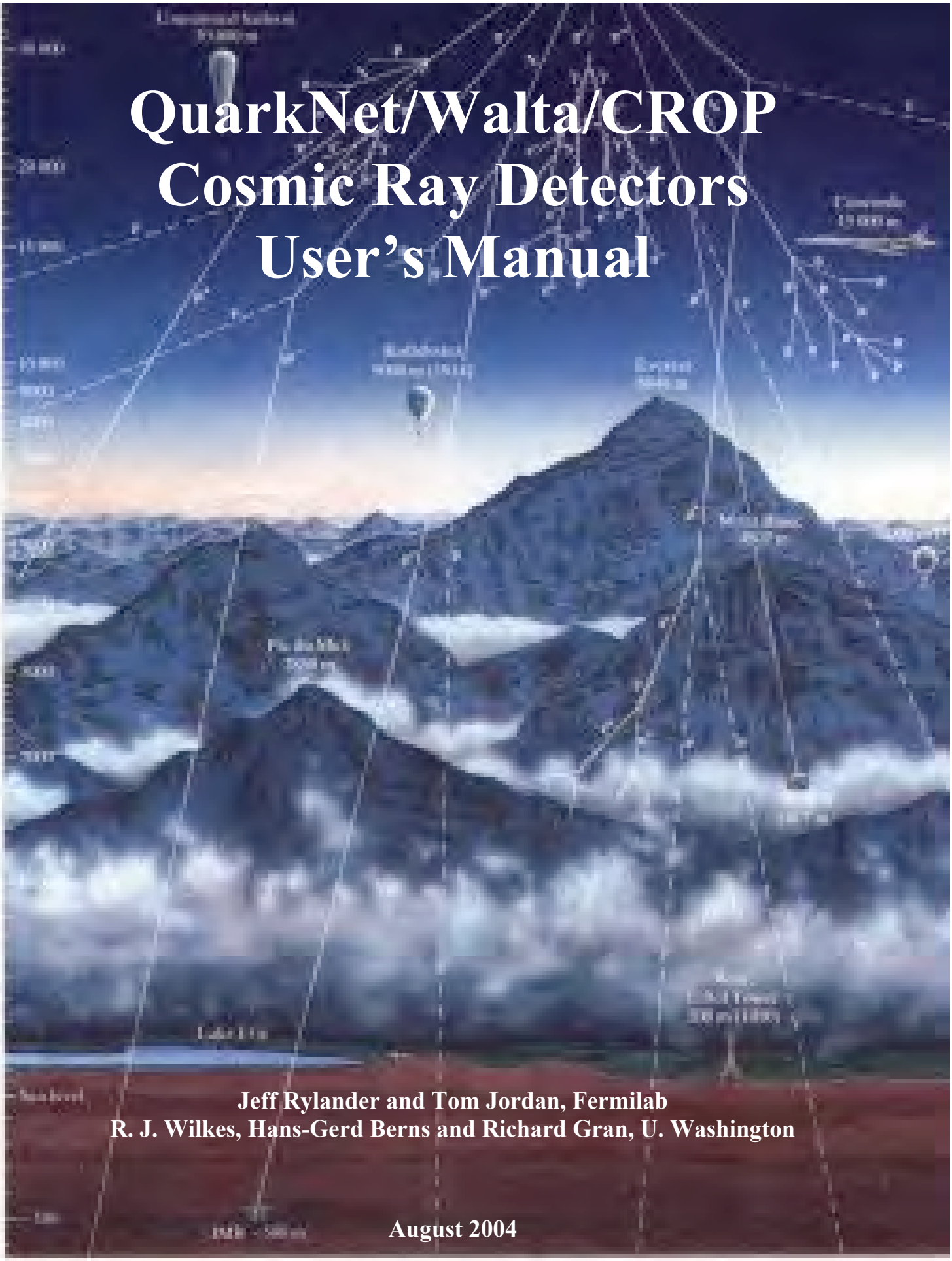
# QuarkNet/Walta/CROP Cosmic Ray Detectors User's Manual



Jeff Rylander and Tom Jordan, Fermilab
R. J. Wilkes, Hans-Gerd Berns and Richard Gran, U. Washington

August 2004

# QuarkNet/Walta/CROP Cosmic Ray Detectors User's Manual

This manual provides information for setting up and using a cosmic ray detector with the **QuarkNet Version 2 Data Acquisition (DAQ) board** and ancillary hardware. It serves both beginning and advanced users.

Much of the technical information was originally compiled in a user's manual written by R. J. Wilkes. Hans Berns, Rik Gran, Sten Hansen, and Terry Kiper contributed some of the technical documentation with input from the practical experience of many beta testers.[1]

Although some of the components of your cosmic ray detector may have come from various sources, the heart of the detector is the QuarkNet DAQ board. The development of this board is a collaborative effort involving Fermilab, the University of Nebraska and the University of Washington. Appendix A has a description of the history of this project.

**Project Development Team**

**Fermilab:** Sten Hansen, Tom Jordan, Terry Kiper

**Univeristy of Nebraska:** Dan Claes, Jared Kite, Victoria Mariupolskaya, Greg Snow

**University of Washington:** Hans Berns, Toby Burnett, Paul Edmon, Rik Gran, Ben Laughlin, Jeremy Sandler, Graham Wheel, Jeffrey Wilkes

[1] For more technical documentation on the DAQ board, see http://www.phys.washington.edu/~walta/qnet_daq/

# Table of Contents

# Chapter 1: In the Classroom

## Cosmic Ray Experiments

For most of today's particle physics experiments, large accelerators accelerate particles to very high energies. However, it is possible to do high-energy physics in your school without a particle accelerator! Nature serves as an accelerator for cosmic rays—particles that are of astrophysical origin (It is not completely understood!) and seem to be everywhere throughout the universe. These "primary" cosmic rays, which are mainly protons (and a few heavier nuclei), interact with nucleons in the earth's upper atmosphere in much the same way that fixed target collisions occur at particle physics laboratories. When these primary particles interact with nucleons in the atmosphere, they produce mainly pions and kaons. In the collisions, if most of the incoming momentum is transferred to an atmospheric proton, the following reactions are common:

$$p + p \rightarrow p + p + \pi^+ + \pi^\circ + \pi^-$$

and

$$p + p \rightarrow p + n + \pi^+ + \pi^+ + \pi^-.$$

If most of the momentum is transferred to a neutron, then these reactions are common:

$$p + n \rightarrow p + p + \pi^+ + \pi^- + \pi^-$$

and

$$p + n \rightarrow p + n + \pi^+ + \pi^\circ + \pi^-.$$

The products of such interactions are called "secondary" particles or "secondary" cosmic rays. Some of these products, however, are very short -ived and generally decay into daughter particles before reaching the earth's surface. The charged pions, for instance, will decay into a muon and a neutrino:

$$\pi^- \rightarrow \mu^- + \bar{\nu}_\mu$$

or

$$\pi^+ \rightarrow \mu^+ + \nu_\mu.$$

Although these reactions are not the only possibilities—they are examples of common reactions that produce secondary particles and their daughters. Counting all secondary particles detected at sea level, 70% are muons, 29% are electrons and positrons, and 1% are heavier particles.

If these secondary particles have sufficient energy, they may in turn interact with other particles in the atmosphere producing a "shower" of secondary particles. The particles in this shower will strike a wide area on the earth's surface (perhaps several $m^2$ or even $km^2$) nearly simultaneously. Figure 1 illustrates one of these showers.

Although your students can do several cosmic ray experiments with a single classroom setup, the QuarkNet website <http://quarknet.fnal.gov/grid/> provides student access to data from multiple cosmic ray detectors to detect and study larger showers. Several professional versions of this same experiment are taking data that may help determine the origin of some of these highly energetic primary cosmic rays.

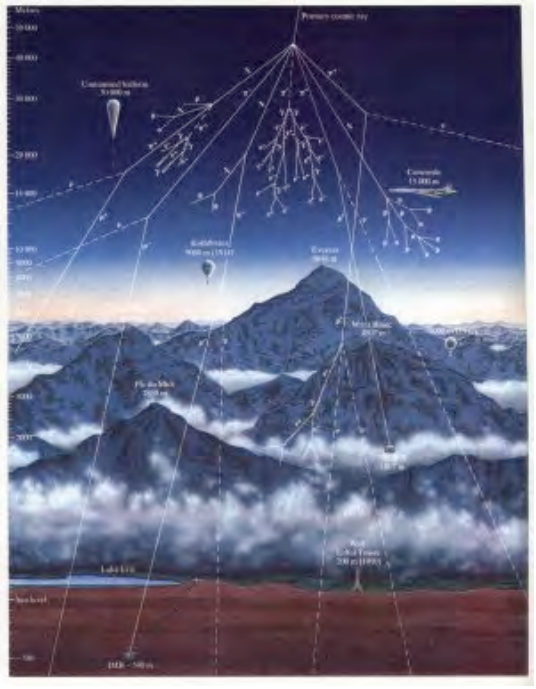


*Figure 1. A cosmic ray shower produced by a primary cosmic ray entering the earth's atmosphere.*

## How Can I Use a Cosmic Ray Detector In My Classroom*?*

The very nature of cosmic ray experiments is quite different from traditional labs that can be completed in a lab period or a couple of lab periods. Cosmic ray experiments typically require a great deal of run time to collect data. Extra class time, however, is not something most teachers have in an already full course. You may also wonder how to provide each student time to do an experiment when you have only one setup. These are realistic questions. As a possible solution, the following approach allows you not only to incorporate these experiments into your course but also to help your students discover how high-energy physics experiments are *really* done.

Organize the cosmic ray experiments as a long-term project that spans a quarter or even more. If students work on this project intermittently, you can run these cosmic ray experiments in parallel with your "standard" curriculum with days set aside periodically for cosmic ray detector work.

After an introduction to cosmic rays, the equipment, and the research questions that physicists ask in this field, have student groups write up a proposal for the experiment that *they* want to perform. These experiments might include calibration and performance studies, muon lifetime experiments, shower studies, or flux studies as a function of one of many variables, e.g., time of day, solar activity, east/west asymmetry, angle from vertical, barometric pressure, etc. Your students can also look for particles that are strongly correlated in time and may be part of a wide-area shower.

Once proposals are accepted, give each student "collaboration" a few days to a week to run their experiment. In some cases, two groups might pool their time and share the same data but for different physics goals. This is not unlike how *real* high-energy physics experiments are done! If time permits, students can do a second run with the student-suggested modifications.

While one group is using the hardware, other groups can design their setup, analyze their data (or other data), visit the "Poster" section of the QuarkNet cosmic ray website to research the results of other student groups, etc. Having your students post their work here and/or having them give oral presentations (sometimes with guest physicists!) may make for a great summary to the project.

## What Experiments Can My Student Perform?

There are four categories of experiments that students can do with the comic ray detectors:

1. Calibrations and performance studies
2. Flux experiments
3. Muon lifetime experiments
4. Shower studies

## Calibrations and Performance Studies

Before students can "trust" the cosmic ray equipment, they can and should do some calibrations to study the response of the counters and the board. Calibration studies include plateauing the counters, threshold selection and barometer calibration. These are discussed in Chapter 5. In addition, the QuarkNet online analysis tools include a "system performance" study for uploaded data. The details of this study are outlined in Chapter 5 and on the website.

## Flux Experiments

Your students can do a variety of flux experiments investigating such things as cosmic ray flux as a function of time of day, solar activity, east/west asymmetry (showing the $\mu^+/\mu^-$ ratio by assuming these charged particles will bend in the earth's magnetic field), angle from vertical, barometric pressure, altitude. The list goes on. This can be an exciting set of experiments as students study the factors that *they* want to test. Chapter 5 discusses flux studies in more detail.

## Muon Lifetime Experiment to Verify Time Dilation

A classic modern physics experiment to verify time dilation is the measurement of the muon mean lifetime. Since nearly all of the cosmic ray muons are created in the upper part of the atmosphere (≈30 km above the earth's surface), the time of flight for these muons as they travel to earth should be at least 100 μs:

$$t_{flight} = \frac{v_{muon}}{d} = \frac{3x10^8\, m/s}{30x10^3\, m} = 100\mu s\,.$$

This calculation assumes that muons are traveling at the speed of light—anything slower would require even *more* time. If a student can determine the muon lifetime (as described in Chapter 5) and show that it is significantly less than this time, they are presented with the wonderful dilemma that the muon's time of flight is longer than its lifetime!

This time dilation "proof" assumes that all muons are created in the upper atmosphere. Although this is actually a good approximation, your students cannot test it. However, by using the mean lifetime value and by measuring flux rates at two significantly different elevations, you *can* develop experimental proof for time dilation. This experiment requires you to have access to a mountain, an airplane, *or* collaboration with a team from another school that *is* at a significantly different altitude! Here is a wonderful opportunity for schools to work together proving time dilation. A very thorough explanation of this experiment is outlined in the 1962 classroom movie titled, "Time Dilation: An Experiment with Mu Mesons." [2] This movie helps you (and your students) understand how the muon lifetime measurement (along with flux measurements at two different altitudes) can be used to verify time dilation.

## Shower Studies

With the GPS device connected to your DAQ board, the absolute time stamp allows

[2] This 30 minute movie can be ordered on CD for $10 from http://www.physics2000.com/.

a network of detectors (at the same site or at different schools) to study cosmic ray showers. Your students can look for small showers or collaborate with other schools in your area to look for larger showers.

The QuarkNet online analysis tools allow students to not only look for showers but to make predictions about the direction from which the shower (and thus the primary cosmic ray) originated. Details for doing shower studies can be found in Chapter 5 and on the QuarkNet website.

# Chapter 2: Hardware Introduction

## Hardware Overview

Figure 2 shows a typical cosmic ray detector setup composed of:

1. Counters—scintillators, light guides, photomultiplier tubes and bases (two shown).
2. QuarkNet DAQ board.
3. 5 VDC adapter.
4. GPS receiver.
5. GPS extension cable.
6. RS-232 cable (to link to computer serial port).
7. Optional RS-232 to USB adapter (to link to computer USB port instead of serial port).
8. Lemo signal cables.
9. Daisy-chained power cables.

For this setup, the DAQ board takes the signals from the counters and provides signal processing and logic basic to most nuclear and particle physics experiments. The DAQ board can anlyze signals from up to four PMTs. (We show two in the figure.) The board produces a record of output data whenever the PMT signal meets a predefined trigger criterion (for example, when two or more PMTs have signals above some predetermined threshold voltage, within a certain time window). The output data record, which can be sent via a standard RS-232 serial interface to any PC, contains temporal information about the PMT signals. This information includes: how many channels had above-threshold signals, their relative arrival times (precise to 0.75 ns),

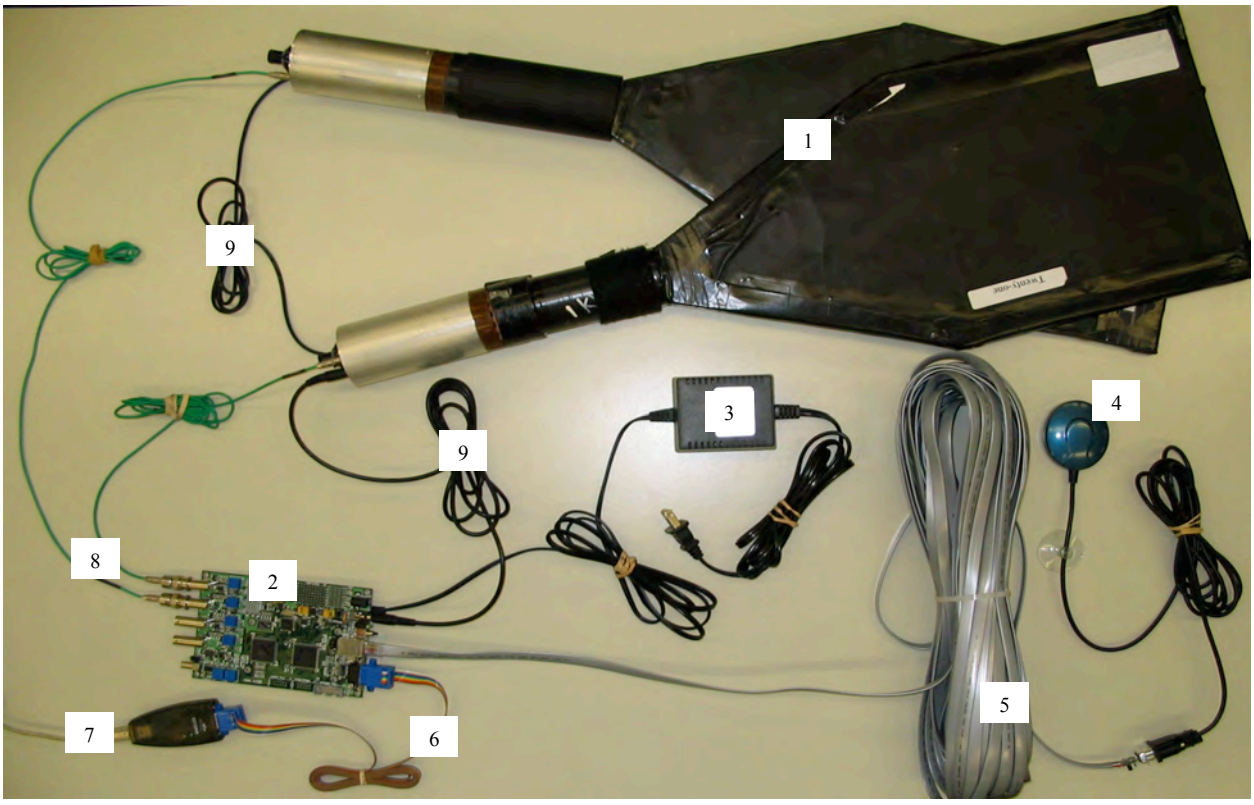



*Figure 2. A typical cosmic ray detector setup.*

and the starting and stopping times for each detected pulse. In addition, an external GPS receiver module provides the absolute UTC time of each trigger, accurate to about 50 ns. This allows counter arrays using separate DAQ boards—such as different schools in a wide-area array or two sets of counters at the same site—to correlate their timing data. Keyboard commands allow you to define trigger criteria and retrieve additional data, such as counting rates, auxiliary GPS data, and environmental sensor data (temperature and pressure).

