

MUTABLE DETECTOR ARRAY SOFTWARE FOR THE DETECTION OF GAMMA
EMISSIONS IN CLASSROOMS AND THE FIELD

A Thesis

by

GENTRY CHARLES HEARN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2012

Major Subject: Health Physics

Mutable Detector Array Software for the Detection of Gamma Emissions in Classrooms
and the Field

Copyright 2012 Gentry Charles Hearn

MUTABLE DETECTOR ARRAY SOFTWARE FOR THE DETECTION OF GAMMA
EMISSIONS IN CLASSROOMS AND THE FIELD

A Thesis

by

GENTRY CHARLES HEARN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Craig Marianno
Committee Members,	John Ford
	Thierry Lints
Head of Department,	Yassin Hassan

August 2012

Major Subject: Health Physics

ABSTRACT

Mutable Detector Array Software for the Detection of
Gamma Emissions in Classrooms and the Field. (August 2012)

Gentry Charles Hearn, B.S., Tarleton State University

Chair of Advisory Committee: Dr. Craig Marianno

Detector arrays are required for many applications in health and defense. Whether searching for contraband or controlling radioactive spread after an event, a “passive” data collection strategy is a key component. This can take the form of portal monitors attached to a fixed location or a movable array, attached to a vehicle or person, which searches for abnormalities in the radiation background. The main goal of this project was to create software that would allow the digiBASE-E to be used to create arrays of gamma detection instruments and gather data over a long span of time. To take full advantage of the digiBASE-E, the software focused on the list mode capabilities of these devices. List mode attaches a timestamp to each detection event. Every particle detected can be traced to a particular point in time, and the full history of the device’s detection over the acquisition period can be reconstructed. The list mode ability of the digiBASE-E is a powerful tool for producing arrays of detectors, as a more familiar spectrum can be generated for any arbitrary section of time, even after-the-fact.

The software package, called “CraneWow”, was field tested at the Port of Tacoma in August of 2011. Perl scripts included as part of the package were used to

partially analyze the data collected, allowing for verification of the proof-of-concept's success. The software was written in C/C++, with supplemental scripts written in Perl to facilitate processing of the data once collected. Additionally, a user manual and programming guide were written to allow easy use and maintenance of the software.

DEDICATION

To my parents

ACKNOWLEDGEMENTS

I would like to thank Dr. Marianno for taking a chance on me and for putting up with me throughout this project. Thanks to my other committee members, Dr. Ford and Dr. Lints, for their support.

I would also like to thank the Nuclear Engineering Department for providing such a supportive environment and community. Without the support of the Department and my friends and colleagues, I never would have made it this far.

Thanks especially to my family, for always nudging me to fulfill my potential, and for providing the means to do so.

Finally, thanks to Drs. Tom and Linda Talley, without whom I would likely not have any of my degrees.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION	v
ACKNOWLEDGEMENTS	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES.....	ix
LIST OF TABLES	x
INTRODUCTION.....	1
Objectives.....	2
Review.....	5
NARRATIVE.....	8
Origins.....	8
Software, Hardware, and Documentation Difficulties	11
Final Steps.....	15
DIGIBASE-E SHORTFALLS	17
Throughput.....	17
Documentation and Examples.....	21
PROGRAM OVERVIEW.....	25
CraneWow.....	25
Perl Post-Processing Scripts.....	30
PROOF OF CONCEPT.....	33
Overview.....	33
Results.....	34
CONCLUSION.....	36

	Page
REFERENCES	38
APPENDIX A DEFINITIONS	40
Word	40
Raw Data	40
File Formats	41
List Mode	43
PHA Mode	43
Sync Mode	43
Real time	44
Live time	44
TCP/IP	44
DHCP	44
RJ-45 / 8P8C	44
Power over Ethernet	45
Radiation	45
Scintillator	45
ADC Word	46
Spectrum	46
Perl	47
APPENDIX B USER MANUAL	48
APPENDIX C PROGRAMMING MANUAL	49
VITA	50

LIST OF FIGURES

	Page
Figure 1 - The ORTEC digiBASE-E. [2]	2
Figure 2 - Crane-mounted spreader bar. [3]	4
Figure 3 - Example CraneWow window for a digiBASE-E named "Small"	10
Figure 4 - Test procedure diagram for "midline" configuration.	18
Figure 5 - CraneWow Master Box.	25
Figure 6 - CraneWow Detector Window, unselected.....	26
Figure 7 - CraneWow Detector Window, selected.....	27
Figure 8 - CraneWow Detector Dialog, List Mode and High Voltage engaged.	28
Figure 9 - Example histogram time spans file.....	31
Figure 10 - Sample count-rate graph from Tacoma trip.....	34
Figure 11 - Sample radiation spectrum from Tacoma trip.	35

LIST OF TABLES

	Page
Table 1 - Throughput crash test data for digiBASE-E.	19

INTRODUCTION

Detector arrays are required for many applications in health and defense.

