        REC_VMEHDLC_NODE   "ALL_NODES"
```

mode1r_rb.c

```
/* Hardware Controls - TPC FEE READOUT: mode1r_rb
 *
 *
 * Refs: "System Test Data Paths", internal note, M. Lisa
 *     "Proposed HDLC addresses for STAR FEE RDOs", internal note, F. Bieser
 *
 *
 *  Creighton University Hardware Controls group:
 *
 *  Becky Burke, Mike Cherney, Jan Chrin, Pete Colarco,
 *  Jeff Gross, Tom McShane, John Meier, Paul Teeter
 *
 *                          22 January 1996
 *
 *
 *  This program generates NGEN events containing ADC pedestals and
 *  writes/reads each event to/from VME off-board memory using channel
 *  access routines.
 *  Pad sequences are read from a file in the "mode 1" data format
 *  (see internal note by M. Lisa) for later use by TPC data programs,
 *  e.g. pad monitor.
 *
 *
 *  Note that data is arranged in memory according to the note of
 *  F. Bieser, where the first (last) pad and time bucket numbers have
 *  values of Pad 0 (1151) and Time 0 (511).
 *
 *  This differs to the numbering scheme adopted in the "mode 1"
 *  output data files where the first (last) pad numbers have values
 *  of Pad 1 (1152).
 *
 *  It is important to be aware of this subtle difference to avoid
 *  confusion!
 *
 *
 *  Further information from:   Jan Chrin (chrin@bluejay.creighton.edu)
 *                    or John Meier (jcmeier@bluejay.creighton.edu)
 *
 *
 *
 *  Code Structure:
 *
 *  fee_memory;         Called once only
 *                Maps read_out item to pad & time_bucket
 *
 *  wrt_evt_mem;         Writes Fake ADC Data to memory
 *                Last Event also written to file (#define OUT_AFD)
 *
 *  read_val_data;       Reads Valid Data (pad sequences)
 *                             from file (#define IN_VD)
 *
 *  wrt_evt_data;       Writes Event No to file (#define OUT_EV)
 *
 *
 *  wrt_adc_data;       Writes ADC Data to file (#define OUT_D)
*       |
*       |------------>  get_adc_pads;   Gets ADC data from memory
*
*
*
*
*
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>
#include <tsDefs.h>
#include <cadef.h>
#include <ezca.h>


#define DEBUG
#define NGEN  1      /* No. of generated fake events */
#define NMAX  10  /* Maximum no. of events to read out*/
            /* When NGEN exceeds NMAX then event counter restarts at
              event #1 and overwrites the previous event files */

#define NPAD   1152       /* Maximum no. of pads  */
#define NTIM   512       /* Maximum no. of time buckets  */

#define NPSQ NPAD       /* Maximum no. of pad sequences */

#define NBYT NPAD*NTIM        /* No. of bytes/event to read out */
#define NBYT_LIMIT 1000    /* Maximum allowed byte transfer  */

#define START_ADDR  0x800000 /* -2030043136 Start Address 0x87000000 */

#define OUT_AFD "/export/home/starslow/acct/sysuser/data_files_test/adc_fake_data_"

#define OUT_EV "/export/home/starslow/acct/sysuser/data_files_test/events_written"

/* Input file is VALID_DATA_n where n=event_no */
#define IN_VD "/export/home/starslow/acct/sysuser/data_files_test/valid_data_"

/* Output file for ADC is DATA_n where n=event_no */
#define OUT_D "/export/home/starslow/acct/sysuser/data_files_test/data_"

/* prototypes */

void prn_info(char *, char *);
void prn_predefined_macros(void);

void wrt_evt_mem(int);
void wrt_evt_data(int *);
void read_val_data(int, int *);

void wrt_adc_data(int);
void get_adc_pads(unsigned short padtim[NTIM][NPAD], int);

void fee_memory(void);

int  imax(int x, int y);
int  name_ofp(char *, char *, int);
```

```
/* global variables */
int seq_start[NPSQ]={0}, seq_stop[NPSQ]={0};
int nseq=0;

unsigned short mem_array[NBYT]={0}; /* Stores data to be written to memory */

long item2pad [1152*512]={0};   /* Relates read_out item to pad & tb */
int pad2item[512][1152]={0};    /* Relates tb & pad to read_out item */


int main(int argc, char *argv[])
{
    int i, event_no=0, ierror=0;


        printf("\nStarting... \n\n");