This chapter includes an overview of each major component shown in Figure 2. Additional details regarding some components are in the appendices.

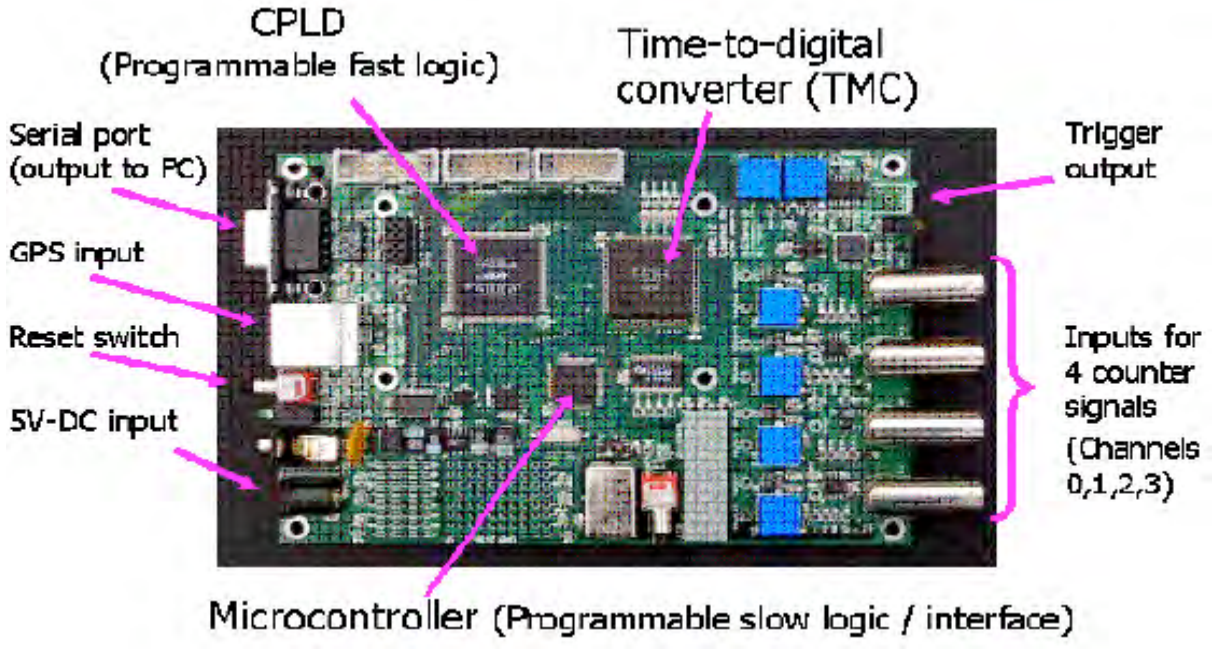



*Figure 3. The QuarkNet DAQ v2 board with major components labeled.*

## DAQ Readout Board

Figure 3 is a picture of the DAQ board with several major components labeled. This board is the logic link between the scintillation counters and the PC. The board provides discriminators and trigger logic for four channels of PMTs. The board includes five built-in scalers, allowing simultaneous counts of singles on each channel, plus triggers at whatever logic level you specify (2- to 4-fold majority logic).

A standard RS-232 interface can be connected to any PC (Windows, Linux or Mac). The datastream consists of simple ASCII text lines readable by any terminal-emulator program. In addition, a GPS clock provides accurate event time data synchronized to Universal Time (UTC), so widely separated sites can compare data. In datastream mode, the DAQ board outputs a series of text lines reporting event data: trigger time in UTC (with 24 ns resolution and absolute accuracy about 100 ns), leading and trailing edge times for each pulse recorded within the coincidence time window (with 0.75 ns precision), and data from the GPS and internal clocks. Additional keyboard commands allow reading of temperature, barometric pressure and other sensors.

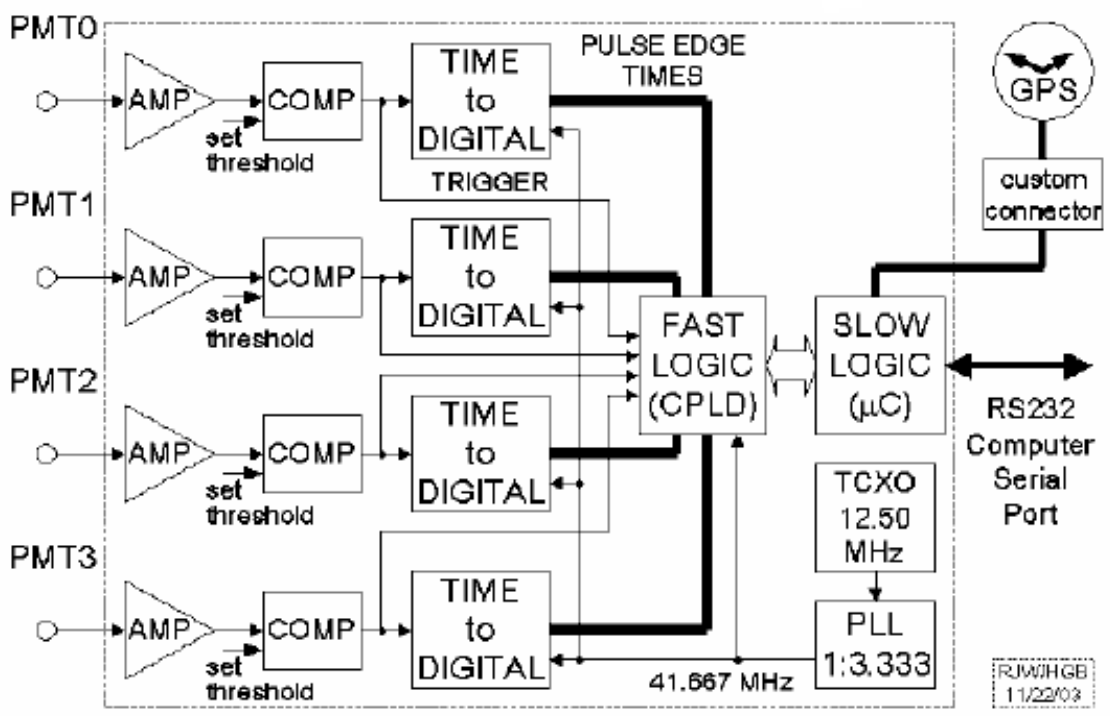



*Figure 4. Block diagram of the QuarkNet DAQ v2 board.*

## Board Components

Figure 4 is a block diagram of the QuarkNet DAQ v2 board. (See Appendix B for a schematic diagram.)

### Discriminators

PMT signals are first pre-amplified by a factor determined from a set of changeable resistors (Amplification factor x10 is used for QuarkNet, x2 by CROP, x1 by WALTA.) Discriminators are implemented using voltage comparator chips with the reference threshold voltage varied via screwdriver adjustable potentiometers (trimpots). Threshold levels can be set from 0 to -750 mV by adjusting each potentiometer while monitoring voltage at a nearby test point. It is important to remember that the voltage comparators look at the amplifier output, so raw PMT signal levels are multiplied by any amplification factor present before being compared. For example, if you used a -30 mV threshold with NIM discriminators and have x10 amplification, your threshold level on the QuarkNet DAQ board should be set to -300 mV.

### Complex Programmable Logic Device (CPLD)

Trigger logic is implemented using a CPLD chip. Software revisions for this chip must be prepared using special software but can be downloaded via the serial port. This flexibility allows the engineers to distribute updates that alter the fast logic, if necessary. Any trigger logic level from singles to 4-fold can be set by keyboard commands to the board. Majority logic is used: any combination of three active channels causes a trigger at the 3-fold level, for example.

### Time-to-Digital Converters (TDCs)

Discriminator output pulses are fed into TDCs[3] which measure the arrival time of leading and trailing pulse edges. TDCs keep track of their state (high or low) at 0.75 ns time intervals. If the trigger criterion is satisfied, the TDC data are latched and read out, giving leading and trailing edge times for each channel relative to the trigger time in units of 0.75 ns. This allows you to calculate PMT pulse widths (time over threshold, or TOT, see Figure 5) as a crude estimate of pulse area and thus energy.

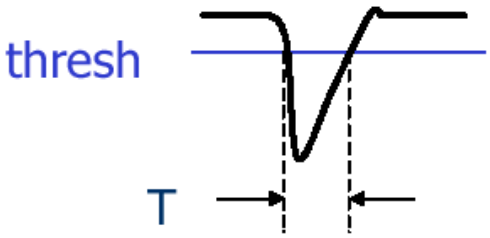



*Figure 5. Typical PMT pulse from which the Time Over Threshold (TOT) can be determined.*

### Microcontroller (MCU)

The MCU is really just a special-purpose CPU that provides the onboard 'slow' logic (with a time scale of microseconds, not nanoseconds) to interface the board to you via a terminal window or equivalent on your PC. At present, the MCU can be reprogrammed to redefine functionality only by using special software and burn-in hardware.

### Auxiliary Sensors

There is a temperature sensor built into the microcontroller chip. This sensor is present so that CPU temperature can be measured. This temperature and the supply voltage are reported whenever the board is started up. While the board components are rated for temperatures between –20°C and +80°C, the board temperature should not normally go above +50°C.

A second temperature sensor is located on the booster board built into the GPS cable's "far end" DB9 connector. This sensor can be used to log outdoor temperature at the counter locations, and is read out with a keyboard command. The GPS module may be damaged if its temperature goes below –40º C or above +85º C.[4]

A barometric pressure sensor is built into the board as well. It can also be calibrated and read out (in units of kPa) with a keyboard command.

## Board Functionality

### Threshold Detection and Setting

After amplification, the PMT signals are fed to voltage comparators. The comparators set a HIGH logic level whenever the amplified PMT signal exceeds their negative threshold voltage setting as shown in Figure 6. This logic level has 0.75 ns resolution. This means that we can measure the time difference between pulses on different channels *at the same site* down to less than a nanosecond! We depend on the GPS clocks, with 24 ns resolution, for timing *between* school sites.

[3] "TDC" is a generic term in particle physics technology. The specific chips used on the board are TMCs ("Time Measurement Chips"). You may see this designation in some documentation.

[4] At DAQ card power-up or reset, the temperature on the DAQ card (actually, inside the MCU chip) is reported. Note that this is *not* a good indicator of air temperature at the card site, even if the card is located outdoors, since the chip generates considerable heat while operating. It *is* a useful way to make sure the card is healthy, however! Temperatures over +40º C may be dangerous to the card's chips.

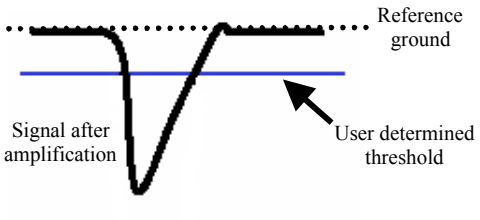



*Figure 6. A typical negative PMT pulse showing the user-defined threshold value compared to reference ground.*

To measure the current threshold voltage value, compare the test points, to ground. The test points for each threshold are labeled "TPVTH" and the ground labeled "TPGND" in Figure 7.

Setting the threshold level too high will miss all but those events that correspond to a large amount of energy deposited in the detector; setting the threshold too low will result in background noise within the electronics being mistaken as event counts.

So how do you determine the appropriate threshold setting? Consider the graph Figure 8 on the next page as a 'ballpark' determination of the optimum threshold value to use for a channel. For the data shown in this graph, the PMT voltage was set to that determined by plateauing the counter (see section 6.1 for help on how this is done) and kept constant throughout the experiment.

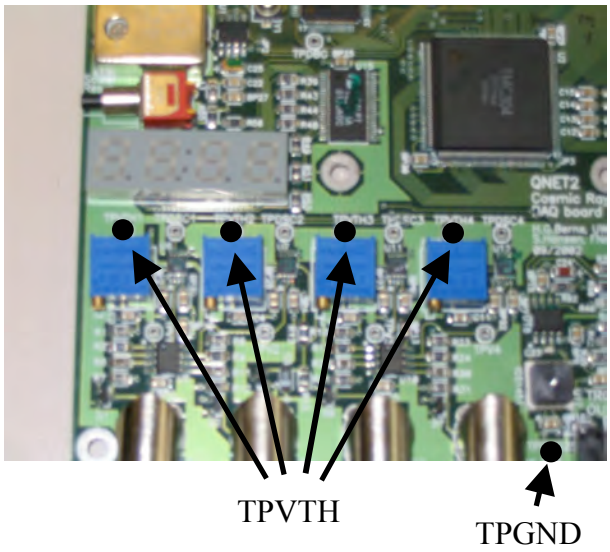



*Figure 7. Close-up picture of the DAQ board showing threshold test points used to measure threshold voltage for each channel.*

The plot shows the singles rate in that channel as a function of threshold value. You can see that the counting rate decreases as the threshold setting increases. Once the threshold value became as high as 0.5 V, the counting rate decreased more linearly. The "kink" in the graph may be used to suggest that noise contaminated the counting rate much more significantly when the threshold was set below 0.5 V. Operating slightly above this value is probably the optimum threshold value to choose.

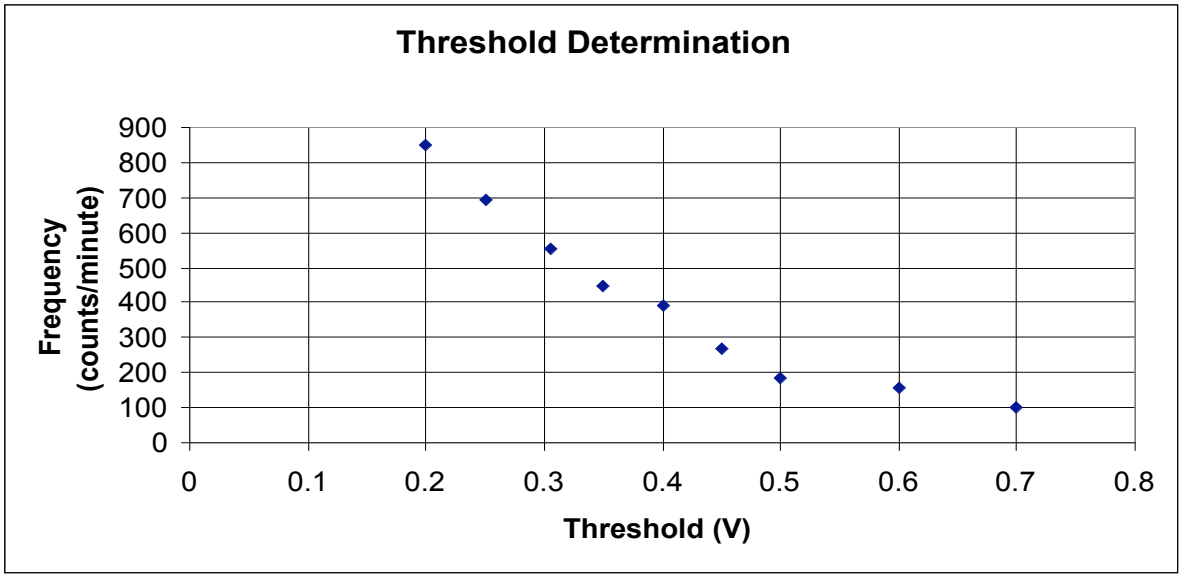



*Figure 80. A graph of frequency vs. threshold for a counter used to determine threshold setting.*

## Coincidence Logic

The DAQ board can also determine whether signals in separate channels are coincident in time. For example, if the trigger criterion is set to 2-fold, then as soon as any channel goes above threshold, a time window is opened. (Window time-width is adjustable.) If any other channel goes above threshold during this time window, *all* event data are latched and outputted for the overlap time interval when *both* are active. Notice that pulse data are reported for a time interval that is *not* of fixed length but just covers the overlap period when two or more channels are active. Leading and trailing edge times are reported for *any* active channels (not just for the two channels that launched the trigger), with empty data entries for channels that remained inactive during the trigger window. For a single event trigger, the DAQ board may need to output several lines of data. The first line has an "event flag" for identification. Any following lines without this flag are simply additional data for the same event.

## Rate Counters

The card has five built-in counters, numbered 0 through 4: counters 0 through 3 record the singles count for each channel, and counter 4 records the trigger count for whatever coincidence level you have set. By zeroing these counters with a software command, then running the board for a fixed length of time and reading out the counters at the end, you can obtain singles rates for each channel as well as the coincidence rates.

# Power Supply

The board requires a stable +5 VDC power supply, with 800 mA or greater output current. A 110VAC to 5VDC/2.4A modular switching power supply, of the type used for many small electronic devices, is supplied with the board. Replacements are readily available at Radio Shack or a similar electronics store.

PMT bases provided by QuarkNet can be powered by the same +5 VDC power supply

that is plugged into the DAQ board. These bases are daisy-chained from the board using a common male-male phono jack cable which is also readily available at Radio Shack or similar store.

Note: It is important to distinguish these DAQ power supply modules from other units used for computers or other electronic devices requiring DV voltage. Connection of the wrong power supply to your DAQ board will probably fry it.

# GPS Receiver

The external LeadTek GPS module shown in Figure 9 is connected through a special cable which has a seemingly-standard D connector (serial port connector) on the far end. This connector mates with the commercial GPS module's serial connector. However, the special cable's D connector actually contains a tiny circuit board with components needed to ensure good performance over 100' or longer cables. The GPS clock will not work if the special cable is not used.

## Interfacing the GPS with the DAQ Board

The external GPS module is interfaced to the Quarknet DAQ board with a special extension cable, which can be over 100' long

*Figure 9. The LeadTek GPS receiver and cable. The cable shown here can be connected to a GPS extension cable.*

if necessary. Although the GPS module's RS-232 D plug appears conventional, it has been modified with a special connector, which contains the circuitry required to deliver DC power to the module and interface the 1PPS (1 Pulse Per Second) signal to the board. This connector should *not* be inserted directly into your PC's serial port.

You may wish to use the GPS module alone to observe satellite data without operating the DAQ card. For stand-alone operation of the GPS module, you need to construct a special adapter to allow connection of a +5VDC power supply directly to the module—you can use the same power supply used for the DAQ card. The wiring diagram for the adapter is shown in Figure 10 on the next page. Parts shown in the wiring diagram can be obtained at most electronics supply stores. The LED is optional and just blinks to display the 1PPS signal, indicating valid data.