Whether searching for contraband or controlling radioactive spread after an event, a “passive” data collection strategy is a key component. Passive, in this sense, means that the subject of the search is not stimulated to emit radiation. This can take the form of portal monitors attached to fixed locations or a movable array (attached to a vehicle or person) that searches for abnormalities in the radiation background. The advantages of multiple detectors are many.[1]

The digiBASE-E from ORTEC (Figure 1) is a relatively new piece of hardware that encapsulates many of the electronic devices used to receive and interpret signal from a radiation detector. These small photomultiplier tube (PMT) endcaps can be attached to a gamma detector, receiving power and communicating through power-over-ethernet (PoE). PoE is a technology which provides electrical power using standard Ethernet cabling, along with the networking signal. This feature makes them ideal for quick and scalable gamma detection arrays through the elimination of extra cords and the ability to use existing infrastructure. However, commercial software that takes full advantage of the other capabilities of these endcaps does not exist.



Figure 1 - The ORTEC digiBASE-E. [2]

The key feature of relevance to this project is the ability to record data in “list mode”. This mode allows each radiation detection event to be associated with a time stamp. Thus, a complete history of the radiation environment experienced by the detector can be recreated. When collecting data for long periods in a dynamic situation (such as the many cargo containers picked up and dropped off by a crane at a seaport), changes in the radiation spectrum can be matched up with events in the area surrounding the detector.

Objectives

The main goal of this project was to create software that would allow the digiBASE-E to be used to create arrays of gamma detection instruments and gather data over a long span of time. The list mode ability of the digiBASE-E is a powerful tool for producing arrays of detectors, as a more familiar spectrum can be generated for any arbitrary section of time, even after-the-fact. This allows for long-term data gathering without prior knowledge or immediate monitoring of the timing of events, such as

pickup or drop-off of a cargo container. The software is able to obtain and record the list mode data, as well as provide basic manipulation of the device settings. This software is also able to create histograms of arbitrary time-length, whether in realtime or from a previously saved data file.

The software was written specifically for a project involving radiation detection at seaports. The cranes which pick up and move the large cargo containers are an ideal place to scan for radiation anomalies. Each of the millions of containers entering the United States each year will be handled by such a crane. The spreader bars at the end of the cranes that actually grasp the containers have space for detectors to be mounted (Figure 2). The project collected count rate and spectral information from several sodium iodide detectors mounted on the spreader bar, recording data about the gamma radiation spectrum seen during normal port operations. This data has, in the past, not been made available by the companies trying to build and sell these systems.



Figure 2 - Crane-mounted spreader bar. [3]

The end product, in the form of the software package CraneWow, was designed for simple gathering and storage of the data, as well as a real-time generation of a file containing the current radiation spectrum as seen by each device. This data can be loaded into Excel for an immediate overview of the current situation. The data file, while not readable by other existing software, can later be processed into formats openable by Excel or several radiation spectrum programs, with many analysis techniques available.

A further goal of the project was to create a generalized program able to be used for other projects in the future, and for teaching. The summing capabilities and real-time data file generation for different detectors allows for detector location information

(possibly revealing source locations or self-shielding issues) to be obtained for an array with minimal difficulty. In addition, the code must be well-organized and documented to facilitate further upgrades and updates. A user and programmer manual have been written to allow for ease of use and to further enhance the ability of future programmers to maintain the software. The user manual contains set-up and use information, as well as a troubleshooting guide for common problems. This limits the time spent manipulating the system versus time spent gathering useful data.