#ifdef DEBUG
        prn_predefined_macros();
#endif


         fee_memory();  /* Called once per run; this routine fills the */
                    /* item2pad[] and pad2item[][] reference arrays */


    for (i=1; i<= NGEN; i++) {

        ++event_no;

        if (event_no > NMAX) event_no =1; /* Reset counter */

/*          wrt_evt_mem(event_no); */ /* Write Fake ADC Data to Memory */

         read_val_data(event_no,&ierror); /* Reads Valid Data */

         if (ierror) {

         printf("\n%s%d%s\n",
         "Skipping event ",event_no," because of illegal pad sequence!");

         continue;

         }

         wrt_evt_data(&event_no);  /* Write Event No. */

         wrt_adc_data(event_no);   /* Write ADC Data */

    }


   printf("\n%s%d\n\n","total no of events written = ", event_no);


    return 0;
}

void prn_predefined_macros(void)
```

```
{
/* Test the 5 predefined macros in ANSI C */

        printf("%s%s\n%s%s\n%s%s\n%s%d\n%s%d\n",
          "__DATE__ = ", __DATE__,
          "__TIME__ = ", __TIME__,
          "__FILE__ = ", __FILE__,
          "__LINE__ = ", __LINE__,
          "__STDC__ = ", __STDC__);

        if (__STDC__) printf("\n%s\n",
          "Implementation follows ANSI Standard C ");

}


void wrt_evt_mem(event_no)
{
 FILE *ofp;

 short write =3, dsiz=1, proc=1;
 unsigned short nelm=0, node=1;
 int offs=1;
 int ainc=2;
 int i, intmax=0;

 long addr = START_ADDR;
 long nbyt_get=0;
 long nbyt=NBYT, nbyt_limit=NBYT_LIMIT;


 char s1[70] = OUT_AFD;
 char s2[10];


 srand(time(NULL));


 for (i=0; i<NBYT; i++) {

  mem_array[i] = rand() % 12  + 28;  /* from 28 to 39 */

/* mem_array[i] = 4369;  */  /* pattern 1111 */
 }

/*  For Test Purposes Only */
/*
  mem_array[0] = 4369;
  mem_array[1] = 4369;
  mem_array[1152*512-1] = 4369;
  mem_array[1152*512-2] = 4369;
*/

 if (event_no == NMAX || event_no == NGEN) {

 name_ofp(s1, s2, event_no);

 ofp = fopen(s1, "wb");

 for (i=0; i<NBYT; i++)
 fprintf(ofp, "%04x\t ", mem_array[i]);

 fclose(ofp);

 }

 ezcaPut("jem_hdlevme.RDWT", ezcaShort, 1, &write);
 ezcaPut("jem_hdlevme.AINC", ezcaLong,  1, &ainc);
 ezcaPut("jem_hdlevme.DSIZ", ezcaShort, 1, &dsiz);
 ezcaPut("jem_hdlevme.NETN", ezcaShort, 1, &netn);
 ezcaPut("jem_hdlevme.NODE", ezcaShort, 1, &node);
 ezcaPut("jem_hdlevme.OFFS", ezcaLong,  1, &offs);
 ezcaPut("jem_hdlevme.ADDR", ezcaLong,  1, &addr);



  if (nbyt <= nbyt_limit) {

    ezcaPut("jem_hdlevme.NUSE", ezcaLong,  1, &nbyt);
    ezcaPut("jem_hdlevme.VAL",  ezcaShort, NBYT, mem_array);
  }

  else {

    ezcaPut("jem_hdlevme.NUSE", ezcaLong,  1, &nbyt_limit);

    intmax =  (int) ( nbyt / nbyt_limit);

        for (i=1; i<= intmax; i++) {

          ezcaPut ("jem_hdlevme.VAL", ezcaShort, nbyt_limit,
                      &mem_array[0+(i-1)*nbyt_limit]);

          addr = START_ADDR + i*NBYT_LIMIT*ainc; /* increment address */

          ezcaPut("jem_hdlevme.ADDR", ezcaLong,  1, &addr);