The GPS module itself is weatherproof, but the connectors should be protected. Also, the screws on the cover of the GPS module tend to rust, so they should be covered with plastic paint or nail polish before deployment outdoors.

## GPS Startup

Once powered-up, the GPS module should quickly "find itself" if it is in a location with a clear view of at least half the sky. Usually it does *not* work well through windows and should be physically outdoors. The receiver will lock onto four or more satellites, download the data needed to operate accurately and start averaging its position and clock settings at one-second intervals. Within a few minutes after startup, you should obtain accurate GPS data. The unit's LED display blinks red when first powered-up and searching, and changes to steady-

green when it has acquired enough satellite data to locate itself accurately. Time data are not accurate until then.

The GPS module provides several kinds of data. The commercial GPS module directly supplies the date and "coarse" time (in UTC, not local time) down to milliseconds, or three decimal places after seconds, and reports its geographic location in latitude and longitude down to the equivalent of a few 10s of meters.

## The "1PPS" Signal

The stock LeadTek GPS module was modified slightly for our application to allow the more precise timing we need down to 10s of nanoseconds (eight decimal places following integer seconds). The GPS receiver outputs a logic pulse at the beginning of each UTC second, called the 1PPS (1 Pulse Per Second) signal.

The GPS receiver sends two types of datastreams to the board. The first is RS-232 ASCII data telling what time it is, at what latitude, longitude and altitude the receiver is, and information about the satellites the receiver is using. The other data is a 5V 100ms pulse telling exactly when the data is true. Each stream of 5V 100ms pulses arrives every second, thus the 5V pulse is named 1 pulse per second (1PPS). The microcontroller on the board records the counter value during which the pulse is received. The time is according to a counter running at ~41.67 MHz.

In principle, the leading edges of 1PPS signals from GPS receivers anywhere in the world are all in synch, to within the accuracy of the non-military GPS system (about 100 ns.) This feature allows accurate time synchronization between school sites. The special connector and cable attached to the commercial GPS module transmits the 1PPS signal about 100' or more without excessive timing degradation.

## How to Calculate Event Times

The DAQ board has onboard a 41.667 MHz oscillator, which "ticks" every 24 nanoseconds ($1/41.667 \times 10^6$ sec). Such a device does not maintain its frequency very accurately, and our oscillator's frequency may drift by 10-100 Hz from its nominal value (perhaps even more under extreme temperature changes). A counter (scaler) keeps counting the clock ticks. Whenever this 32-bit counter reaches its maximum capacity of 4.3 billion, approximately every 100 seconds, it just "rolls over" back to zero, like an old car's odometer, and continues counting. Clock calibration occurs the internal clock counter every time a 1PPS signal arrives.

Find the precise event time by getting the date and time down to the nearest second from the GPS module's coarse time (with one correction described below), finding the number of clock ticks between the last 1PPS and the event trigger, and dividing by the number of clock ticks between the last two 1PPS pulses to get the fraction of a second down to 24 ns precision. Of course, you have to know whether the counter rolled over during the last second when doing the arithmetic.

There is a caveat to the previous paragraph: If the data rate is *very* slow—less than about one event per 100 sec—the internal counter may roll over more than once between readings. In this case, all bets are off, and the seconds cannot be reliably calculated. The next firmware update will include commands to produce an output line at each 1PPS, and/or automatic DG commands at regular intervals, ensuring that the necessary

information is logged between counter rollovers.

One minor correction is also needed: the GPS module reports the time down to the millisecond, but there is a delay of a fraction of a second relative to the 1PPS edge. For example, when the 1PPS signal says it is exactly 12:01:25.000000000, the GPS time field in the data record may say it is 12:01:24.850. This delay (reported as +0.150 sec in this example) will cause you to associate the wrong integer second with the event time if you are not careful. (The GPS time delay may be negative, i.e., a *forward* shift, with the ASCII time data ahead of the 1PPS signal.) The DAQ board is programmed to list the delay in milliseconds for each data record so the reported time in seconds can be corrected to match the 1PPS data. The algorithm described in Appendix D implements these calculations.

For people who like really deep-down details, the delay mentioned above, which you can determine and correct, is not the whole story. There is an additional delay of 150-250 ms, which is unresolvable, since it is buried in the manufacturer's firmware. Luckily, it is too small to cause the rounding procedure described above to go wrong. Remember, all this is just about determining the *integer seconds* part of the time. Nanosecond-level timing is not affected. Also, we have not discussed corrections for the antenna cable delay—the event time calculated is when coincidence occurs at the DAQ board, not at the counters. This is of no consequence as long as all school sites in an array use approximately the same cable lengths, within a few 10s of meters. The algorithm showing how to get the precise event time is shown in Appendix D.

## Using the GPS without the DAQ Board

To use the GPS module via the serial port of a PC (without the DAQ board) requires the construction of a simple adapter circuit shown in Figure 10. The LED is optional and simply blinks to display the 1PPS signal and indicates when the GPS module has locked onto a position and time fix. 5VDC power can come from a 110VAC adapter or a PC's PS/2 mouse jack. Power drawn is equivalent to what a mouse would consume, and will not affect your computer. You can substitute a connector to mate with a standard 5VDC power adapter, available from any electronics store.

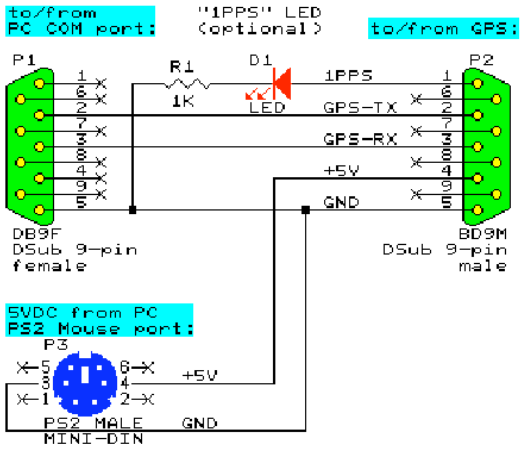



*Figure 10. The adapter circuit wiring diagram required when the GPS is connected directly to the computer (without connection to the DAQ board).*

## Scintillation Counters

A typical counter used in this experiment is shown in Figure 11. A counter is comprised of a plastic scintillator, a light guide, a PMT, and a PMT base which provides the necessary operating voltage for the PMT and sends an electrical signal to the DAQ board when a light pulse is "seen" by the PMT. For the counter shown in Figure 11, only the square portion of the paddle actively scintillates. The trapezoidal shape is merely a

light guide which directs the light toward the PMT using total internal reflection. When calculating the surface area of the active portion of your detector, use only the square portion.

*Figure11. A typical cosmic ray counter consisting of scintillator and light guide (the flat paddle region to the right), the PMT (black cylinder), and the PMT base (light-colored cylinder on the left end.)*

It will be necessary to "plateau" your counter before the counting rate can be trusted. Small fluctuations in the high voltage can offer significant variations in counter response if the operating voltage is not near the voltage plateau. Details for how to plateau a counter are in Chapter 6.

## Scintillators

Since muon flux at sea level is ~1 x $10^{-2}$ muons/$cm^2$/sec, and since these comprise about 70% of the cosmic rays at sea level, a detector of significant surface area will help to increase the counting rate to 10s or even 100s of events each minute. If shower experiments or flux studies are of interest, then increasing surface area is the primary way to increase the counting rate. If muon lifetime experiments are of interest, however, then increasing the thickness of the scintillator will also increase the counting rate since only the muons that have stopped and decay *within* the detector can be used for lifetime studies.

The process by which a scintillator scintillates light when a changed particle traverses is not discussed in this manual. This phenomena, however, is an important one in helping your students understand the operation of the detectors.

## PMT/Base

Along with the production of the DAQ board, QuarkNet and Fermilab have produced a Cockcroft-Walton base that can be used with an RCA 6342 PMT or similar 2" 10-stage tube. This base does not require the use of a high voltage power supply but is powered by the same +5VDC supply that is used for the DAQ board. The base is housed in an aluminum can (see Figure 12). Appendix B includes a schematic for this PMT base. If a separate tube and base are used in your experiment, you may need to make slight resistor modifications to the board in order to achieve the proper signal size after amplification.

The PMT voltage across the first and last dynode—the operating tube voltage—ranges from 0–1600 VDC for the RCA 6342 PMT and the corresponding base provided by QuarkNet. A potentiometer at the end of the base can be used for adjusting this voltage. The "pot" is marked off from 0 to 10 which correspond linearly to the voltages from 0 – 1600 V. This means that increasing the pot by "1 unit" corresponds to an increase in 160 V across the first and last dynode.

If you want to check the value of the tube voltage, two test points on the end of the base are provided to probe with a standard DC low voltage voltmeter (see Figure 12). This probe voltage measurement must be multiplied by 385 to obtain the actual tube voltage.

An important question to ask is, "What should be the voltage setting for my counter?" Because the response of each counter is unique, the answer to this ques-

tion is different for each counter. To determine the operating voltage for *a particular* counter, you must "plateau the counter." Details can be found in Chapter 6.

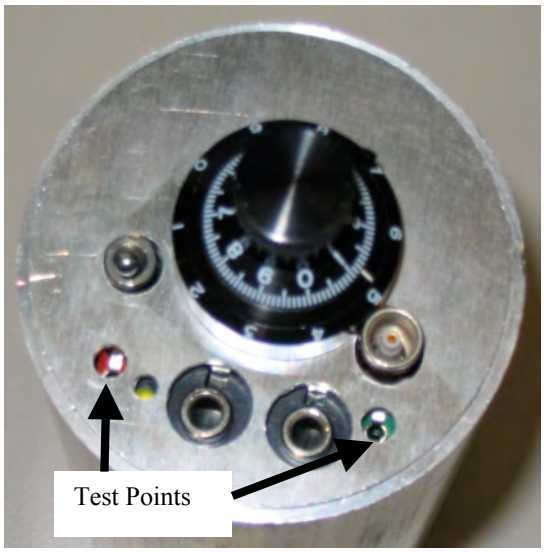



*Figure12. The end of a PMT base showing test points that may be probed with a low voltage DC voltmeter to measure PMT operating voltage. A multiplication factor of 385 is used to determine actual tube voltage.*

Lemo cables, which are thin coaxial cables with "lemo" connectors on the ends, carry the PMT signal to the DAQ board. A toggle switch on the end of the base allows you to turn on/off the tube without disconnecting the power cable. An amber LED near the test points tells when the base is "on."

## Cables for Connecting to PC

The DAQ board connects to your PC's serial port with a standard serial cable. If your PC does not have a standard RS-232 serial port, you can use a USB-to-serial converter, available commercially.

# Chapter 3: Data Display

## Data Display on a PC

View the data with a terminal emulation program such as Zterm (Mac or Windows), Hyperterminal (Windows) or a "homemade" Unix-based program you can download from the QuarkNet website. See Appendix C for details on setting up these terminal emulation programs.

## Keyboard Commands

Once the terminal emulator is initialized and the scintillation counters and board are powered, you should begin to see data on the screen that looks something like that shown in Figure 13.

The meaning of these "words" will be discussed in great detail in Chapter 6. Note that a single event may result in more than one line of data! In this section, we discuss the meaning of some of the keyboard commands that you can type in order to modify/read various values to/from the board that are not displayed on the screen until requested.

Your PC is capable of sending commands *to* the DAQ board in addition to receiving the datastream *from* the board. Figure 14 shows an example of three keyboard commands that you type in order to measure the temperature at the GPS sensor, the value of the scalers, and to display GPS data.

The "HE" (help) command sends a summary of all keyboard commands. Figure 15 shows this command summary.

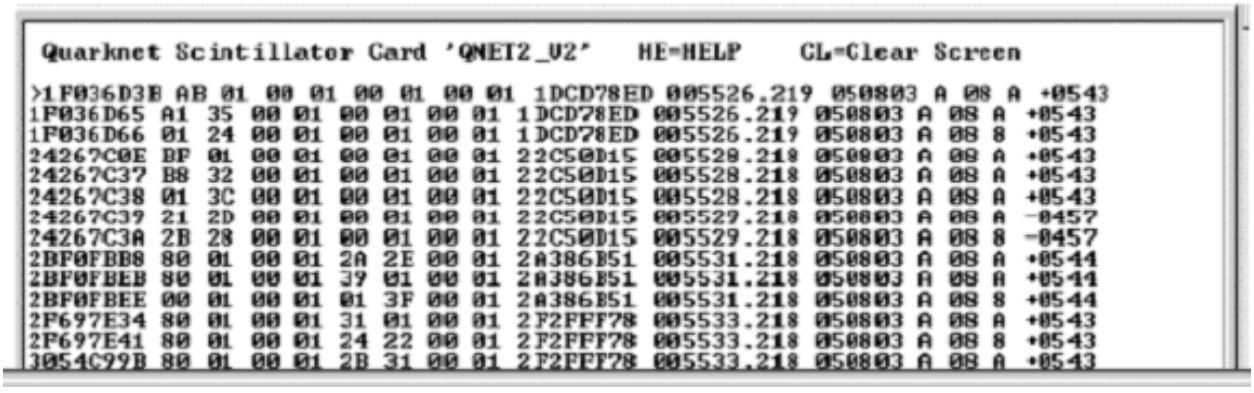



*Figure 130. Sample screen of datastream from DAQ board when viewed with a terminal emulation program.*

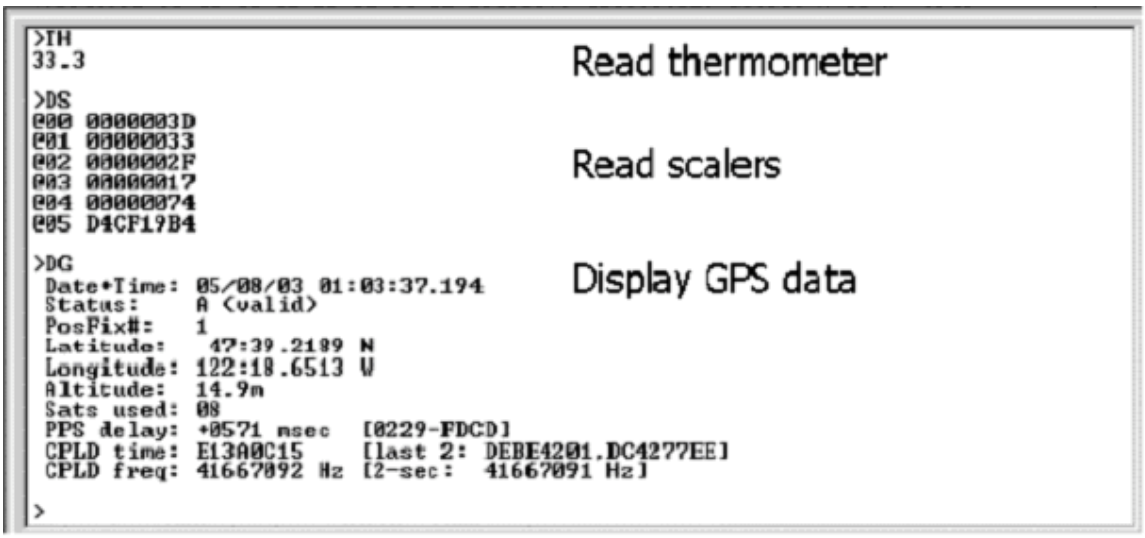



*Figure14. Example of keyboard commands that may be sent to the DAQ board with the appropriate board responses.*

```
Scintillator Card, QNET2  Firmware Ver2.3, 09/09/03   HE=Help
Serial#=1002   uC_Volts=3.3   uC_TempC=26.6   GPS_TempC=235.0   kPa=0

Barometer - BA=Display, BA bb.b gg.g calibrates kPa Baseline, Gain (See HB).
Counter   - CE=Enable, CD=Disable, Controls TMC Running bit @ CPLD CCR1.
DC        - Display Counters and Control Registers of CPLD, address 0-4.
DF        - Display Scalar Fifo Data (first 12 Bytes as three 32bit numbers).
DG        - Display GPS Date, Time, Position and Status.
DS        - Display Scalar Fifo, Counters 0-3, Triggers, and 1_PPS Time.
DT        - Display Time Control Registers of TMC, address 0-3.
Flash     - FL=Load Binary File, FR=Read SumCheck, FC=Copy_to_CPLD.
GP        - Init Link with GPS unit (GGA=1/sec, RMC=1/sec, disable others).
Help      - HF=Trigger format, HS=Status format, HB=Barometer format.
NA n      - NMEA GPS Data Append (n==1 On),(n!=1 Off), add GPS to output.
NM n      - NMEA GPS Data Echo (n==1 On),(n!=1 Off), (GPS_Baud=9600).
Reset     - RB=TMC+CPLD,  RE=MSP430+TMC+CPLD.
SA n      - SA=Save TMC+CPLD Registers to Flash, (SA 1=Restore Defaults).
SB n      - Set Baud Rate (PC Link), 1=19200, 2=38400, 3=57600, 4=115200.
SN nnnn   - Serial Number(BCD), SN=Display Number, SN nnnn=Store Serial Number.
ST n      - Send Status Data (n==1 On),(n!=1 Off), (See HF).
TH        - Thermometer Data Display, -40 to 99 degrees C.
WC mm nn  - Write Counter Control Registers CPLD address mm with data nn.
WT mm nn  - Write Time Control Registers TMC address mm with data nn.
```

*Figure15. A list of all keyboard commands viewable by typing "HE."*

Some of the most commonly used commands are discussed in Table 1. See Chapters 4 and 6 for more detailed descriptions of how and when to use these commands.

| *Keyboard Command* | *Description* |
|---|---|
| CD | **Counters Disabled.** Does not write event lines on the PC monitor even though the scalers will still increment. This is useful when you do not want to "see" all the data coming in. |
| CE | **Counters Enabled.** Writes event lines on the PC monitor. This command has the opposite effect of the CD command. |
| DS | **Display Scalers.** Displays values of the scalers in hexadecimal. Scalers 0-3 store the singles rates for channels 0-3, and counter 4 records the number of n-fold coincidences, where "n" is whatever level you set. These counts are running totals since the last time the counters were zeroed. |
| RB | **Reset Board.** Resets scalers and TMC only. |
| RE | **Reset Everything.** Resets everything (including card setup parameters that you may have modified); will zero all scalers. |
| TH | **Thermometer.** Reports reading of temperature sensor on the GPS module connector, about 3 feet from the GPS module itself. |
| GP | **GPS Data.** Displays current date, time, GPS position, and number of satellites visible. |

*Table 1. Some of the most common keyboard commands with descriptions.*

# Chapter 4: The DAQ Card

## GPS Observation

To read out the current GPS time, latitude, longitude, altitude and number of satellites in view, type "GP" as shown in the previous section. You can, however, directly monitor the GPS module's serial datastream by sending the command "NM 1." This command will pass *all* data from the GPS module directly to your PC. Normally, the board's MCU reads, interprets and distills the raw GPS information to produce the DG command output. Although this will continue to be true even after you send the "NM 1" command, you will get many more lines of detailed data (in the GPS industry's standard NMEA format). Refer to the link to understand this data format.
<http://www.phys.washington.edu/~berns/archive/Leadtek/GPS_Protocol_ReferenceManual.pdf>

To disable NMEA data-forwarding and return to normal operation, send the command NM (i.e., no "1").