Review

Gamma rays are one result of radioactive decay by unstable isotopes. These photons allow the nucleus to rid itself of energy stored due to an unstable energy state. This instability happens frequently for the higher-mass elements, as well as the byproducts of nuclear reactions such as fission fragments and activated materials. All of these materials are of interest to those that monitor the nation's ports, as these can all be used in the creation of radiological dispersion devices (RDDs) or improvised nuclear devices (INDs). A variety of means exist for the detection of these materials, chiefly the detection of the gamma rays that they emit.[4, 5]

The digiBASE-E is designed for use with a sodium iodide crystal, doped with thallium, but can be used with any scintillator attached to a standard two-inch PMT base. This project employed the use of sodium iodide doped with thallium (NaI(Tl)). These crystals, when hit with ionizing radiation, emit small bursts of light that are detected by electronics attached to the crystal. It is these signals which indicate a radiation detection event.[4, 5] NaI was chosen for this project because it is relatively inexpensive, can be

produced in large sizes, and has adequate resolution for radionuclide identification. The digiBASE-E is designed for use primarily with this type of crystal. [6] The digiBASE-E is a relatively new piece of hardware. Thus, information on applications specific to the digiBASE-E is somewhat scarce. However, there have been several projects involving the digiBASE, which is a USB-centered endcap with similar capabilities.

Researchers at Idaho national labs devised an explosives detection system incorporating the digiBASE. A target, such as a truck, is bombarded with neutrons before being allowed to enter a base. The resultant radiation from activated material is recorded using detectors attached to several digiBASE units and analyzed to determine possible contents. This is an example of an active detection system, where the object being scanned is stimulated to emit radiation.[7] Researchers at ORTEC constructed a car-mounted array using the digiBASE system. The array included a neutron detection unit and several sodium iodide gamma ray detectors, as well as a laptop for storage and processing. The entire system was able to fit into a roof-mounted car cargo container. The system correlated data with GPS information to determine the location of detected radiation.[8]

The project at hand involves placing radiation detection equipment on the spreader bar of a seaport crane. While there is at least one commercial system, produced by VeriTainer, that provides such a function[9, 10], the data regarding background, minimum detectable activity, and common operating events is unavailable in published literature. The VeriTainer system is based on an array of eight sodium iodide radiation detectors built into a spreader bar. While the VeriSpreader is currently in use, the

company has not released operational data. Veritainer conducted several tests in ports with their own design, similar to the Tacoma field test of the CraneWow system. In addition to gamma detection, the VeriSpreader system also employs He-3 tubes as neutron detectors. Due to the focus on the digiBASE-E, neutron detection is beyond the scope of this project.[9-11]

The cargo containers coming into the Port of Tacoma (PoT) are currently scanned by large radiation monitors attached to trucks, which are deployed near the unloading operation. When the cargo containers are first removed from the ship, they are placed on trucks, which queue up in front of the large portal monitors. The trucks drive slowly through the monitors, which pick up changes in count rate versus the background. If the alarm is tripped, the offending cargo container is pulled aside for a more thorough test using handheld detectors.[12] This process, unfortunately, causes a fair amount of delay in the processing, sorting, and eventual shipping out of the cargo containers. The ability to perform this initial scan without the artificial slowdown of the queue and the scanning process would be invaluable to seaports in terms of efficiency and cost-effectiveness. Even if the scan during crane operations required a short pause, removing a bottleneck from the entire offloading procedure would have a worthwhile effect.

NARRATIVE

Origins

CraneWow is based on software provided in ORTEC's programming examples, and in the software development kit (SDK), called the ORTEC Programmer's Toolkit.[13] The project was first received in the summer of 2010. The provided code was packaged into one Visual Studio "solution" (a collection of code) called "ListModeNew". This would become the principal reference when working with the digiBASE-E; much of the code section that pulls out the raw data and the data file format is based on ListPro, included in the ListModeNew solution. Other than ListPro, which collects list mode data as it streams in, the solution also contains ListHisto and ListDump. ListDump takes a data file and produces a text file transcription in its entirety. ListHisto is designed to create text histograms for arbitrary time segments from a list mode data (DAT) file, but it does not function. This is apparently due to a bug in the program regarding the start and finish times for the data run. The program is unable to accept boundaries for the histograms as it believes all DAT files are only a few microseconds long.

Initially, it was not possible to compile and test these programs as several key files were missing from the supplied files, including both the ListModeNew solution and the SDK. RecTypes2.h is a file that simply contains a number of data types and "structs" (logical organizational structures for data) used by ORTEC to deal with its own data files. Mcbcio32.h was a header file for some of the procedures used to interact with

the digiBASE-E. After consulting with ORTEC about these files, they were provided by the ORTEC staff. This allowed development to begin.