#ifdef DEBUG
           printf("\n %d %s %d %s \n %s %x %s \n", i,
              "memory block transfer", nbyt_limit, "bytes",
               "addr", addr, "in hex");
#endif
           }

        nbyt_get = nbyt % nbyt_limit;


        if (nbyt_get != 0) {

          ezcaPut("jem_hdlevme.NUSE", ezcaLong,  1, &nbyt_get);

          ezcaPut ("jem_hdlevme.VAL", ezcaShort, nbyt_get,
                      &mem_array[nbyt_limit * intmax]);

          }


#ifdef DEBUG
           printf("\n %d %s %d %s \n %s %d %s \n", intmax,
                "memory block transfers of", nbyt_limit, "bytes",
                "plus one transfer of", nbyt_get, "bytes");
#endif

   }

}


void wrt_evt_data(int *evt_no)
{
   FILE *ofp;

        if (*evt_no > NMAX || *evt_no == 1) {

          *evt_no = 1;
          ofp = fopen(OUT_EV,"wb");
         }

        else {
          ofp = fopen(OUT_EV,"ab");
         }

#ifdef DEBUG
             if( *evt_no == 1 || *evt_no == NMAX)
             printf("\n%s%d\n\n", "event_no ", *evt_no);
#endif

             fprintf(ofp, "%4d", *evt_no);
             putc('\n', ofp);

             fclose(ofp);

}

void read_val_data(int event_no, int *ierror)
{

    int i;
    FILE *ifp;
    char s1[70] = IN_VD;
    char s2[10];

             *ierror=0;

             name_ofp(s1, s2, event_no);

             ifp = fopen(s1,"rb");

             if (!ifp) {
             printf(
             "\n%s\n\n%s\n%s\n%s\n\n%s%d%s%d%s\n\n%s%d%s\n\n",

             " *** WARNING from read_val_data *** ",
             "The valid_data input file:", s1,
             "does not exist!",

             "You asked for ",NGEN," events - but only ",event_no-1,
             " valid data files exist!",
        "Exiting program modelr after ", event_no-1, " events");

        exit(1);
        }


        fscanf(ifp, "%d", &nseq);

        for (i = 1; i<= nseq; ++i) {
       /*      putc('\n', ofp); */
          fscanf(ifp, "%d %d",&seq_start[i],&seq_stop[i]);

          if(seq_start[i]>NPAD||seq_stop[i]>NPAD) {
          printf("\n%s%d%s%d%s%d%s\n",
          "Pad sequence ",i," with range ", seq_start[i], " to ",
          seq_stop[i]," is illegal!");

          *ierror=1;

          }
        }

        fclose(ifp);


}


void wrt_adc_data(int event_no)
{
    unsigned short padtim[NTIM][NPAD] = {{0}};
    int i,j;
    FILE *ofp;
    char s1[70] = OUT_D;
    char s2[10];

    name_ofp(s1,s2,event_no);

    ofp = fopen(s1, "wb");

    get_adc_pads(padtim, event_no);


    for (i=0; i < NTIM; ++i)

    for (j=0; j < NPAD; ++j) {

         fprintf(ofp, "%d", padtim[i][j]);
         putc('\n', ofp);

#ifdef DEBUG

       if( event_no == 1 || event_no == NGEN) {
         if(NTIM <= 20 && NPAD <= 20) {
         printf("%s %3d %3d %d\n", "bucket:pad",i ,j, padtim[i][j]);
          }
       }
#endif

    }
```

```
     fclose(ofp);

}


void get_adc_pads(unsigned short padtim[NTIM][NPAD], int event_no)
{
/* Routine to get ADC data */

     short dsiz=1, proc=1, write =1, read=2;
     unsigned short node=1, netn=0;
     int offs =1;
     int i, j, k;
     int intmax=0;
     int ainc = 2;

     long addr = START_ADDR;
     long nbyt_get=0;
     long nbyt=NBYT, nbyt_limit=NBYT_LIMIT, nuse=1;

     unsigned short vmearray[NBYT]={0};

     int diff, temp;

     int method = 1;  /* Select Method 1 (method = 1) or 2 (method != 1) */
                  /* Method 1 is usual - see below for explanation   */


     if (method == 1) {

/* Method 1 - for large ADCdata samples;                    */
/*        copies entire event from memory with one ezca call */
/*        (or more ezca calls if nbyt > nbyt_limit)        */
/*        and then directs valid data to padtim[][] array    */