If you connect the GPS module directly to your PC's serial port, as discussed before, you can also use a software package provided by LeadTek to display sky maps of satellites and other interesting data. This can be found online.
<http://www.phys.washington.edu/~berns/archive/Leadtek/GMonitor.exe>

## Singles and Coincidence Rate Measurement

Important measurements you want to make are the rate at which an individual counter is hit ("singles rate") as well as the rate at which a 2- to 4-fold coincidence occurs ("coincidence rate"). To determine these, type the following commands shown in Table 2:

| | |
|---|---|
| CD | **Counter Disabled.** Prevents the datastream from off your screen. |
| RB | **Reset Scalers.** But immediately after type . . . |
| DG | **Display GPS Data.** Gives the initial time—when you began counting. (You could use a stopwatch to measure the elapsed time instead.) |
| DS | **Display Scaler.** After an adequate amount of time has passed (say 5 minutes or so) take the final reading of the scalers and then immediate type . . . |
| DG | **Display GPS Data.** Give the final time. |

***Table 2. The keyboard command sequence performed for determining counting rate.***

The counting rates for any particular scaler (0-3) can be determined by dividing the hex valued scaler by the difference of the beginning and ending times. The number of n-fold coincidences that occurred within the elapsed time is recorded in scaler 4. Dividing the number of coincidences by the elapsed time will yield the coincidence rate. Figure 16 illustrates this command sequence and the corresponding board outputs.

As an example of the singles rate, consider the data in which the counting rate for channel 0 was $BCF_H = 3023_{10}$. The elapsed run time was 21:35:45 – 21:30:46 = 4:59 or 4.98 minutes. Thus, the singles rate is

$$R_{singles} = \frac{3023 counts}{4.98 \min} = 607 \frac{counts}{\min}$$

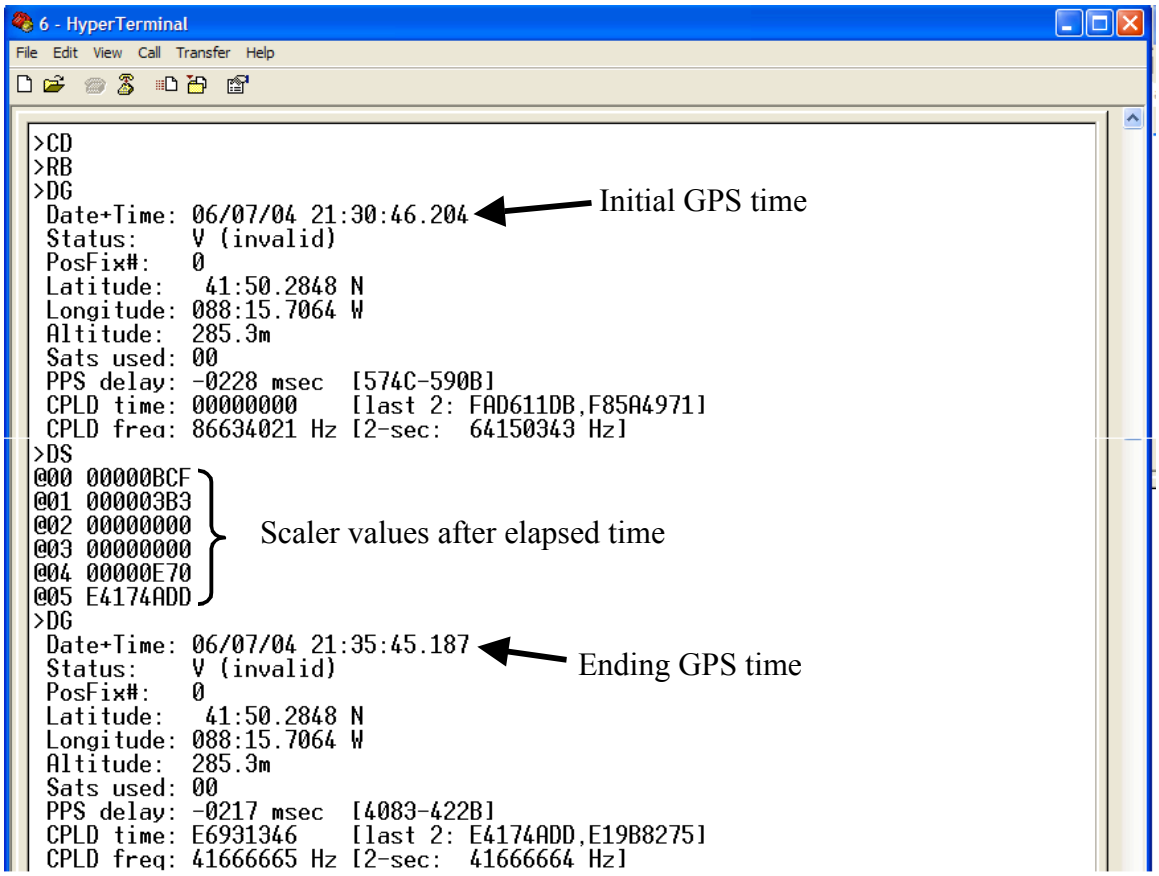



*Figure16. An example of keyboard commands and board responses used to determine the singles counting rate in a particular channel.*

## Data Collection

Once you have setup and tested your equipment and have plateaued the counters, you are ready to collect and save data for a run. Although the commands for starting and stopping a run vary slightly with each terminal emulation program (see Appendix C for these commands), the idea is the same: Once the datastream appears on your monitor, select the command that will allow you to begin capturing text. You will be prompted to type a name and a folder into which these data will be saved. Your emulator will save these data as a text file that can be analyzed later. The data will be written to this file continuously. For example, opening this text

file while data is being saved will display all the data collected up to that moment.
When you want to stop a run (which could be minutes, hours or even weeks later!), use the stop command for your particular emulator. (See Appendix C for this command.) This file can be uploaded to the server and the data analyzed.

# Chapter 5: Data Upload and Analysis

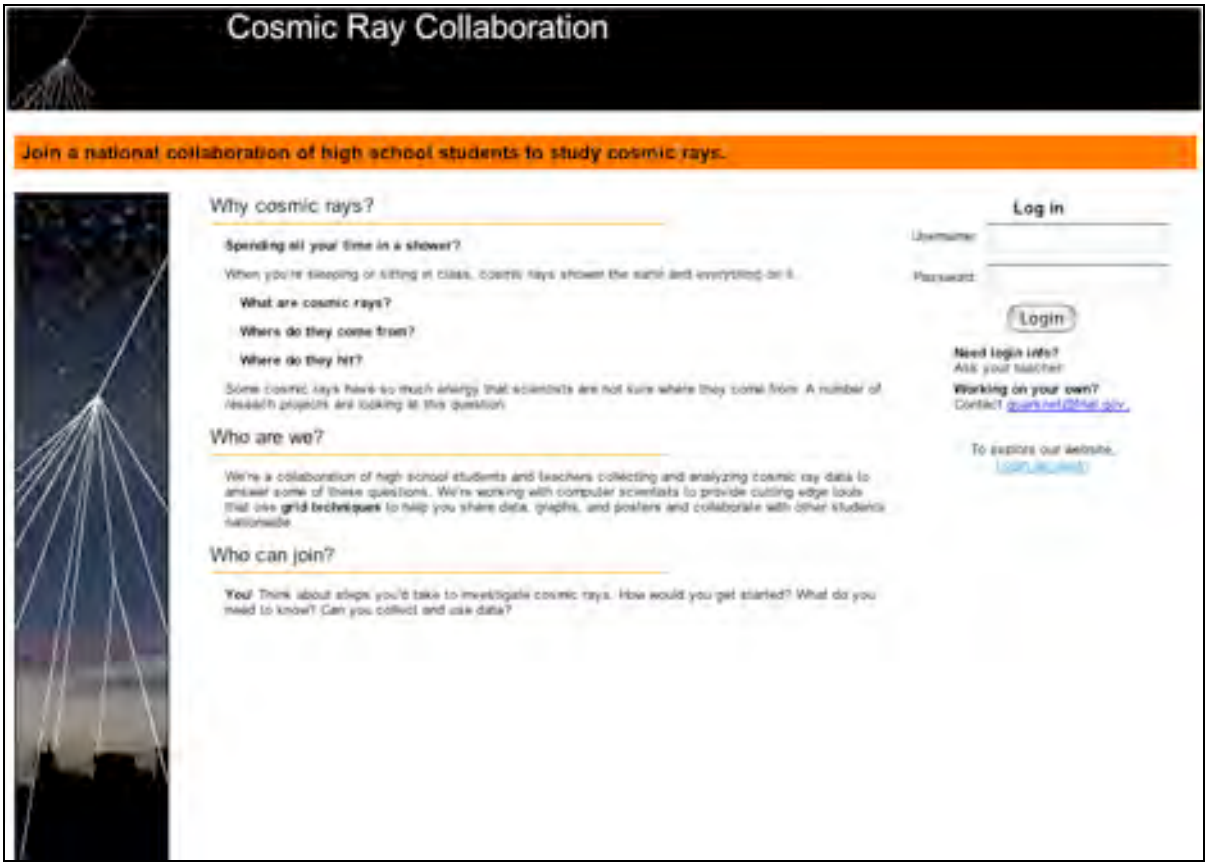



*Figure 17. The QuarkNet Cosmic Ray Collaboration webpage serves as the studenthomepage for the cosmic ray project.*

## QuarkNet Website Overview

### Teacher Pages

The QuarkNet teacher activities database is found at <http://eddata.fnal.gov/lasso/quarknet_g_activities/view.lasso>.

By selecting the "Grid Cosmic Ray Collaboration" link, you can find out more about this project and use online tools for implementing this project in your classroom. You will find a page to register their school as part of the cosmic ray investigations network so that your students may upload data from their detector. A link to the student homepage is also located here.

### Student Pages

The student homepage for this project is <http://quarknet.fnal.gov/grid> (Figure 17).

Among many resources, students will find:

- **Studies** – Interesting questions along with ideas to answer these questions through experiment.
- **Resources** – Links to related projects, physicists and other student group contacts.
- **Upload Data** – Links to upload data and geometry files.

- **Data Analysis** – Powerful analysis tools to analyze data for a variety of experiments. *Even students without a detector can analyze data.*
- **Poster Session** – A method for students to share their results, search the work of other students and collaborate with students at other schools.

## Data Upload to the Server

While many of the cosmic ray detector experiments that your students can do require only the equipment at your local site, analyzing the data is cumbersome with software such as Excel. Working locally discourages student collaboration like that found in real high-energy experiments. Accordingly, QuarkNet is developing a website that allows data uploads and analysis. This simplifies the analysis of these complex data and supports student collaboration.

Before students can upload data to the server, you will need to register your detector as part of the cosmic ray investigations network. This is easily done on the teacher website. Registration will require you to enter the serial number of the detector found on the bottom of the DAQ card itself, or by typing the keyboard command "SN" in the terminal emulator.

By selecting the "Upload" link from the <http://quarknet.fnal.gov/grid> page, you will seee instructions and links for uploading data. (You do not need to worry about uploading data that may be "contaminated" by keyboard command lines typed during the operation of this run as the analysis software will create a new file for this data using only data lines that are the complete 16 words across.)

Once you have uploaded data, to see that it has been loaded successfully select the "Data" link. You can "View" your data in your school's folder. The link to view this data lists the date the data was collected. If a data run spans more than one day, the data will be broken into separate files labeled with the appropriate date.

## Analysis Tools

Once your students have collected and uploaded the data, there are four studies they can do. Analyze data from your detector as well any other detector that has uploaded data. You can even look for coincidences between your school and other schools if run times overlap! This section will explain the nature of each study/analysis and provide a framework from which you can get started.

### System Performance

The main objective of the "system performance" study is to understand the quality of the data that you have collected or want to analyze.

The primary technique used to evaluate data in the performance study is a histogram of the number of events as a function of time over threshold (TOT, see Chapter 2). In a very loose sense, TOT is a measure of the amount of energy deposited in the scintillator for a given event. Ideally, the DAQ board would have been equipped with an ADC (Analog-to-Digital Converter) to measure pulse height and thus the energy deposited in the scintillator. To keep the cost down, however, a TDC (Time to Digital Converter) was used. (See Chapter 2 for more on the TDC.). Since taller pulses (pulses with more energy) are generally wider, there is a correlation between the time that the pulse exceeded the threshold value and the amount of energy deposited in the scintillator.

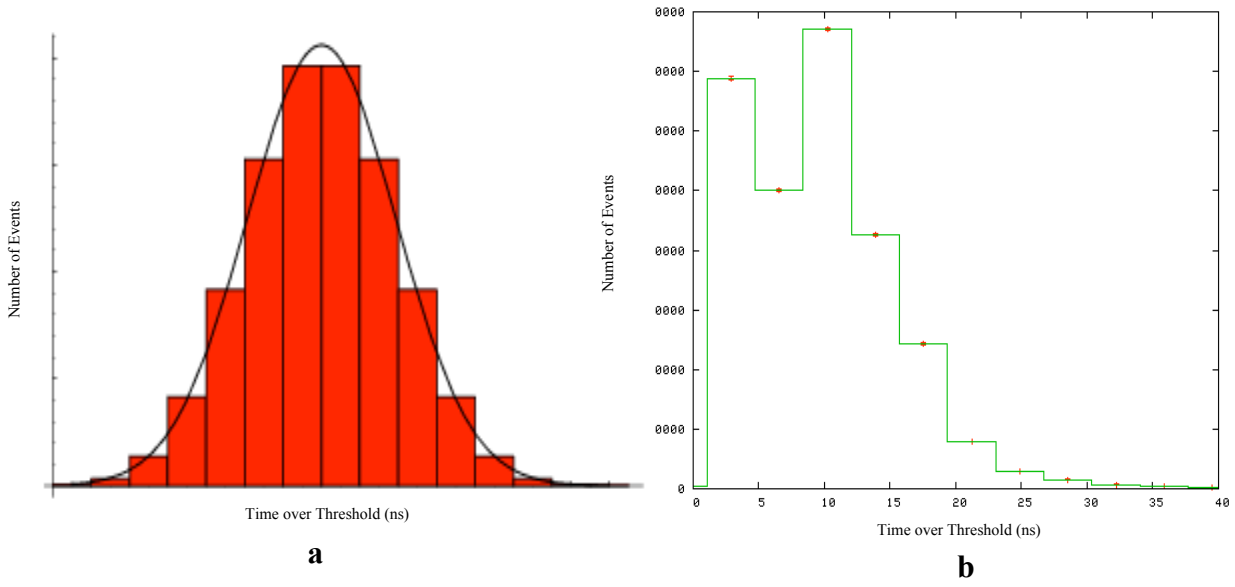



*Figure 18. (a) The Gaussian distribution expected from an 'ideal' system performance graph, (b) an actual system performance grph.*

Figure 18(a)[5] shows the ideal, Gaussian-shaped distribution about a mean value for the time over threshold; Figure 18(b) shows an actual data run plot.

The actual data run histogram peak does occur around 12 ns (which is typical for the 1/2" thick scintillator, the PMT efficiency, and threshold settings used in our experiment). This actual data, however, appears more skewed than Gaussian. This histogram may tell us that there is "good" data in our actual run data, but that it is contaminated with short pulses (a.k.a. noise) that were just above threshold. Increasing the threshold value may "clean-up" this channel in the future.

There is probably a large portion of the actual run data in Figure 18(b) that represents actual muons traversing the counter. No real data histogram will ever look perfectly Gaussian, but a shape somewhat resembling that of Figure 18(a) can be a reasonable indicator that much of your data is "good" and represents real muons.

One factor you can modify in plotting a histogram is the number of bins into which the data are divided. Choosing a different number of bins for the same data set can significantly change the shape of the graph and may help evaluate the quality of the data.

## Flux Experiments

Cosmic ray flux, $\Phi_{CR}$, can be defined as

$$\Phi_{CR} = \frac{events}{(time)(area)}.$$

Since the area of your detector will probably not be varied in during an experiment, flux studies ask the question, "How does the rate at which cosmic rays pass through my detector depend on . . . ?" Since there are many ways to complete this question, this study is

[5] Figure is used with permission. Eric W. Weisstein. "Gaussian Distribution." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/GaussianDistribution.html

a fairly straightforward yet excellent open-ended research question for your students.

Flux experiments may include several geometries of your detectors such as (1) a single counter, (2) multiple counters placed in the same plane to increase detector area, (3) closely stacked counters that require coincidence, and (4) separated, stacked counters that require coincidence used to look at flux from only a particular direction of the sky. There are several interesting variables to test in flux studies. Have your students generate the questions that *they* want to study.

In analyzing flux study data, students select the data set and channel(s) to be analyzed. The analysis tools produce a histogram of flux vs. time similar to the one shown in Figure 19. Students can juxtapose data regarding their independent variable as a function of time to look for correlations.