The ORTEC Programmer's Toolkit came with many manuals, which were among the most useful parts of the package. The Toolkit also came with several example programs showing basic retrieval of data from the device, as well as a rudimentary interface for sending text commands directly to the device. It was this last program, `Diag.exe`, that was used as the basis for `CraneWow`. The main motivation for the use of `Diag.exe` as a starting point was that an unidentifiable quirk of Visual Studio 2008 prevented proper addition of the ORTEC controls to a fresh project. The instructions for starting a fresh project were provided for an older version of Visual C++, and did not translate well to the newer Visual Studio user interface. Furthermore, the controls simply would not show up. While a more direct, manual approach may have allowed their eventual use, modifying an existing project was most expedient.

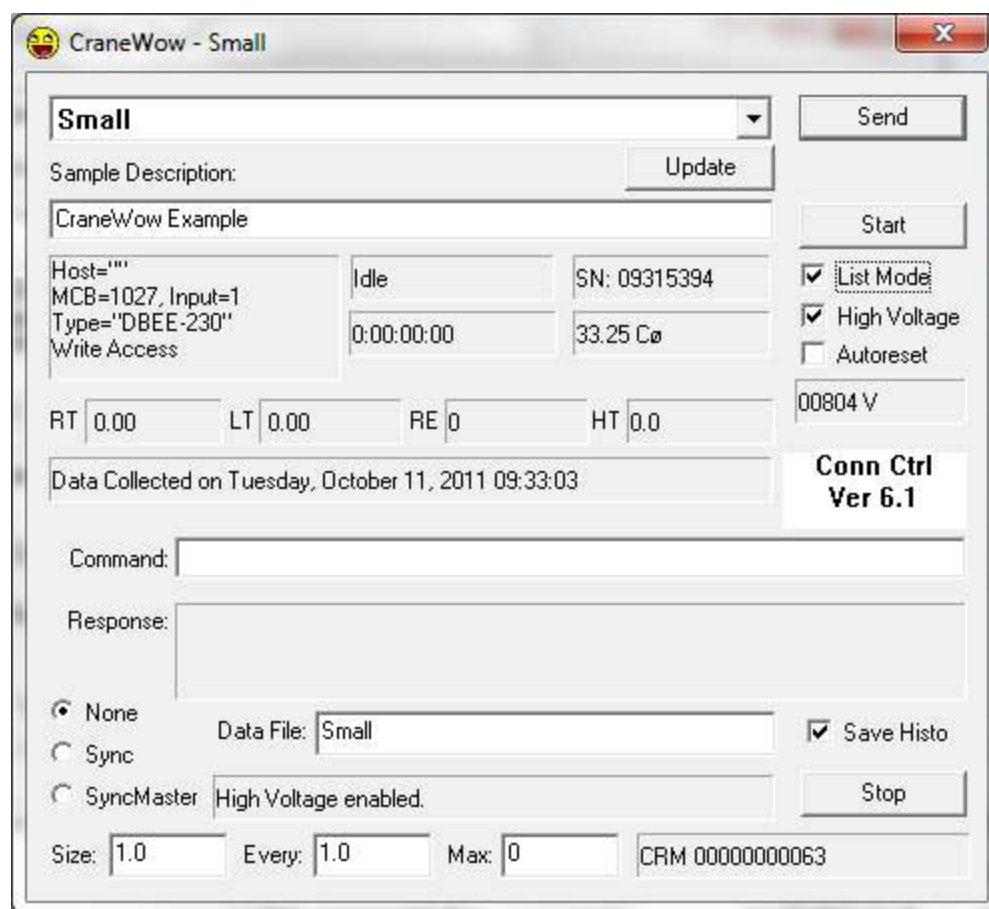


Figure 3 - Example CraneWow window for a digiBASE-E named "Small".

Diag.exe already contained the Conn and Drop controls, which were necessary and sufficient for the proper use of the future CraneWow. These controls allow for selection of and communication with each digiBASE-E by the program. Diag.exe had no method of obtaining or storing data from the device, but by sending text commands through the Conn interface, the device could be configured and started. To make the program more user-friendly, buttons and checkboxes had to be added to automatically send the correct text commands. Text displays, automatically parsing responses from the device into a human-readable format, were also added. These displays were set to

update periodically (one second intervals) during operation. As the project and testing progressed, new buttons and displays had to be added (Figure 3).