       if ( nbyt <= nbyt_limit )

       ezcaGet ("jcm_hdlcvme.VAL", ezcaShort, nbyt, &vmearray[0]);

       else {

            ezcaPut("jcm_hdlcvme.DSIZ", ezcaShort, 1, &dsiz);
            ezcaPut("jcm_hdlcvme.AINC", ezcaLong,  1, &ainc);
            ezcaPut("jcm_hdlcvme.NUSE", ezcaLong,  1, &nbyt_limit);
            ezcaPut("jcm_hdlcvme.NODE", ezcaShort, 1, &node);
            ezcaPut("jcm_hdlcvme.NETN", ezcaShort, 1, &netn);

            ezcaPut("jcm_hdlcvme.OFFS", ezcaLong,  1, &offs);
            ezcaPut("jcm_hdlcvme.ADDR", ezcaLong,  1, &addr);
            ezcaPut("jcm_hdlcvme.RDWT", ezcaShort, 1, &read);
            ezcaPut("jcm_hdlcvme.PROC", ezcaShort, 1, &proc);



            intmax = (int) nbyt/nbyt_limit;


            for (i=1; i<= intmax; i++) {
```

```
            ezcaGet ("jem_hdlevme.VAL", ezcaShort, nbyt_limit,
                        &vmearray[0+(i-1)*nbyt_limit]);


            addr = START_ADDR + i*nbyt_limit*ainc; /* increment address */
     /*
          printf(" %s%x%d%d\n ","add/vmearray/mem_array ",
          addr, vmearray[0+(i-1)*nbyt_limit], mem_array[0+(i-1)*nbyt_limit]);

    */
           ezcaPut("jem_hdlevme.ADDR", ezcaLong,  1, &addr);
           ezcaPut("jem_hdlevme.PROC", ezcaShort, 1, &proc);
    /*        ezcaGet("jem_hdlevme.ADDR", ezcaLong,  1, &addr);

          printf("\n  %s %x", " addr ", addr);
    */
           }

           nbyt_get = nbyt % nbyt_limit;

          if (nbyt_get != 0)  {

           ezcaPut("jem_hdlevme.NUSE", ezcaLong,  1, &nbyt_get);
           ezcaPut("jem_hdlevme.PROC", ezcaShort, 1, &proc);

           ezcaGet ("jem_hdlevme.VAL", ezcaShort, nbyt_get,
                 &vmearray[nbyt_limit * intmax]);
          }


      }


      for (k = 1; k <= nseq; k++) {

        /* offset pad sequences by -1 for correct matching to memory */

        for (j = seq_start[k] - 1; j <= seq_stop[k] - 1; j++){

          for (i = 0; i < NTIM; i++) {

                padtim[i][j] = vmearray[pad2item[i][j]];

/* 0x7000 test      padtim[i][j] = vmearray[i*NPAD + j];    */

          }
        }
      }

/* Comparison of event written with event read-back  */
/*
        temp = 0;
        diff = 1;

     for (i = 0; i < NBYT; i++) {

       diff  = mem_array[i] - vmearray[i];
```

```
        if (diff) {  printf("\n \n %s \n \n %s \n %s %d",

        "**** ERROR **** ",

        "A comparison of data written to memory with that read back",
        "has found a disparity in event",event_no);

        temp = 1;

        printf("\n %s %d %d %d","element ",i,mem_array[i] , vmearray[i]);

        break;

        }
      }
        if(!temp)
        printf("\n %s %d \n\n","DATA WRITE/READ VALIDATED for event",
        event_no);
*/
     }

    else  {

/* Method 2 - for small ADC data samples;                              */
/*          makes several ezca calls filling padtim[][] array directly; */
/*          ainc should normally be 2                              */


    ezcaGet("jem_hdlevme.AINC", ezcaLong,  1, &ainc);
    ezcaPut("jem_hdlevme.NUSE", ezcaLong,  1, &nuse);

    ezcaPut("jem_hdlevme.RDWT", ezcaShort, 1, &read);

     for (k = 1; k <= nseq; k++) {

        for (j = seq_start[k] - 1; j <= seq_stop[k] - 1; j++){

          for (i = 0; i< NTIM; i++) {

             addr = START_ADDR + pad2item[i][j]*ainc;