## Muon Lifetime Experiments

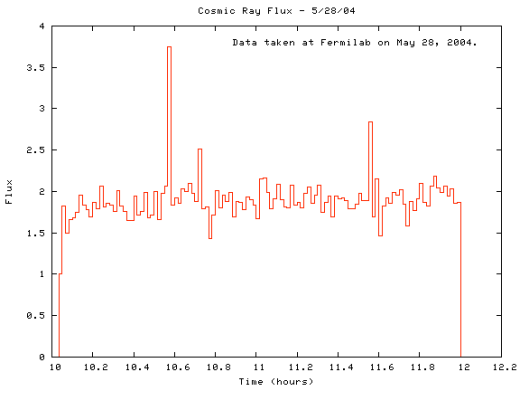



*Figure19. Cosmic ray flux (events/detector area/sec) vs. time graph over a period of two hours.*

A classic modern physics experiment is the measurement of the muon mean lifetime. This experiment has historical significance as the very first experimental evidence of time dilation. Chapter 1 discusses how the lifetime measurement can be used to verify time dilation. Here we develop the muon lifetime measurement itself.

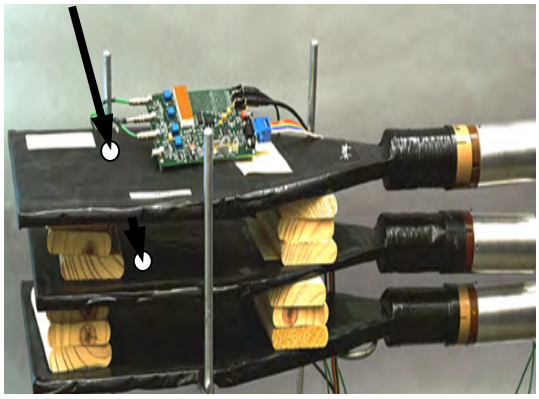


*Figure20. A three counter muon lifetime detector system.*

Consider Figure 20. When a muon enters a counter, a signal is generated in that counter. If the muon traveled through counter 1 and into counter 2, both of these counters should have a signal at approximately the same time. If the muon has stopped within counter 2, however, it will "wait around" until it decays and generate a second signal in counter 2 upon decay. The time $t_{DECAY}$, the time between the two signals in counter 2, can be extrapolated from the data file. Since no signal occurred in counter 3, this is a strong indication that a muon did indeed decay in counter 2.

Theory suggests that the muon lifetime exhibits the characteristic exponential decay that is common to other physical phenomena, such as radioactive decay. In the case of radioactive materials, the decay expression is given by

$$N(t) = N_0 e^{-\frac{t}{\tau}}$$

where $N_0$ is the initial population, N(t) is the population after time t, and $\tau$ is the mean lifetime. The fact that muons in our experiment do not coexist, whereas they do in radioactive materials, does not change the decay expression. This does, however, give a slightly different meaning to N(t) and $N_0$. In this experiment, $N_0$ is not the number of muons present at t = 0 but represents the total number of muons that were captured and decayed within the detector. N(t) is then the number of these muons with $t_{DECAY} > t$.

It should also be mentioned here that the muon's time of flight *before* it entered the detector is of no concern. Though $t_{DECAY}$ is not its entire lifetime, plotting the number of decays as a function of decay time will yield the same exponential decay shape from which lifetimes can be found. The common "zero time" that we use here is when each muon entered the detector. If a large number of decays are collected in the manner described above, a plot can be produced using the web analysis tool similar to the one shown in Figure 21.

While doing a lifetime study, no coincidence level should be set locally. Instead, coincidence levels may be requested in the web analysis. Students select the data file to be analyzed, the coincidence level, and the time interval between which coincidences must have occurred. The lifetime can be determined by fitting this decay plot with an exponential decay function of the form described above. It is also possible for students to estimate the background value that should be subtracted from each bin before fitting. This is also shown in Figure 21.

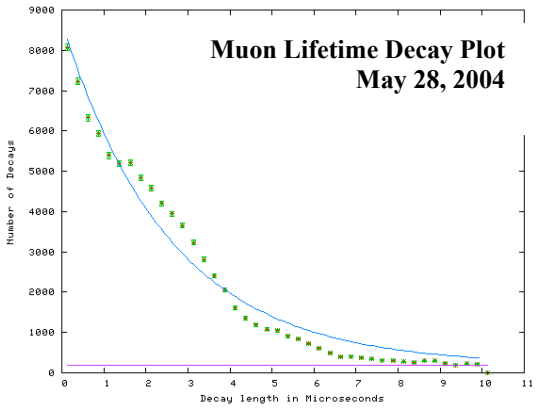



*Figure21. A lifetime graph showing number of number of decays as a function decay from when the muon entered the scintillator. Fitting this data to an exponential decay function can yield the lifetime.*

## Shower Studies

With the absolute time stamp that GPS offers, a network of detectors (at the same site, or at different schools) can study cosmic ray showers. The web analysis tools allow students to make predictions about which direction in the sky the shower (and thus the primary cosmic ray) came from.

Students select the specific data sets to analyze (these may be from different locations), the coincidence level required, and the time interval between which coincidences must have occurred. A 3-D plot of position on the earth (xy) is graphed as a function of signal time (z). This graph allows them to 'see' the direction from which the shower came.

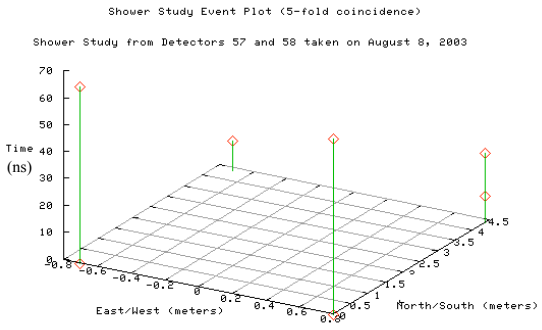



*Figure22. Shower study plot showing coincidences within a 70 ns time window.*

# Chapter 6: Advanced Topics

## Plateauing Counters

In order to determine the operating voltage of each of your counters, it is necessary to perform a simple procedure in which you "plateau the counter." This step is necessary because counting variations due to drifts in the tube gain or voltage can occur during an experiment. By using an operating voltage near the center of this plateau, these effects are minimized.

## How to Plateau a counter

There are two methods to plateau a counter:
1) Plateauing using a single counter
2) Plateauing using multiple counters
The first technique is simpler, however, it may make "seeing" the plateau more challenging.

### Plateauing Using a Single Counter

To plateau a single counter, choose a threshold value for the discriminator and hold this constant for your experiment. (Chapter 2 discusses how to choose a threshold value.) Turn on the DAQ board and connect it to your computer so that you can view the datastream on your monitor using a terminal emulation program. (See Appendix C for help setting up a terminal emulation program.) Once you see the datastream, convince yourself that lower potentiometer dial values result in a lower counting rate while higher settings produce a very high datastream rate on your monitor.

Noting the potentiometer reading on the bottom of the PMT base, perform a simple experiment to determine the singles rate for the channel that the counter is connected to using the method described on page 19. Continue the process for many different voltages (potentiometer settings) to produce a graph similar to the one shown in Figure 23. Remember that the scalers increment in hex format. Also, note that the channels are numbered 1, 2, 3, and 4 on the board, numbered 0, 1, 2, and 3 respectively on the monitor. This numbering scheme (starting from 0 instead of 1) is consistent with engineering and computer programming practice.

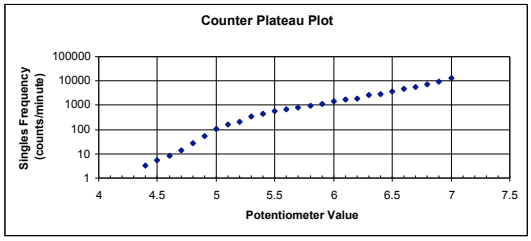



*Figure23. Example of counter plateau graph used to determine operating voltage of PMT.*

The optimum operating voltage is where the semi-log graph looks the most horizontal. For the graph shown, this occurs around 5.5 V. Sometimes it is difficult to "see" the plateau with this technique. If multiple counters are available, the technique outlined below make the plateau more apparent.

**Plateauing Using Multiple Counters**
Consider the three-counter setup in Figure 24. Let's plateau counter 2. (This can occur even if counters 1 and 3 have not been plateaued.) You can setup an experiment to investigate the coincidence frequency between counters 1 & 2, 2 & 3, and 1 & 3 as a function of the potentiometer setting on the base of counter 2. (You may need to collect data for these three coincidence sets during separate runs unless you have a splitter for the signal cables and a second DAQ board.)

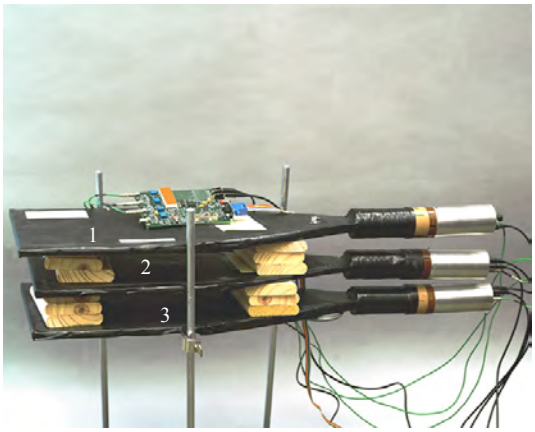



*Figure24. Shown here is a three-counter setup used in plateauing a counter in order to determine the optimum operating tube voltage.*

Since coincidences between 1 & 3 should be relatively constant over the experiment, this can serve as a baseline from which to compare the 1 & 2 and/or 2 & 3 coincidences. A plot of $\frac{1\&2}{1\&3}$ vs. counter 2's potentiometer reading or a plot of $\frac{2\&3}{1\&3}$ vs. counter 2's potentiometer reading produces a plateau plot similar to that shown in Figure 25. You can see that this plateau region is somewhat "flatter" than was the case with the single counter technique discussed above.

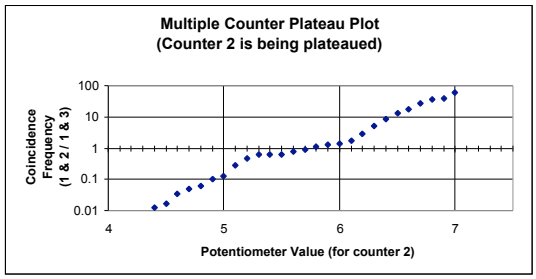



*Figure 25. Example of multiple counter plateau graph used to determine operating voltage of the PMT for counter 2.*

Figure 25 shows that the "flattest" part of the graph is near the potentiometer setting of 5.5. This value will serve as a reasonable value at which to operate this counter.

## Data Words

This section explains the meaning of each of the data "words" that appear as a result of a single event. It is not imperative for you to understand these words in order to collect and analyze data, but understanding these words can help as you attempt to evaluate the quality of your data.

The datastream from the DAQ board is in ASCII format. Each data line contains 16 words as shown are a sample set of data on the nest page in Figure 26. Words 1-9 are in hex format. The data shown below is for a single event—*a single event can span several lines of data!*

```
80EE0049 80 01 00 01 38 01 3C 01 7EB7491F 202133.242 080803 A 04 2 -0389
80EE004A 24 3D 25 01 00 01 00 01 7EB7491F 202133.242 080803 A 04 2 -0389
80EE004B 21 01 00 23 00 01 00 01 7EB7491F 202133.242 080803 A 04 2 -0389
80EE004C 01 2A 00 01 00 01 00 01 7EB7491F 202133.242 080803 A 04 2 -0389
80EE004D 00 01 00 01 00 39 32 2F 81331170 202133.242 080803 A 04 2 +0610
```

*Figure26. A sample event containing five lines of data. Each line is comprised of the same 16-word format with each word separated by a space.*

Below is a brief description of each of the "words."

***Word 1:*** A 32-bit trigger count of the 41.66 MHz CPLD clock mounted on the DAQ board with a range from 00000000...FFFFFFFF. Its resolution (1 LSB increment) is 24 ns.[6,7] A trigger count of 00000000 means that the DAQ card is still in the initialization phase, i.e., the GPS receiver has not started to generate 1PPS pulses yet. Do not use the initial data until the trigger count becomes non-zero!

***Word 2:*** TMC count of the **R**ising **E**dge at input **0** (RE0). It is also the trigger tag. The format used is shown below:
*bits 0-4 = TMC count of rising edge, resolution = 0.75 ns (=24 ns/32)*
*bit 5 = channel edge tag (1 = valid rising edge, 0 = no rising edge)*
*bit 6 = not used, always 0*
*bit 7 = trigger tag (1 = new trigger, start of a new event; 0 = follow-up data of a trigger event)*

***Word 3:*** TMC count of **F**alling **E**dge at input ***0*** (FE0). The format used is shown below:
*bits 0-4 = TMC count of falling edge*
*bit 5 = channel edge tag (1 = valid falling edge, 0 = no falling edge)*
*bits 6-7 = not used, always 0.*

***Word 4:*** TMC count of rising edge at input 1 (RE1); same format as RE0, except bit 7 is always 0.

***Word 5:*** TMC count of falling edge at input 1 (FE1); same format as FE0.

***Word 6:*** TMC count of rising edge at input 2 (RE2); same format as RE1.

***Word 7:*** TMC count of falling edge at input 2 (FE2); same format as FE1.

***Word 8:*** TMC count of rising edge at input 3 (RE3); same format as RE1.

***Word 9:*** TMC count of falling edge at input 3 (FE3); same format as FE1.

***Word 10:*** A 32-bit CPLD count of the most recent 1PPS (1 pulse per second) time mark from the GPS receiver. This hex word ranges from 00000000...FFFFFFFF and has a resolution of 24 ns just like word 1.[8]

[6] 24.00 ns if the CPLD clock is exactly 41666666.67 Hz. The actual CPLD frequency of each card is usually tuned within ±30 Hz of the target frequency of 41666667 Hz. Generally, an Hz drift from the target CPLD frequency result in accuracy errors of up to n*24 ns. For example, a 30 Hz drift means that the accuracy time error has a range of ±720 ns if exactly 24.00 ns are assumed for a CPLD clock tick time. It is reasonable to assume that an error within ±1000 ns is acceptable for a school network if the schools are more than 1 mile apart from each other.

[7] The CPLD clock frequency fluctuates slightly over time, depending on temperature changes and oscillator ageing drifts. Therefore, in order to achieve high accuracy (±50 ns) in computing the absolute trigger times, you need to poll the current CPLD frequency at a regular basis (say once every 5 minutes) with command DG (Display GPS data). If the event rate is high enough (at least 1 event per 100 seconds), the CPLD frequency can be computed from the 1PPS counter numbers of consecutive events.

[8] The same uncertainty comments apply here as in word 1.

***Word 11:*** UTC time of most recent GPS receiver data update. Although one update is sent each second, it is asynchronous with the 1PPS pulse. The format used is shown below:

*HHMMSS.mmm*
*where: HH = hour [00...23]*
*MM = minute [00...59]*
*SS = second [00...59]*
*mmm = millisecond [000...999]*

***Word 12:*** UTC date of most recent GPS receiver data update. The format used is shown below:

*ddmmyy*
*where: dd = day of month [01...31]*
*mm = month [01...12]*
*yy = year [00...99]*
*e.g. 03=2003]*

***Word 13:*** A GPS valid/invalid flag.
*A = valid (GPS data OK),*
*V = invalid (insufficient satellite lock for 3-D positioning, or GPS receiver is in initializing phase); time data might be OK if number of GPS satellites is 3 or more and previous GPS status was "A" (valid) within the last minute.*

***Word 14:*** The number of GPS satellites visible for time and position information. This is a decimal number between 00…12.

***Word 15:*** This hex word is a DAQ status flag. The format used is shown below:

*bit 0: 0 = OK*
*1 = 1PPS interrupt pending (Warning flag: If DAQ card is busy, then 1PPS count might lag behind or get mismatched.)*
*bit 1: 0 = OK*
*1 = trigger interrupt pending (possibly high trigger rate; if continues, then data might be corrupted.)*
*bit 2: 0 = OK*
*1 = GPS data possibly corrupted while DAQ uC was/is busy.*
*bit 3: 0 = OK*
*1 = Current or last 1PPS rate is not within 41666666 ±50 CPLD clock tick.s (This is a result of a GPS glitch, the DAQ uC being busy, or the CPLD oscillator not tuned correctly.)*

***Word 16:*** The time delay in milliseconds between the 1PPS pulse and the GPS data interrupt. A positive number means 1PPS pulse is ahead of GPS data, and negative number means GPS data is ahead of 1PPS. To get the actual GPS time to the nearest second, round (word 11 + word 16/1000) to nearest full second. This gives the actual GPS time at the last 1PPS pulse.

See Appendix F for the interpretation of the data for the previous example.

# Coincidence Counting Variations

## Setting with Keyboard Commands

Chapter 4 discusses how to use some keyboard commands to read scaler values over a known period of time in an effort to determine basic counting rate. That simple calculation does not discuss two parameters—gate window and TMC delay (*d* and *w* respectively). The gate window, *d*, refers to how close in time pulses must be to cause a trigger. The TMC delay, *w*, more difficult to define, determines which pulse edges get read out into the datastream when a trigger occurs, by delaying this information until the trigger actually happens. (See Appendix E for a full explanation of each quantity and the details of time coincidence and data handling.)

Consider the actual physical quantities and relate them to the two parameters, *d* and *w*.

- **$T_{TRG}$:** This quantity is the maximum time window during which real physics processes might cause a trigger. In other words, if you set the card for 2-fold coincidences, and one channel receives a signal at t = 0, at least one other channel must show a signal before t = $T_{TRG}$ or you will not receive any information about that pair of pulses. The examples in Appendix E may help.
- **$T_{WIDTH}$:** This quantity is the duration after the first pulse in a trigger during which you want to record all rising and falling edge times. The channel in which these extra pulses occur does not matter; all of them will be saved.

The physical quantities $T_{TRG}$ and $T_{WIDTH}$ are the values for *d* and *w* you set on the board. You set $T_{TRG}$ and $T_{WIDTH}$ based on the type of experiment that you want to perform. Below is a list of several types of experiments and typical parameter values for each.

**Muon Telescope Flux Experiments**
**Set *d* = 2, and *w* = 401.**
If you want to build a muon telescope with counters less than a meter apart, $T_{TRG}$ is the time it takes a muon to go from the top counter to the bottom counter. Since these muons travel at essentially the speed of light (≈ 1 ft/ns), the $T_{TRG}$ is only a few ns so you set *d* to the minimum value of 2 clock ticks, or 48 ns. Similarly, to simply count muons, you are only interested in a short time window for edges, so you could take $T_{WIDTH}$ = 48 ns as well.