Software, Hardware, and Documentation Difficulties

The example codes provided by ORTEC were invaluable in trying to program the digiBASE-E data collection program. However, the ListModeNew collection of programs, while modified to accommodate each new ORTEC product capable of list mode, was unfortunately not completely updated for the digiBASE-E; many procedures and functions used in the case of a digiBASE-E were in fact written for the digiBASE. While the two devices are similar, there are some differences, most importantly in a few bits of the data stream, allowing for live time tracking. The digiBASE-E manual was not very clear or detailed on the subject, and was especially vague on what time scales were being used for different time stamps within the data stream. The readme files included with ListModeNew were much more detailed, but referenced the wrong product. Furthermore, the aforementioned problem of the programs themselves using digiBASE procedures to process digiBASE-E data further compounded the difficulty.

In the end, the information in the manual was verified as being correct, but much time and energy was lost attempting to reconcile these two sources and to fill in the gaps in information left by the manual. It further made testing more difficult, as there was no clear, properly-functioning “baseline” with which to compare. Eventually, saved data formatting had to be verified manually via a hex editor, examining the data files bit by bit to ensure that they were properly recorded. The data itself was verified by on-the-fly

text histograms, with the content of the saved data verified much later once a post-processing solution was written.

Originally, an on-the-fly visualization, using the same data aggregated in the on-the-fly text histograms, was planned. This was intended to provide a more immediate picture of what was going on with the detectors; the user could then notice problems with calibration, sudden spikes in particular energy regions, and the like. This feature was eventually scrapped as the deadline drew near for the program's use. The main difficulty in implementing the feature lay with the original Diag.exe program's use of Microsoft Foundation Classes (MFC) to provide the user interface. The author's personal experience in UI design involved mostly Java and its provided packages. While MFC is a common choice for providing Windows interfaces, learning the ORTEC COM objects and the quirks of the digiBASE-E itself were of higher priority. The original design used a window with a "Static Text" object as a baseline, drawing thin bars on it to create the histogram. The unsolved problem as it was last left involved a "device context" error. After some research, it seems that the bars of the histogram were attempting to draw to the wrong layer of abstraction in the windowing scheme. A future version of the program could attempt to recreate the histogram window properly and provide a visualization of the histogram as it is passed to the MakeHistoTxt function.

An issue that came up in later testing appeared when we tried to implement the synchronization mode available for the digiBASE-E list mode. This mode inserts an extra word into the data stream every 100 ms when a sync pulse is generated by one of the devices. This device is designated as the sync master. As described in the manual,

the devices must be connected via the SMA coaxial input and output ports in a “daisy chain”. Unfortunately, when the mode was implemented, one or more of the digiBASE-Es would not capture data. After thoroughly checking the manual, it was feared that a buried programming error was causing the problem. After consulting with ORTEC on the issue, the solution was simply to connect the last in the line of “slave” devices to the input port of the “master”. This requirement is not mentioned in the manuals nor is it suggested by the description of sync mode functionality. Once this had been done, sync mode performed as expected. Of note is the fact that this occurs even in PHA mode, where the synchronization gating setting should be ignored.

The sync mode was used by our project to ensure a simultaneous beginning point for data collection among all devices. While the mode would likely prove useful in coincidence experiments and the like, the method for correcting time stamps based on synchronization information is not outlined in the manual. The only example provided is in the example programs, in which the comments are brief. As the uses for CraneWow required accuracy only within one second, the extended use of the mode was discarded.

A difficulty that presented itself during initial testing of the system was that of data throughput limitations. When count rates exceeded roughly 15,000 counts per second in a single detector, CraneWow would eventually cease to receive data from the digiBASE-E. This would frequently also result in a crash of the entire array. This results in the CraneWow program running without receiving data from the digiBASE-Es. The rate at which this happened was lower when several digiBASE-Es were

involved, and once one or two in the array had crashed, the others might run without incident for a long time. This would only occur when using list mode. After ensuring that there were no memory leaks within CraneWow, and that the same problem occurred with some of the example software, ORTEC was contacted about the issue. The first solution offered was a patch to the ORTEC MCB Server that runs in the background. This seemed to improve the situation but did not alleviate it.