/*  0x7000 test  addr = START_ADDR + (j + NPAD*i) *ainc;    */

             ezcaPut("jem_hdlevme.ADDR", ezcaLong,  1, &addr);
             ezcaPut("jem_hdlevme.PROC", ezcaShort, 1, &proc);

             ezcaGet ("jem_hdlevme.VAL", ezcaShort, 1, &padtim[i][j]);


          }
        }
     }

    }

}


int imax(int x, int y)
{
```

```
if (x>y) return x;
else
      return y;
}



int  name_ofp(char *str1, char *str2, int evt_no)
{
/* Add Event No. to output file name */

      sprintf(str2, "%d", evt_no);
      strcat(str1,str2);

      return 0;
}


void fee_memory()
{
int k, mult=1000;
long pad, time;

 item2pad[0]  = 0   *mult;
 item2pad[6]  = 384 *mult;
 item2pad[12] = 768 *mult;
 item2pad[18] = 96  *mult;
 item2pad[24] = 480 *mult;
 item2pad[30] = 864 *mult;
 item2pad[36] = 192 *mult;
 item2pad[42] = 576 *mult;
 item2pad[48] = 960 *mult;
 item2pad[54] = 288 *mult;
 item2pad[60] = 672 *mult;
 item2pad[66] = 1056*mult;


 for (k=0; k<72; k++) {

    if (k%6 != 0)
    item2pad[k] = item2pad[k-1] + 16 *mult;

 }

 for (k=72; k<1152; k++) {

    item2pad[k] = item2pad[k-72] + 1 *mult;

 }


 for (k=1152; k<1152*512; k++) {

    item2pad[k] = item2pad[k-1152]  + 1;

}

 for (k=0; k<1152*512; k++) {
        pad = item2pad[k]/1000;
        time= item2pad[k] - pad*1000;
        pad2item[time][pad] =k;

    }

  }
```

References

[CQ 85] C. Quigg, *Elementary Particles and Forces,* Readings from Scientific American, Particles and Forces At the Heart of the Matter (1990).

[DO 91] J.E. Dodd, *The Ideas of Particle Physics: an introduction for scientists.* Cambridge University Press, Cambridge (1991).

[FE 91] T. Ferbel, *Experimental Techniques in High-Energy Nuclear and Particle Physics.* World Scientific Publishing Co., Singapore (1991).

[HY 83] B. Hyams et al., *Nucl. Instr. Meth.* 205 (1983).

[JC 96] J. Chrin et. al., *Results from the STAR TPC System Test,* submitted to IEEE Transactions on Nuclear Science, (1996).

[JG 94] J. Gross, M. Cherney, T.S. McShane, *A Unified Control System for the STAR Experiment,* IEEE Transactions on Nuclear Science, 41 (1994)184.

[JG 95a] J. Gross, *Development of a Distributed Slow Control System for the Solenoidal Tracker at RHIC (STAR).* Masters thesis, Creighton University, 1995.

[KL 86] K. Kleinknecht, *Detectors for Particle Radiation.* Cambridge University Press, 1986.

[LO 92] T. Lohse, W. Witzeling, *Instrumentation in High Energy Physics.* World Scientific Publishing Co., Singapore (1992).

[LU 93] T. Ludlum et al., *BNL-49177 Informal Report* (1993).

[PB 95] P. Barale, F. Bieser, J. Hunter, S. Jacobson, S. Klein, C. McParland, *The STAR TPC FEE HDLC Link.* STAR internal note (1995).

[PE 92] A. Peisert, *Instrumentation in High Energy Physics.* World Scientific Publishing Co., Singapore (1992).

[SA 77] F. Sauli, *Principles of Operation of Multiwire Proportional and Drift Chambers*, CERN 77-09 (1977).

[SK 95] S.R. Klein et al., *Front End Electronics for the STAR TPC,* submitted to IEEE Transactions on Nuclear Science. (1995).

[ST 92] STAR Collaboration, *Conceptual Design Report for the Solenoidal Tracker at RHIC (STAR)*. LBL Document (1992).

[WO 94] C. Wong, *Introduction to High-Energy Heavy-Ion Collisions*. World Scientific Publishing Co., Singapore (1994).