**Muon Lifetime Measurements**
**Set *d* = 2, and *w* = 401.**
If you want to look for muons that stop and decay (2.2 μs mean lifetime) in a counter, you look for the decay electron pulses several microseconds after the trigger, so you have $T_{WIDTH}$ = 400 (9600 ns, or about 10 μs).

**Cosmic Ray Showers**
**Set *d* = 50 (= 1.2 μs) and *w* = 100.**
In extensive air showers, the main "pancake" of arriving particles should be between 10 m and 200 m thick. The earliest and latest particles might arrive at the counters about 0.6 μs apart, for a vertical shower. Thus you want $T_{TRG}$ to e about 1 μs. $T_{WIDTH}$ can be about the same.

However, you should *not* set huge time windows "just to be safe." Doing this causes the card to be busy all the time, and its effective dead time will become a serious problem. The values of *d* and *w* should be as small as possible for the given physics goals.

Default values, effective when the card is powered up, are ***d* = 6 (= 144 ns) and *w* = 10 (= 240 ns).** Modifying these values is probably unnecessary for muon telescope experiments. However, for air shower detectors or muon lifetime experiments, you should set them at larger values. These values serve as a good guide. If you want to use the values of *d* = 50 and *w* = 100, Table 3 shows how to set the DAQ card to these values:

| | |
|---|---|
| RE | **Resets the card** by setting all counters to zero and return everything to defaults |
| CD | **Disables the counters and TMC** |

| | |
|---|---|
| WT 01 00<br>WT 02 32 | **Sets the time register** with read pointer (register 01) = 00 and the write pointer (register 02) to *d,* in hexadecimal. The TMC delay *d* will be the difference of these values (so in fact you could set 01 to some value *nn*, and 02 to *nn* + *d*). This example sets *d* = 50. We have set register 02 to $32_H = 50_{10}$. The maximum is $7F_H = 127_{10}$. |
| WC 00 1F | **Sets the coincidence level and enable desired channels**. Set control registers with the WC (Write Control register) command. Register 0 sets the coincidence level, using its first (left) hex character. *Here, 0 means singles, 1 means 2-fold, and n-1 means n-fold coincidence level is to be required. This is different from how coincidence is described in the data analysis software found online.* The second (right) hex character represents the desired pattern of enabled/disabled channels in its bits, with 0s for disabled channels and 1s for enabled channels. In this example, 1F means set 2-fold coincidences with all 4 channels enabled: $F_H$ = 1111. If you wanted to disable channel 2 you would use 1101 = $D_H$. If you wanted to enable three-fold coincidences with channel 2 turned off, you would use WC 00 2D. |
| WC 02 64<br>WC 03 00 | **Sets the value of *w* (gate width).** Registers 2 and 3 hold the gate width setting in integer clock ticks (units of 24 ns). The lower 8 bits of the binary number representing *w* go in register 2, and the higher 8 bits in 3. Thus, the desired gate width *w* is given as *abcd* and is typed as<br>WC 02 cd<br>WC 03 ab<br>As an example, to get *w* = 100 clock ticks ($100_{10} = 0064_H$ = 0000 0000 0110 $0100_2$) you would type what is shown to the left. |
| CE | **Re-enables the onboard counters** |
| DS | **Displays the scaler values** |

*Table 3. A keyboard command sequence that allows for coincidence counting variations of the gate window and TMC delay.*

## Interpreting Coincidence Data from the DAQ Board

This example shows how to interpret the coincidence data from the board. Assume you are looking for 2-fold coincidence between any two of the inputs. As discussed in the previous section, "WC 00 1F" will set up this 2-fold coincidence. If, after some time has past, you type "DS," the card will return the scaler values for each channel as discussed previously. In addition, the '@04' line returns the value of the 2-fold coincidences. This number is also in hexadecimal. The screen capture on the next page (Figure 27) illustrates these commands and outputs.

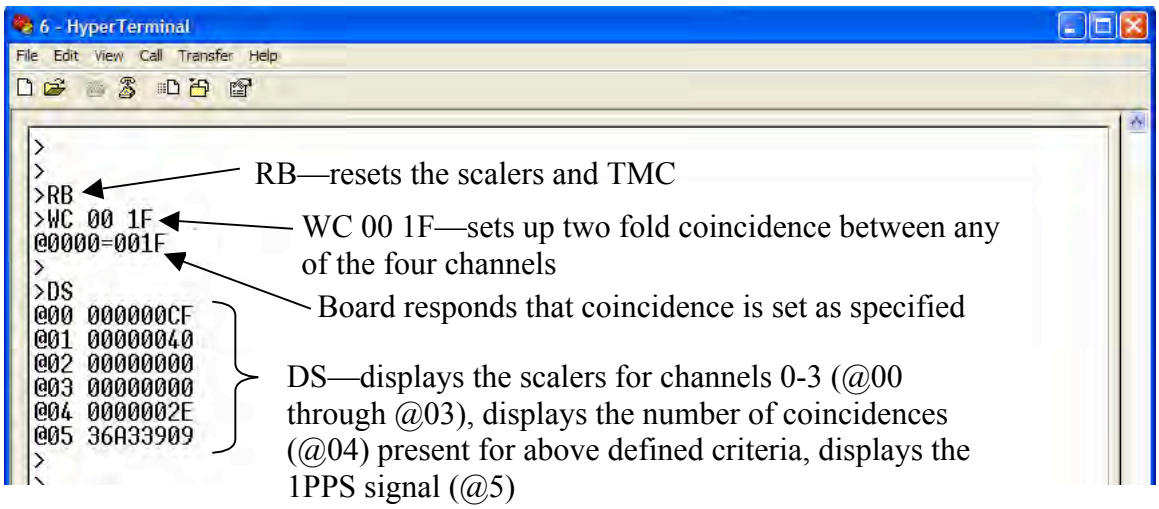



*Figure27. A sample terminal emulation window displaying how to set coincidence levels and how to read this coincidence data.*

## On-board Barometer Calibration

You can measure air pressure with a barometer on the DAQ board. The barometer sensor must be calibrated before it can yield pressure in units of kPa. To calibrate the sensor, you will need a real barometer (or some way to retrieve actual pressure in the local area). The actual barometer reading should be recorded every time the command "BA" is typed on the keyboard. The 'BA' command displays a number that represents the ADC (analog-to-digital converter) count readout of the on-board barometer. Figure 28 illustrates this command. Notice that the value changes each time the "BA" command is typed due to fluctuations in local air pressure.

If you take actual barometer readings each time the ADC count is recorded (over a period of significant fluctuation in air pressure), a graph similar to the one shown in Figure 29 can be produced.

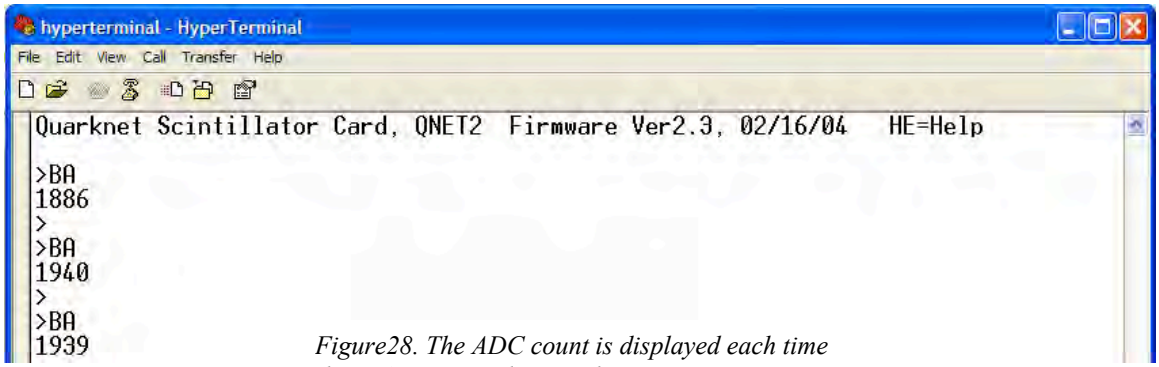



*Figure28. The ADC count is displayed each time the 'BA' command is typed.*

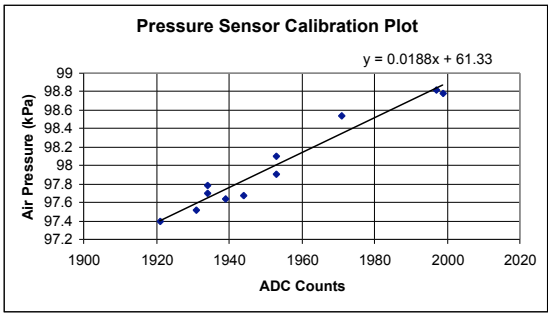



*Figure29. Calibration plot used to convert ADC counts to air pressure in kPa.*

Determine the slope and y-intercept for the best-fit line and input those values into the board via a keyboard command and your pressure sensor is calibrated.

Figure 29 illustrates the command sequence used in the barometer calibration. Note that the reciprocal of the slope $\left(\frac{1}{0.0188} = 53.2\right)$ is what the software calls "gain," while the y-intercept (61.3) is what the software calls "baseline."

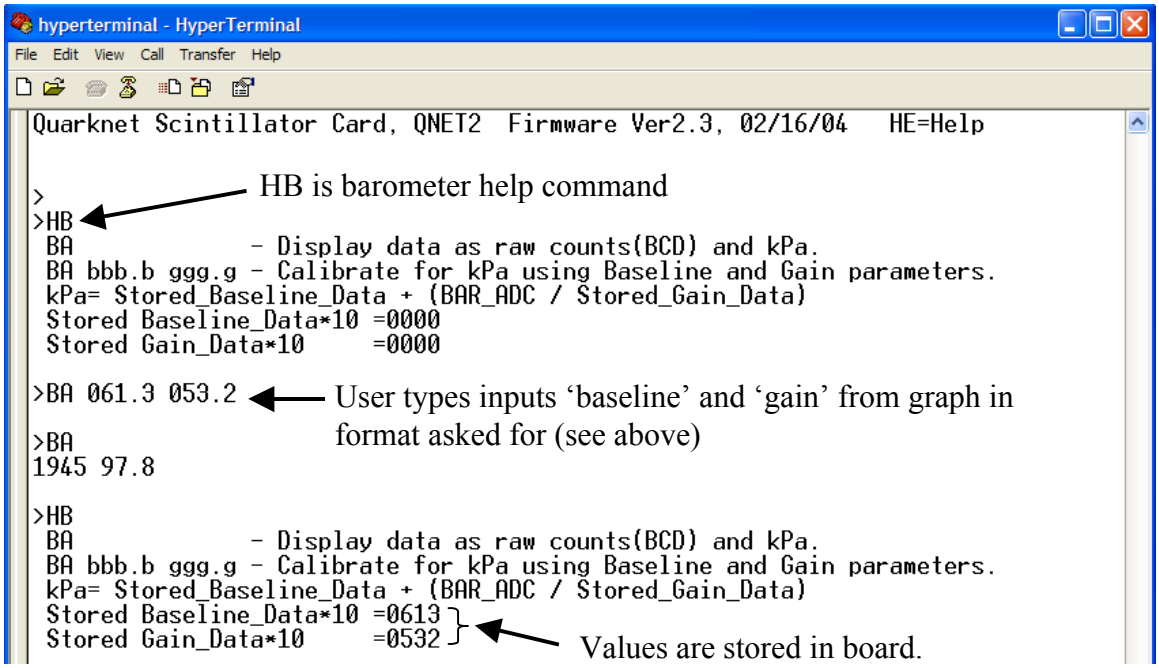



*Figure30. Command sequence used to calibrate the barometer.*

# Appendices

# Appendix A: History of Card Development

After participating in the 1994 Teacher Research Associates Program (TRAC) at Fermilab under the mentorship of Dane Skow, high school physics teacher Jeff Rylander was excited about his experience doing high-energy physics research and wanted to bring a *real* high-energy physics experiment back to his classroom. During the next year, Jeff collected the equipment needed to perform a cosmic ray muon lifetime experiment in his class. Fermilab electrical engineer Sten Hansen developed a low-cost circuit board for the setup that served as the logic board to analyze a single counter output and then deliver a readable signal to a classroom computer. In 1996, high school students at Maine East High School in Park Ridge, Illinois did this muon lifetime experiment.

Independent of Rylander's experiment, projects like CROP and WALTA began supplying schools with NIM crates and modules, on loan from Fermilab, to participants in school-network cosmic ray detector projects. The loaned equipment had a book value of about $10,000 per school. Although most of the modules were obsolete and no longer in demand for current experiments, the large cost drove the need to provide a very low-cost set of front-end electronics to these teachers. Also, off-the-shelf NIM modules could not handle GPS timing data and PC interfacing. Thus, while adequate for training teachers, these wide-area network projects could not begin taking real data in schools with the NIM hardware.

During 2000, Tom Jordan of Fermilab worked on the possibility of using new programmable logic and time digitizer chips—similar to ones used in Rylander's experiment—to put cosmic ray experiments in the hands of QuarkNet teachers around the country. At Tom's request, Fermilab engineers Sten Hansen and Terry Kiper developed a prototype front-end card for a simple table-top cosmic ray demonstration detector. This "Version 1" QuarkNet board put together components that were cheap, reliable and highly capable of integrating on a small board the discriminator, logic and scaler functions of NIM modules for four PMT channels, and a universal PC interface. However, only relatively coarse (~20 ns) pulse timing information could be provided.

At the SALTA Workshop, held as part of the Snowmass 2001 meeting, QuarkNet joined CROP and WALTA to plan an improved Version 2 board. The goals were:

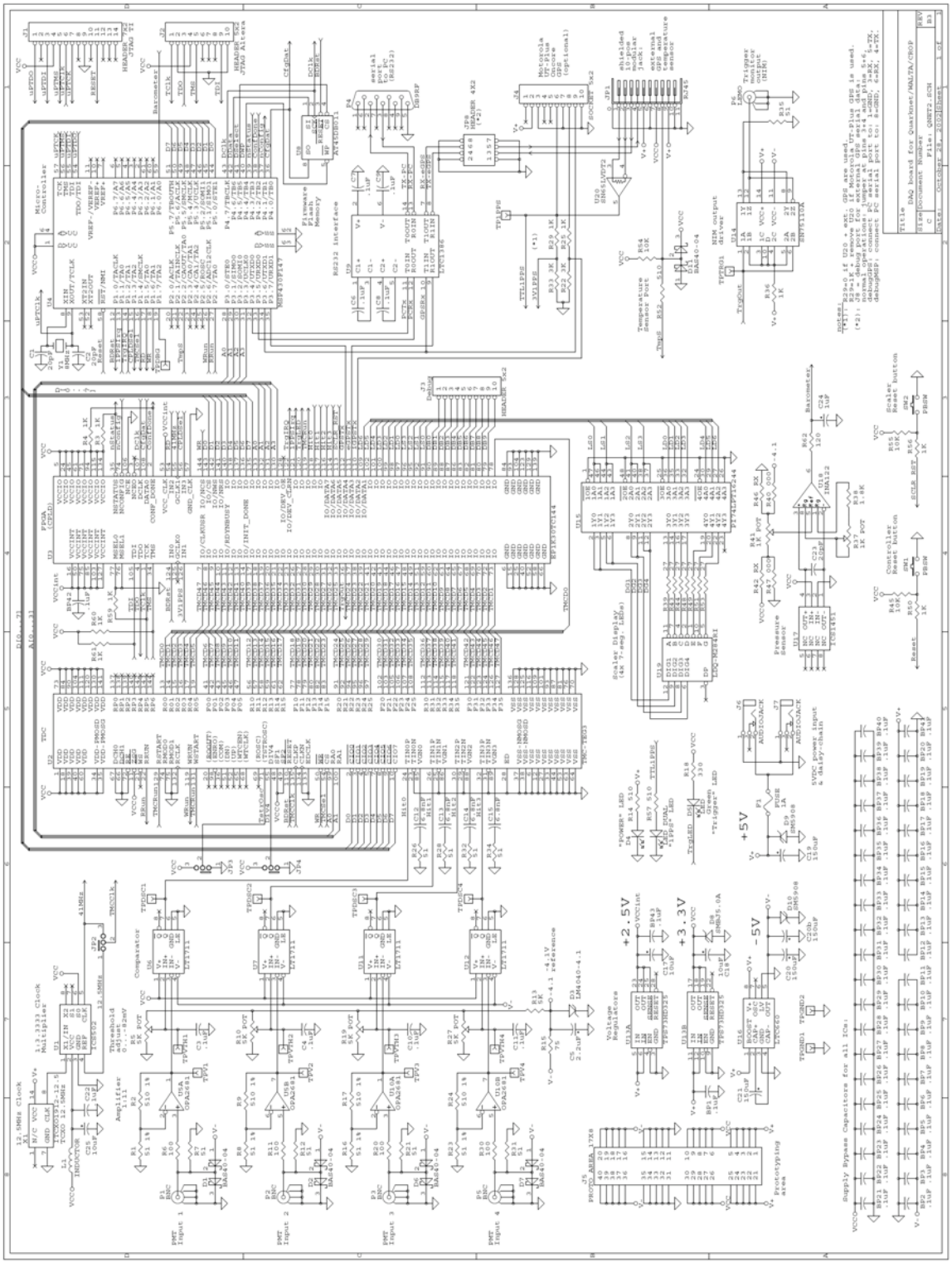
- Nanosecond local timing.
- <100 nanosecond absolute timing.
- Reporting of all pulse edge times within a reasonable time window.
- Up to 4-fold majority logic.
- Total cost per board under US$500.