Upon further consultation, ORTEC suggested that the problem might lie in CraneWow's use of a timer event to gather data, rather than a dedicated worker thread. Threading can allow a program to essentially "think about more than one thing at once." It was suggested that the first-in-first-out memory (FIFO) that stores the data stream until retrieved was filling up, causing the digiBASE-E to crash. Unfortunately, the COM objects used to interface between CraneWow and the MCB Server were nonthreadable, and a major rewrite of the program would be required to use the more direct DLL (Dynamic Link Library) functions. For this reason and due to time constraints, threading was never implemented. A further analysis of the problem was conducted after the Tacoma operation, the results of which are discussed in a later section of this document.

Though the count rates involved were much higher than expected in the operational environment of a working port crane spreader bar detector system, a workaround was devised to ensure that the problem did not interrupt data gathering during operation. The workaround detects a crash by assuming that an empty data stream indicates a FIFO overflow. If enabled, the program will stop, clear, and restart the

digiBASE-E, as well as reset its real and live time clocks to appropriate values. The workaround, while enabled, was never invoked during the field operation in Tacoma.

Final Steps

The final step in having a working system was to produce programs or scripts to convert the raw data files into human-usable formats. The programming and scripting language Perl is a language designed for running through large, linear files and outputting the results as text.[14, 15] After Perl scripts were used to create comma-delimited files (.csv), Excel could then be used to read, analyze, and display the data. The first script (ticker.pl) simply created count rate timelines, while the second (histograms.pl) would create histograms from arbitrary spans of time covered by the data file. The output of these scripts was compared to the original on-the-fly text histograms for verification. Eventually the creation of these on-the-fly text histograms was changed into a disabled-by-default option of CraneWow.

While this set of scripts might have been technically sufficient for the analysis of the data, Excel is not designed with isotope identification in mind and is rather clumsy to use with large numbers of files. The N42 format is an XML-based data format standard defined by ANSI.[16] XML-based files are essentially text files, conforming to a particular structure. Being able to modify and expand upon the existing Perl scripts saved time in creating the conversion script. N42 also has a similar capability as the list mode DAT files, in that a timeline of the radiation environment is recorded. However, unlike the DAT files, N42 does not assign a specific stamp to each event. Instead, the

file consists of many histograms of some length with consecutive start times. (For this project, an interval of one second was used.)

An advantage of the N42 format is that one file can contain the information from multiple detectors. While this requires some additional processing in the conversion script to ensure the times are lined up correctly, it is an advantage when comparing the information gathered from different detectors (in different locations) at the same time.

DIGIBASE-E SHORTFALLS

Throughput

The digiBASE-E is designed to transmit data during list mode by using the 2048 channels normally used in pulse-height analysis mode. Each of these channels is equivalent to a specific range of energy deposited by photons incident upon the scintillation crystal. These channels are four bytes long and contain the channel data. Upon a pull, when the program asks the digiBASE-E for its data, the first channel will contain the number of subsequent channels containing valid data, with the remaining 2047 available to transmit data out of the buffer. The buffer on the digiBASE-E is a first-in-first-out memory bank (FIFO) that fills as the device experiences radiation interaction events and generates timestamps.

If the device is not polled often enough for data, this FIFO buffer can fill up with data. This will cause the digiBASE-E to crash, resulting in empty responses to requests for data. A timer or threaded loop must be used to continually pull data from the digiBASE-E at a rate faster than it is expected to be generated (CraneWow uses the timer method, pulling data once per specified interval). The FIFO overflow failures will occur with even a single digiBASE-E at around thirty thousand counts per second. This failure will occur even with the simplest of programs, which is just concerned with taking the raw data and saving it to a file. While some disk writing overhead may be involved, SSD technology vastly mitigates this problem.