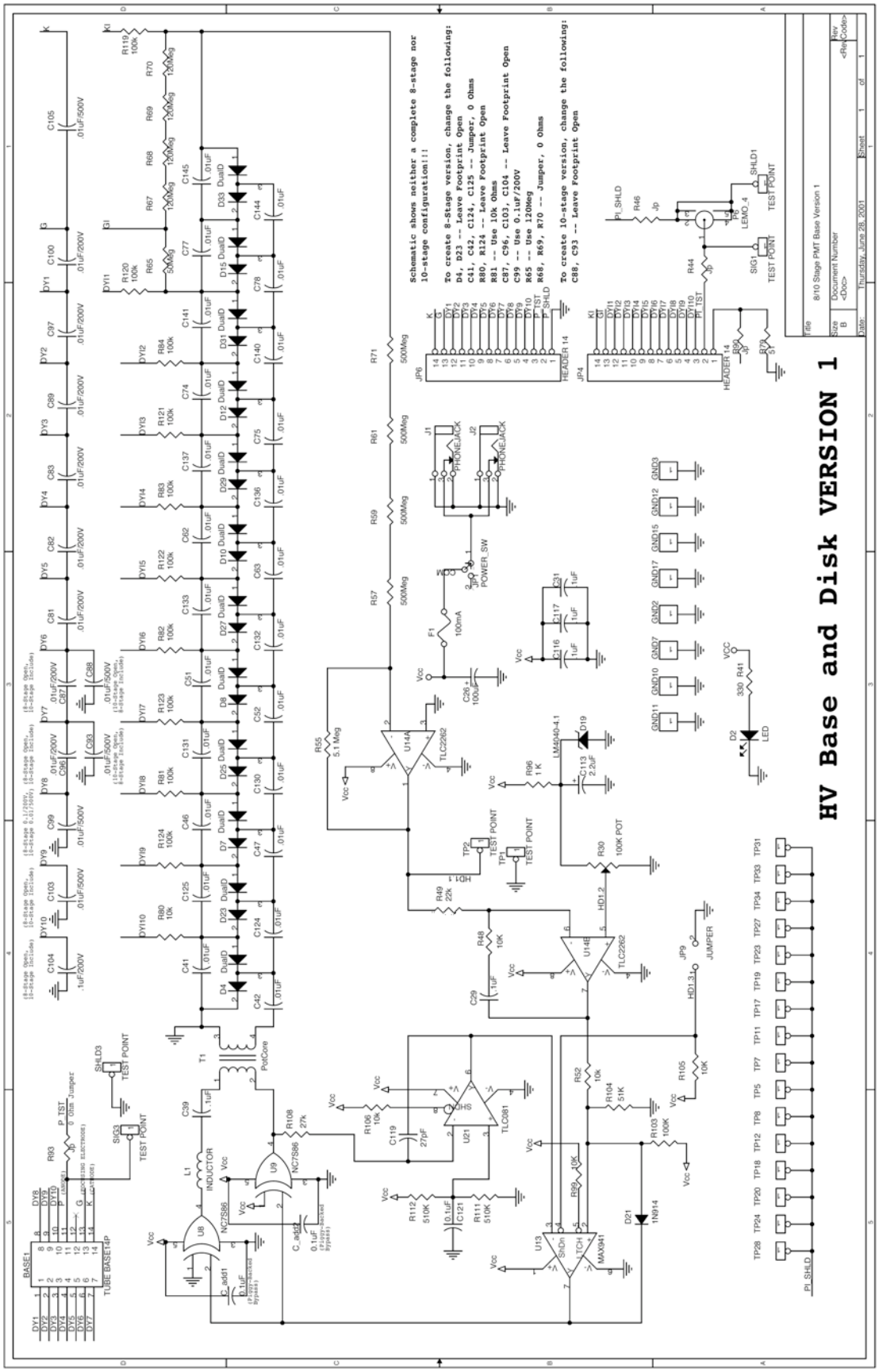
Remarkably, these goals were quickly achieved! A team at KEK, the Japanese equivalent of Fermilab, had developed a time-to-digital module for the ATLAS particle physics experiment using a custom-made ASIC chip called TMC (Time Measurement Cell). Samples, available at Fermilab, are used in the new board design to provide the detailed, precise timing data needed for air shower detectors. In addition, the TMC chips allowed for time-over-threshold (TOT)

estimates of pulse area. University of Washington engineer Hans Berns designed and implemented an on-board system to integrate data from a very low-cost (US$100) GPS receiver module. This provided the final piece of missing functionality. The Fermilab team revised the board and its firmware to better encompass the needs of air shower experiments. Team members from both Seattle and Lincoln worked on user-friendly LabView® interfaces for the boards, as well as black-box production specimens of the Version 2 DAQ software for standard tasks. After lab-testing prototypes, QuarkNet, CROP and WALTA distributed in the summer of 2003.

Currently, Fermilab is helping to produce several hundred DAQ boards and PMT bases for distribution to QuarkNet teachers around the country. A software development team is developing the web interface using Grid computing tools and techniques to allow cosmic ray data networks nationwide. Now students across the country are able to collaborate as they perform *real* HEP experiments.

# Appendix B: PMT Base and DAQ v2 Board Schematic Diagrams




HV Base and Disk VERSION 1
Schematic shows neither a complete 8-stage nor 10-stage configuration!!!
To create 8-Stage version, change the following:
D4, D23 -- Leave Footprint Open
C41, C42, C124, C125 -- Jumper, 0 Ohms
R80, R124 -- Leave Footprint Open
R81 -- Use 10k Ohms
C87, C96, C103, C104 -- Leave Footprint Open
C99 -- Use 0.1uF/200V
R65 -- Use 120Meg
R68, R69, R70 -- Jumper, 0 Ohms
To create 10-stage version, change the following:
C88, C93 -- Leave Footprint Open
8/10 Stage PMT Base Version 1
Document Number
Thursday, June 28, 2001
TUBE BASE14P
HEADER 14
HEADER 4
TEST POINT
JUMPER
POWER_SW
PHONEJACK
LED
INDUCTOR
PotCore
TLC2262
TLC081
MAX941
LM4040-4.1
NC7S86
1N914
100K POT

# Appendix C: Terminal Emulator Setup

## Hyperterminal

Hyperterminal, a terminal emulation program, often comes with a Windows-based PC or can be downloaded from the Internet. Hyperterminal allows the PC's monitor to display the data that is sent from the DAQ board to the PC's serial port or USB port.

Once this program is opened, a window appears that asks for a name for the connection that you are making. Enter a convenient name so that in the future you will not have to go through the initial setup routine each time you want to establish a connection.

Once you select "OK," the window shown in Figure 31 appears. You need to select the COM port that the DAQ board is connected to on the back of your computer.


Choose the COM port that the DAQ board is plugged into on the back of your computer (Serial port is COM1 and USB is COM3 for this computer.) COM3, the USB port, is selected here.

*Figure 30. This window illustrates where the COM port selection is made which tells the computer to which port the DAQ board is connected.*

You need to change two settings from their default values on the next screen. These modifications are shown in Figure 32.

Once these settings have been made, select the "Capture Text" _ "Start Capture" command from the "Transfer" menu. A window appears in which you type a name for the text file to which the datastream will be written. Once a file name and folder for that file have been selected, the "Start" button begins the process of writing data to the text file. As long as the connection is open, data is written to the file. This text file contains data for later analysis. To stop writing data to the file, select "Capture Text" _ "Stop Capture." The connection may be closed, and the text file is ready for analysis.

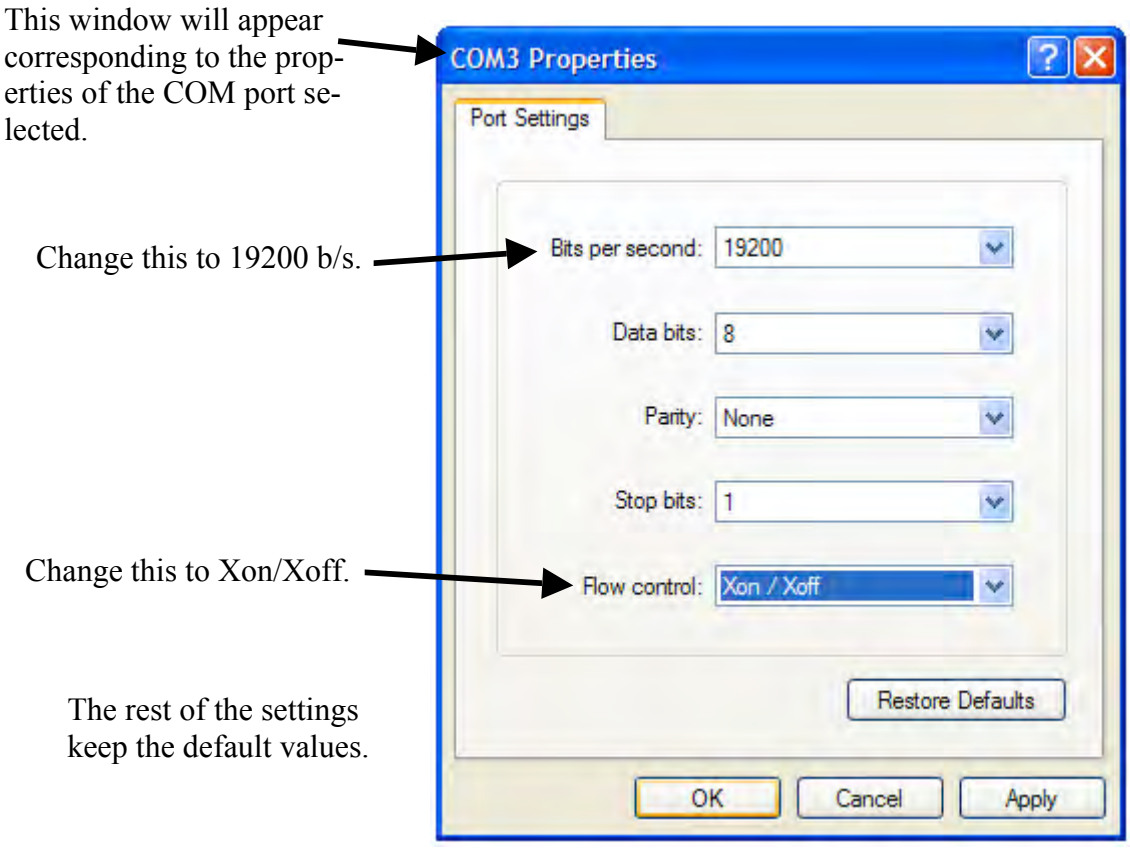



*Figure 32. Where baud rate and flow control settings are modified for proper reading of the DAQ datastream with Hyperterminal.*

## Zterm

Zterm is a terminal emulation program that can be downloaded and used on either a Mac or Windows-based PC. This program allows the computer to display the data that is sent from the DAQ readout board to the computer's serial port or USB port.

Once this program is installed and opened on your machine, the computer should auto detect the port your DAQ board is connected to. If this is in question, you can hold down the shift button while Zterm is opening. This opens a window showing all possible devices that are connected. You can select the USB device or serial port connection.

Select the "Settings" _ "Connection" menus. The window shown in Figure 33 appears. You need to change two values from the default settings: Data Rate = 192000 bits/sec and Flow Control = Xon/Xoff.

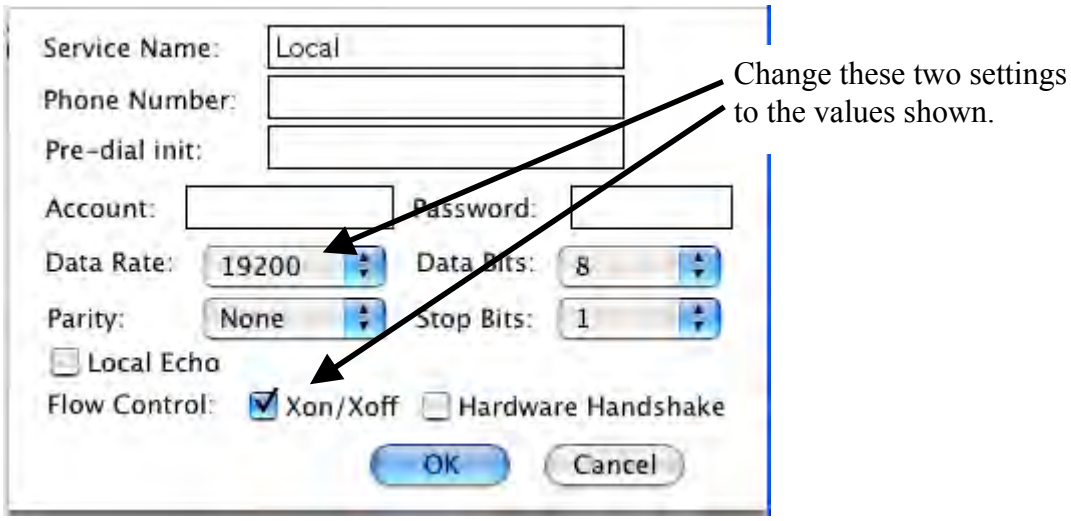



*Figure33. Where baud rate and flow control settings are modified for proper reading of the DAQ datastream with Zterm.*

Once the settings have been made, select the "Start Capture" command from the "File" menu. A window appears in which you type a name for the text file to which the datastream will be written. Once a file name and folder for that file have been selected, the "Start" button begins the process of writing data to the text file. As long as the connection is in place, data is continually written to the file. This text file contains data to be analyzed. To stop writing the data to the text file, select "Hang Up" from the "Dial" menu or just quit Zterm. The connection is closed, and the text file is ready for analysis.

## Unix

A simple Shell script allows the incoming datastream to be written to file on a Unix system. The script is very short and can be downloaded from the QuarkNet website.

You will be prompted to give a name and directory for the file to be written to and the length of time for the run. This program does not allow you to see the data on the monitor, however. Hyperterminal, Zterm, or some other terminal emulation program can be installed in order to view the datastream on the monitor.

# Appendix D: Precise Event Time Calculation Algorithm

$$\left.\begin{array}{l} t_{COARSE} = hh:mm:ss.sss, \quad mm:dd:yy \\ delay = \pm\, 0.ddd \end{array}\right\} \text{from data record}$$

$$\text{integer seconds}: ss = \text{round } (ss.sss + 0.ddd) \text{ to nearest second}$$

$$N_X = \text{clock count at time } X:$$

$$\text{Clock rate } R_1 = (N_{LATEST\_1PPS} - N_{PREVIOUS\_1PPS}) \text{ ticks/sec}$$

$$t_{NANOSEC} = \frac{(N_{TRIGGER} - N_{LATEST\_1PPS})}{R_1}$$

$$t_{EVENT} = ss + t_{NANOSEC}$$

$$\left(\begin{array}{l} \text{Alternatively, use a } K\text{ - second running average for the clock rate :} \\ R_K = \dfrac{(N_{LATEST\_1PPS} - N_{Kth\_PREVIOUS\_1PPS})}{K} \end{array}\right)$$

## Example Calculation

The following two simultaneous events were recorded by two boards triggered by a pulser. One was captured on a laptop and one on a PC. For the purpose of discussing the data format, each column is labeled with a letter. The lines are split in half for readability.

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| | Trigger Count | RE0 | FE0 | RE1 | FE1 | RE2 | FE2 | RE3 | FE3 |
| Laptop | 1BB7FB6A | B7 | 01 | 37 | 01 | 00 | 01 | 00 | 01 |
| | 1BB7FB6D | 01 | 24 | 01 | 24 | 00 | 01 | 00 | 01 |
| | | | | | | | | | |
| PC | 1BB7FB52 | B5 | 01 | 35 | 01 | 00 | 01 | 00 | 01 |
| | 1BB7FB55 | 01 | 21 | 01 | 21 | 00 | 01 | 00 | 01 |

| | J | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|
| | 1PPS Count | Time (GMT) | Date | Valid? | # Satellites | Error Bits | Correction |
| Laptop | 1B1F27A0 | 210727.727 | 110603 | A | 06 | A | +0096 |
| | 1B1F27A0 | 210727.727 | 110603 | A | 06 | 8 | +0096 |
| | | | | | | | |
| PC | 1B1F2787 | 210727.372 | 110603 | A | 06 | A | +0450 |
| | 1B1F2787 | 210727.372 | 110603 | A | 06 | 8 | +0450 |

*Table 4. Example event data from PC and laptop used to illustrate how to calculate precise event times.*

**Column A** is the clock value at which the triggers came. **Columns B through I** contain timing information in fractions of a clock period. Each input has data for the rising edge (RE) and falling edge (FE) of the pulse for each channel. **Column J** is the clock value at which the 1PPS arrived. **Column K** is the time and **L** the date at which the 1PPS arrived. This value must be corrected by **Column P** and rounded to the next second. **Column M** is "A" when the GPS data in a row is valid and "V" when it is invalid. **Column N** is the number of satellites the receiver is using. **O** is four error bits, each of which indicates the trigger IRQ status. Bit 0 indicates that the 1PPS interrupt is pending, bit 1 that the trigger interrupt is pending, bit 2 that the GPS data could be corrupt (write in progress during readout), and bit 3 that the current or last 1PPS pulse was not within 41666666 ± 50 clock ticks. To calculate the absolute time of the 1PPS, we must add K to P and round to the nearest integer second, first converting to seconds after midnight. Separate K into hours, minutes, seconds, and milliseconds. Multiply by 3600 for hours and 60 for minutes.

$$Round\left(K_{\sec}+\frac{P_{m\sec}}{1000}\right)=T_{1PPS}.$$

To calculate the relative time between the input pulses you must understand the edge data represented in columns B through I. Each column contains a tag bit (bit 5) to indicate whether the column contains data. Bits 0-4 are the data. Bit 6 is always 0. Bit 7 of column B only indicates whether a new trigger is contained in the line; for columns C through I bit 7 is always 0. See bit map.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 (Trig tag for RE0) | 0 | Channel Tag | Data | Data | Data | Data | Data |

*Table 5. Bit map for columns B through I for event data shown in Table 4 above.*

Bit 7 of Column B indicates a new trigger. To calculate the time of each trigger, the relative time between the 1PPS and the trigger must be found. This can be calculated by subtracting the counts in Column J from those in A (first converting to decimal). $A-J=\Delta T_{clk}$. The data from columns B through I may be added but give superfluous precision (for small data sets) given the 1PPS is measured only in integer clock periods. To calculate the absolute time of column A, convert $\Delta T_{clk}$ to seconds (multiply by the reciprocal of the clock frequency), $\Delta T_{clk}\left(\frac{1}{f_{clk}}\right)=\Delta T_{\sec}$, and add to $T_{1PPS}$. Thus the absolute time of the trigger is: $T_{1PPS}+\Delta T=T_{abs}$. The long form is: $Round\left(K_{\sec}+\frac{P_{m\sec}}{1000}\right)+\frac{(A-J)}{f_{clk}}=T_{abs}$. In this case the triggers happen at the following times in seconds after midnight:

PC:

$$T_{abs}=Round\left(76047.372+\frac{450}{1000}\right)+\frac{(465042258-455026567)}{41666667}=76048.240376582\ \text{sec}$$

after midnight

Laptop:

$$T_{abs} = Round\left(76047.727 + \frac{0096}{1000}\right) + \frac{(465042282 - 455026592)}{41666667} = 76048.240376558 \text{ sec}$$

after midnight

The difference between the two times in this example is the length of one clock period. This is to be expected because the clock period is the limit of the resolution of the triggers. By writing a program to examine the files containing readout data, you can easily check the time difference for each event. For these data, the time differs no more than 2 clock periods verifying that the GPS data can be used in correlating data between Quarknet DAQ v2 cards.