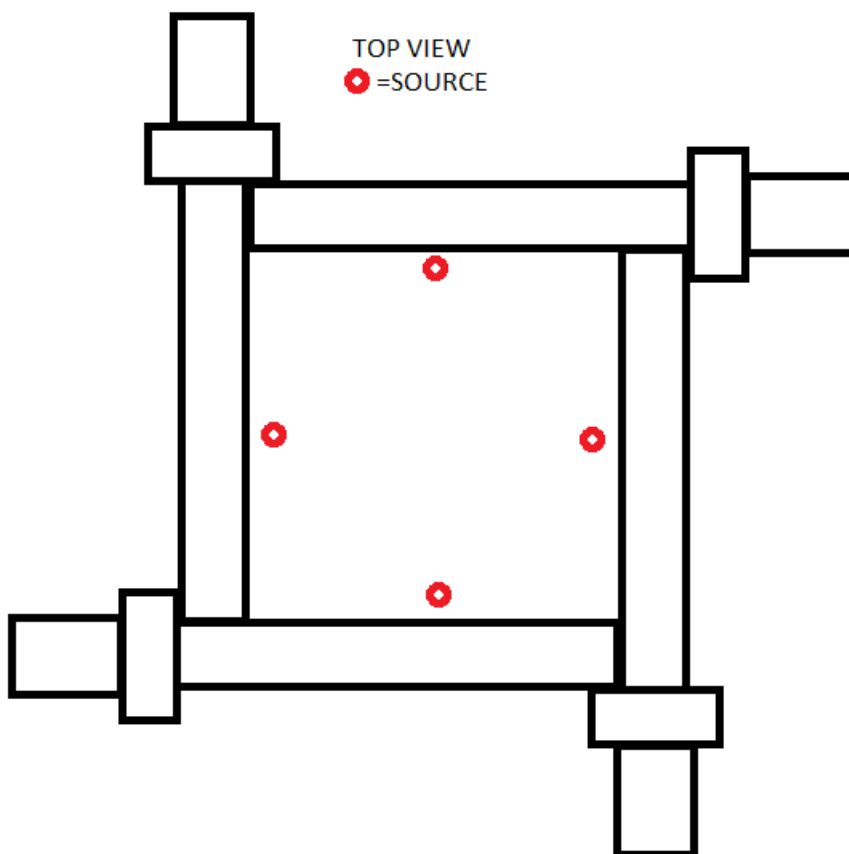


Figure 4 - Test procedure diagram for "midline" configuration.

To examine the FIFO overflow problem, a series of tests were conducted using the digiBASE-Es and test sources. These tests employed four 2x2x16 NaI (Tl) crystal detectors arranged in a square to maximize consistency. Sources were placed at the center of the square or at the midline of the square edges (Figure 4), again to ensure consistency. The sources used were 5 μ Ci Cs-137 sources, as these are a common calibration source and were readily available. The detectors were calibrated to give a similar response in terms of count rate.

Each test focused on a particular count rate. Three tests were run, noting the total events processed and the “HT” value once a crash occurred. The HT value indicates the last timestamp processed by the computer, and gives an indication of the maximum amount of time for which reliable data can be recorded. Once these three runs were completed, one detector would be removed from CraneWow’s watch and the tests would be run again. If a test lasted for longer than thirty minutes, the detectors were assumed to never crash. This was decided to be long enough for the purposes of this test due to less formalized testing experience with the array in the lab.

For the tests with fewer than four detectors, the unused detectors were left in place for consistent backscatter results. Furthermore, the detectors were left with the high voltage on so that subsequent tests would begin with all detectors having similar thermal conditions. It is worth noting that the detector “Righty” has been confirmed to have other hardware issues. For this reason, “Righty” was the first detector to be removed in each test sequence.

Table 1 - Throughput crash test data for digiBASE-E.

Avg. Count Rate (counts/second)	4 detectors	3 detectors	2 detectors	1 detector
31,000	7.033333 s	11.75556 s	273.0875 s	946.1 s
HT Events	219425.9	358399.4	8296545	28340975

Table 1 - Continued

20,000	19.575 s	No crash	No crash	No crash
HT				
Events	394862.1			
15,000	No crash	No crash	No crash	No crash
HT				
Events				

The tests reveal that rates of over twenty thousand counts per second would reliably crash the detectors, causing the digiBASE-E to cease reporting data, usually in less than six minutes. Having more than one detector collecting would crash the system sooner, with crashes occurring in as little as six to seven seconds for all four detectors at just over thirty thousand counts per second. A simple back-of-the-envelope calculation can be made to determine the theoretical maximum event rate possible for a particular poll rate. A setting of 20 ms between each data pull, for example, should allow for fifty polls per second, or 102,350 events per second. Minus the 100 real time stamps, 100 live time stamps, and 10 sync stamps, this translates to 102,240 counts per second per detector as a theoretical maximum. A timer setting of three ms should allow for just over 682,000 counts per second, and so on.