# Appendix E: Details of Time Coincidence and Data Handling

This appendix explains the operation of the card's time coincidence and data handling at a deeper level. To keep the cost low, the project engineers created a very clever scheme for handling the tasks needed with minimal resources. Of course, "clever" also means "not simple." For this reason, the way in which triggering is handled differs in some details from what one might expect for simple modular fast electronics like NIM coincidence units. The card handles coincidences as described below.

## Descriptions of Gate Window and TMC Delay

We define two parameters, which you can set:

1. ***w* = Gate Window**: Just like the "width" setting in a NIM module, the gate window determines how close together pulses must be to cause the card to trigger. Note: In digital electronics, the term "gate" means a signal used to enable ("open the gate for") passage of other signals or data.
2. ***d* = TMC Delay**: This quantity determines which pulse edges get read out into the datastream when a trigger occurs by delaying this information until the trigger actually happens.

Both *w* and *d* are measured in units of 24 nanoseconds, the internal clock tick interval.

Consider the actual physical quantities and relate them to the two parameters listed above.

- **$T_{TRG}$:** This quantity is the maximum time window during which real physics processes that we want to study might cause a trigger. In other words, if we set the card for 2-fold coincidences and one channel gets a signal at t = 0, at least one other channel must show a signal before $t = T_{TRG}$ or you lose detailed information about that pair of pulses. The examples below may help.
- **$T_{WIDTH}$:** This quantity is the duration after the first pulse in a trigger during which we want to record all rising and falling edge times. The channel in which these extra pulses occur does not matter; all of them are saved.

Set the values of *w* and *d* as follows:

- Set $d > T_{TRG}$, but no smaller than two clock ticks.
- Set $w > T_{WIDTH} + d - 1$.

Whenever an above-threshold voltage is detected on any channel, that channel turns on an internal trigger bit for *w* clock ticks *(w* x 24 ns). At the end of that time the internal trigger bit is reset to zero.

The card's startup default setting is for 1-fold coincidence, which means it triggers on singles: any above-threshold signal on any channel produces output data. Unless your PMTs are extremely quiet, actually triggering on singles will quickly overwhelm the card.

When the card is set for 2-fold coincidences, a trigger is declared any time that two or more channels' trigger bits are "1" simultaneously. If the card is set for 3-fold coincidence, then it requires three channels with

their bits set simultaneously, and similarly for 4-fold.

The trigger bits must overlap by at least one 24 ns clock tick to activate a trigger. If one channel's bit drops and another rises, no trigger occurs. Also, the trigger is active for exactly the overlap period. The card has a TRG output that goes high when a trigger occurs. You can watch this on the oscilloscope and see its duration change as the signals are closer or further apart.

The TMC delay *d* is very different than the almost-just-like-NIM coincidence described above. The reason for this is that we want to do more than count triggers; we also want to record all the edges that contributed to a given trigger. It is hard to program the circuits inside the board to look back in time to the first pulse that contributed to the trigger when the second pulse might have arrived a significant amount of time later. (It was also hard to make the old NIM modules do this.) Instead, we separate the part that generates the trigger from the part that reads out the timing information and then we *delay this detailed timing information so that it is read out after the trigger is generate*d. The card needs to save only information that happens when the gates overlap.

Consider the card's startup default settings, $d = 6$ and $w = 10$. Imagine a physics event where one pulse arrives three clock ticks after another. The trigger will occur when the second pulse arrives—three clock ticks after the first. At this point, the card detects trigger bit overlap, and the TRG output goes high. Internally, however, pulse edge time information is delayed six clock ticks *(d =* 6), so the information about that first pulse is available after just three more clock ticks (the trigger is still on). The information about the second pulse arrives after another three ticks. (Again, the trigger is still on.)

Because both edges appear when the trigger is on, they appear in the datastream sent to the computer.

# Examples of Event Timing Diagrams

Below are four examples of event timing diagrams showing 2-fold coincidences, with d = 6, w = 10. In each case, pulses of width 2 clock ticks (48 ns) arrive at inputs 0 and 1, but with different time separations in each example. The timing diagrams show the discriminator output, the channel trigger gate, and the TMC output after delay *d,* for each channel, plus the coincidence trigger gate, and the edge time information obtained from the TMC upon readout.

## Example A

The signal in channel 1 arrives 2 clock ticks (48 ns) after the signal in channel 0. Figure 34 illustrates that the trigger gate is 8 ticks long (overlap time of the two 10-tick channel gates). All edges get reported.

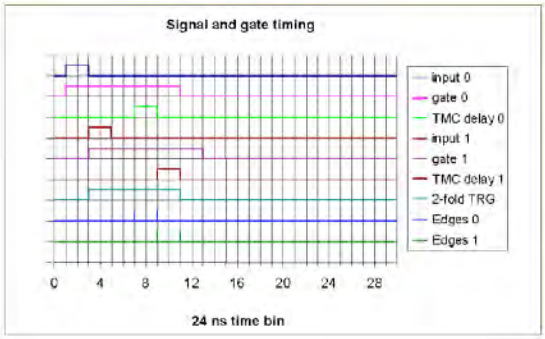



*Figure34. Signal and gate timing diagram for Example A.*

## Example B

The signal in channel 1 arrives 3 clock ticks (72 ns) after the signal in channel 0. In this case, the trailing edge of the delayed TMC signal for channel 1 lies outside the card trigger gate (overlap time of the individual

channel gates), so only its leading edge is reported. This is illustrated in Figure 35.

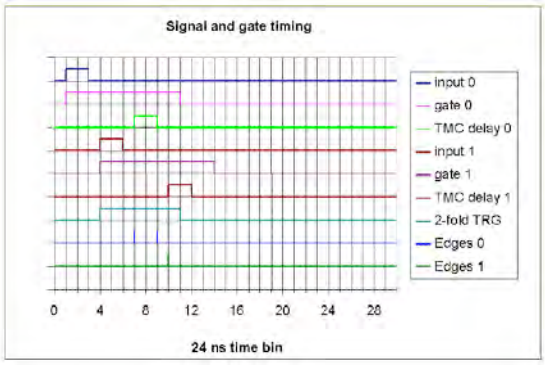



*Figure 35. Signal and gate timing diagram for Example B.*

## Example C (Figure 36)

The signal in channel 1 arrives 5 clock ticks (120 ns) after the signal in channel 0. Note that the delay value is 6 clock ticks (144 ns). In this case, *both* edges of the signal on channel 1 lie outside the trigger (overlap) window and are not reported. If your physics application involves simply counting triggers (as in a muon telescope), this does not present a problem. However, if you are analyzing edge time data, as in a muon decay or air shower experiment, this would appear to be a 1-fold event! If this situation represented a real physics process you wanted to study, you would need to lengthen the values of *d* and *w* to accept all the edges in this event. If this is not done, this looks like a random background event that you have to eliminate or ignore when you analyze the data.

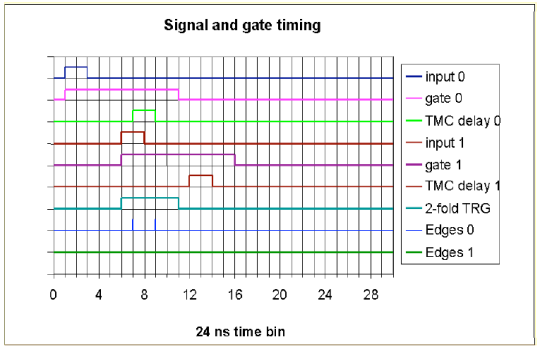



*Figure36. Signal and gate timing diagram for Example C.*

The next example is an actual event recorded by the DAQ card, on Aug. 8, 2003, with a mini-array of 4 counters. In this example, the card was set with d = 6, w = 10 (the default startup values), and for 2-fold or greater coincidence level. (The interpretation of the data words for this event can be found in Appendix F.)

## Example D (Figure 37)

Notice the datastream had 5 lines of data for this single event.

```
80EE0049 80 01 00 01 38 01 3C 01 7EB7491F 202133.242 080803 A 04 2 -0389
80EE004A 24 3D 25 01 00 01 00 01 7EB7491F 202133.242 080803 A 04 2 -0389
80EE004B 21 01 00 23 00 01 00 01 7EB7491F 202133.242 080803 A 04 2 -0389
80EE004C 01 2A 00 01 00 01 00 01 7EB7491F 202133.242 080803 A 04 2 -0389
80EE004D 00 01 00 01 00 39 32 2F 81331170 202133.242 080803 A 04 2 +0610
```

*Figure37. A sample event containing five lines of data. Each line is comprised of the same 16 word format with each word separated by a space.*

The pulse timing diagram, Figure 38, shows the four PMT channels' trigger bits as a function of time in nanoseconds. (Remember, channels are numbered 0 through 3). Pulse edge times are recorded with 0.75 ns precision. Recall, however, that the *trigger* logic works in time bins of 24 ns.

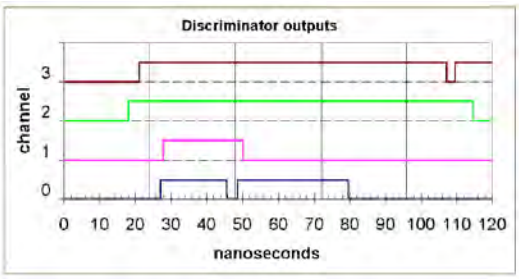



*Figure 38. Discriminator output timing diagram for Example D.*

Figure 39 shows the channel trigger bits as a function of time, in units of 24 ns.

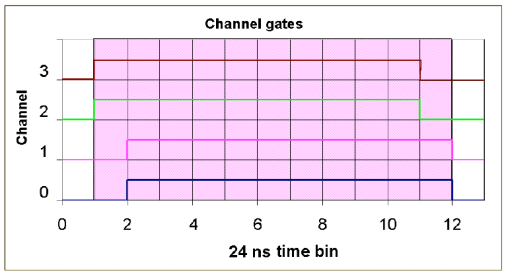



*Figure 39. Channel gate timing diagram (card trigger interval is shaded) for Example D.*

The following sequence of events occurs:

1. The first pulse arrives on channel 2 at t = 1 (in units of 24 ns). Trigger gate 2 is set for 10 ticks. The TMC records the pulse's leading edge at 18 ns relative to the last 24ns clock tick.
2. Three nanoseconds later, a pulse arrives on channel 3. Trigger gate 3 is set for 10 ticks. The TMC records this pulse's leading edge at 21 ns.
3. Trigger gates 2 and 3 are set simultaneously, so a trigger gate (TRG) is asserted. If no further signals arrived, the trigger would stay on for ten 24ns time bins (240 ns) since the signals on channels 2 and 3 arrived inside the same clock tick.
4. In 24ns time bin 2, pulses arrive in channels 0 and 1, setting *their* trigger gates for 10 ticks. Thus, the 2-fold coincidence requirement is met continuously from time bin 1 until time bin 11, a total of 11 clock ticks or 264 ns (shaded area in Figure 39).
5. At the end of this interval the trigger gates for all channels are down, so the TRG gate is dropped. The pulse edge times stored during the time the trigger was on are read out of the TMC chip and reported.
6. Only edges that are within the trigger interval *after* they have been delayed by 6 ticks are put into the datastream. Thus, all the pulses in Figure 36 are recorded, but the second falling edge of channel 3 (which is off the figure to the right) will fall outside the shaded area in Figure 38 and will not be recorded.

*Note:* The actual values encoded in the datastream represent the *delayed* times. If it is vital to get the actual time for each edge, then each of them actually arrived *d* x 24 ns earlier than the time given in the data record. Usually only relative timing matters, and this kind of accuracy is not necessary.

# Appendix F: Acronym and Jargon Dictionary

| *Acronym or Term* | *Description* |
|---|---|
| 1 PPS | One Pulse Per Second. GPS signal used for precise time synchronization. |
| AC | Alternating Current. |
| ASCII | American Standard Code for Information Interchange. Numeric code for text files. |
| ASIC | Application-Specific Integrated Circuit. A custom-made chip, like the TMC chip used in the DAQ board. |
| ATLAS | A Toroidal LHC ApparatuS. A large international collaborative experiment at the Large Hadron Collider (LHC) at CERN Laboratory in Switzerland. < http://atlas.web.cern.ch/ > |
| Binary | Base 2 number system with only 2 digits: 0 and 1. For example, the decimal numbers 1, 2, 3, and 4 are written 1, 10, 11, and 100 respectively in binary notation. To distinquish a binary number from a decimal (base 10) number, a subscript "2" follows the number: $100_2$ is the same as $4_{10}$. |
| Bit | Binary Digit. Component of a number expressed in the binary system using only 0s and 1s. |
| Byte | A block of four binary bits which can store one ASCII character. Also, a unit of capacity for computer storage devices (hard drives, memory chips, etc.). |
| BNC | Either Bayonet/Neill-Concelman (the inventors of the BNC connector), or Berkeley Nucleonics Corporation (opinions differ). The name for a commonly used connector for coaxial cable. |
| Coincidence | In high-energy and nuclear physics, a set of events that occur "simultaneously" (i.e. within some small period of time.) For example, "3-fold coincidence" means particles are detected simultaneously in three separate detectors. |
| Counter | The detector element in high-energy physics experiments. Refers to the scintillator, light guide, photomultiplier tube (PMT) and base assembly. |
| CPLD | Complex Programmable Logic Device. A general-purpose integrated circuit chip which can be used in many ways, as programmed by the user. |
| CPU | Central Processing Unit. The "heart" of a computer. |
| CROP | Cosmic Ray Observation Project. Nebraska-based project to study large scale cosmic ray showers.<br>< http://physics.unl.edu/~gsnow/crop/crop.html > |
| DAQ | Data Acquistion. |
| DB9 | Serial Communications D-shell connector with 9 pins. Used on the GPS device. |
| DC | Direct Current. |

| | |
|---|---|
| Discriminator | In high-energy and nuclear physics, an electronic circuit that produces an output pulse only when its input voltage exceeds a selected (threshold) level. |
| DOE | United State Department of Energy. < http://www.doe.gov/ > |
| Event | In high-energy and nuclear physics, a coincident pattern of detected particles that is likely to be of interest, and should be recorded. (Derived from the use of the same term in special relativity: an occurrence at a particular point in space and time.) |
| FIFO | First In, First Out. A type of memory buffer where data is taken out in the same order as it is put in. |
| FNAL | Fermi National Accelerator Laboratory. Also known as Fermilab. < http://www.fnal.gov > |
| FPGA | Field-Programmable Logic Array. One variety of CPLD. |
| Gate | In digital electronics, a signal that enables another signal to be transmitted. The digital equivalent to a toggle switch. |
| GPS | Global Positioning System. < http://tycho.usno.navy.mil/gps.html > |
| HEP | High-energy physics. The branch of physics that studies the nature and behavior of high-energy particles. |
| Hex | Hexadecimal. A number system with base 16. Use the characters 0-9 and A-F to represent numbers with decimal values of 0-15. Widely used in computers, since one hex digit represents 4 binary bits. Data storage in 'bytes' of 4 bits or multiples of 4 (8, 16, 32, etc.) is very common. Example: $A37F_H = 41855_{10} = 1010\ 0011\ 0111\ 1111_2$. The subscript 'H' denotes a hex number. |
| HV | High Voltage |
| KEK | Koh-Enerhughi-Ken-Kyu-Sho. A high-energy accelerator research organization in Tsukuba, Japan. The Japanese equivalent to Fermilab. < http://www.kek.jp/intra.html > |
| LED | Light-Emitting Diode |
| LEMO | (Company name) Miniature coax cable connectors—serve the same function as BNCs. |
| MCU | Micro Controller Unit. (See also "CPU.") |
| NIM | Nuclear Instrument Module. A standard for interchangeable nuclear physics electronic modules dating back to the 1950s. |
| NMEA | National Marine Electronics Association. Commonly used data specification with navigation instruments, e.g., GPS receivers. |
| ns | Nanosecond = 1 billionth ($10^{-9}$) of a second. |
| NSF | National Science Foundation. < http://www.nsf.gov/ > |
| Op-Amp | Operational Amplifier. An integrated circuit element that amplifies a signal. Op-amps are the building blocks of many kinds of digital circuits. |
| PC | Personal Computer. A generic term that covers Windows, Mac, or Linux- |

| | |
|---|---|
| | based computers. |
| PMT | Photo Multiplier Tube. Vacuum tube that converts extremely low light levels (down to single photons) into electrical signals. |
| QuarkNet | National science outreach program funded by NSF and DOE. < http://quarknet.fnal.gov > |
| RS-232 | Recommended Standard 232. A connector commonly used for computer serial interfaces. |
| SALTA | Snowmass Area Large-scale Time-coincidence Array. < http://faculty.washington.edu/~wilkes/salta/ > |
| Scaler | Pulse counter. |
| TDC | Time-to-Digital Converter. |
| TMC | Time Memory Cell. Name of the specific TDC chip used in the DAQ board. < http://research.kek.jp/people/araiy/TEG3/teg3.html > |
| TOT | Time Over Threshold. |
| TRG | Trigger. In high-energy and nuclear physics, the digital signal that indicates an event of interest has occurred and data should be logged. |
| UNL | University of Nebraska, Lincoln. |
| USB | Universal Serial Bus. |
| UTC | Universal Time Coordinated. Formerly known as Greenwich Mean Time (GMT). |
| UW | University of Washington, Seattle. |
| VAC | Volts of Alternating Current system. Most US appliances use 110 VAC. |
| VDC | Volts of Direct Current system. The DAQ board used 5VDC. |
| WALTA | Washington Large-scale Time-coincidence Array. < http://www.phys.washington.edu/~walta > |