This is not reflected in the test results. Indeed, they are an order of magnitude lower. What could be causing this discrepancy? Data transfer overhead should be

negligible, and the removal of extra networking equipment in the case of a single digiBASE-E eliminates much of this. Data gathering is done in the same manner as the supplied example ListPro program, which itself exhibits this behavior. Something inherent to the process, whether in the digiBASE-E itself, the networking overhead of the computer itself, or the MCB Server, causes a significant enough delay that the program can only effectively poll roughly 15 times per second.

The results do suggest that a threaded approach would improve results when using multiple digiBASE-Es, but only when exceeding count rates of fifteen thousand per second. However, based on the results for single detectors, threading is unlikely to do much to help beyond thirty thousand counts per second. Threading would only cause the program to treat each digiBASE-E individually, and furthermore does not address the possible issues with the MCB Server, which must handle all requests for digiBASE-E interaction. These results indicate that there is a bottleneck somewhere in the digiBASE-E or in the MCB Server program that functions as an intermediary between CraneWow and the array.

Documentation and Examples

A significant amount of confusion came from the various sources of documentation on the digiBASE-E and its operation. While the manual, in the end, proved to be the most accurate, it omits or glosses over explanations of some of the functions, especially when it comes to list mode. List mode is given two pages of appendix in the back of the manual, as well as half a page in the “Gating” section, which is mostly concerned with telling the reader to look elsewhere for information. When

contacted directly, the ORTEC personnel were, without exception, extremely helpful, and once the problem was articulated, they offered reasonable solutions or explanations. However, the brevity of the manual on the subject of list mode made forming those questions difficult.

The example programs provided by ORTEC, which proved to be invaluable in getting the project started, had more detailed explanations of concepts and procedures in comments and readme files. However, this expanded detail turned out to be inaccurate in the case of the digiBASE-E. The programs of the ListModeNew package were clearly not written with the digiBASE-E in mind, with some functions for the digiBASE-E added as the product was released, while others remained incompatible. Some program features that were not updated instead shunt digiBASE-E through incompatible digiBASE functions. The documentation was unfortunately not updated for the new product, and some of the minor differences between the digiBASE-E and the older digiBASE caused confusion and lost time.

A prime example is that of the ListHisto program, included in the ListModeNew solution. This program was intended to have functionality eventually recreated by the Perl scripts written for this project, specifically creating histograms from the DAT files. Several of the routines intended to pull information from the DAT files detect which list mode product is being used, and send the data to be processed by the appropriate function. While some are sent to functions written specifically for the digiBASE-E, other times the data is sent through a portion written for the digiBASE. This causes a problem specifically in the data word reading process, as there are a few differences in

the bit structure of the data between the digiBASE-E and the digiBASE; most notably that there is an extra flag bit in the digiBASE-E data to accommodate two extra word types. This is likely some of the cause of the ListHisto program not working for digiBASE-E DAT files.

A compounding issue blocking the implementation of threading was the lack of explanation regarding the example code and its incompatibility with threading. The digiBASE-E and programming manuals spend roughly even time emphasizing the COM controls and the DLL interface for ORTEC detector equipment. COM (Component Object Model) is a Microsoft programming technology designed to further encapsulate the various components used by programmers, allowing the use of “COM objects” written in various programming languages in the same project. In this way, a particular component can be written in a language that is most expedient for that component, and the COM objects can be used for programs that need to be written in different languages. The other method to interface with the devices is through more direct calls to a DLL. Also a Microsoft technology, DLLs allow for a library of functions used for a particular task, such as rendering graphics or interfacing with devices, to be compiled into a single file and used by many programs. These libraries can be linked “statically”, where the DLL file must be included with the program and placed in a specific location; or “dynamically”, where the DLL file is kept by the operating system and is found at run-time. Dynamic linking is the preferred method as it allows easier sharing of common DLLs among programs and easier updating as the DLLs are improved.

While both technologies are essentially designed to accomplish the same thing, DLLs are more “technical”, allowing for more direct control, while COM objects are more “friendly”, sacrificing some efficiency. The ListModeNew set of programs, which serves as the primary example and documentation for list mode, uses COM objects. This approach simplifies attaching to the device and getting the raw data. However, the major drawback is that ORTEC’s COM objects are not threadable. Unfortunately, this is not documented and had to be discovered with trial and error. This was only discovered when threading was proposed as a possible solution to the throughput errors discussed above. When an example was requested from the vendor, a small program using the mcbcio32.dll library was sent. Unfortunately, the program did not make use of list mode, and the details of using list mode with the DLL would have to be ferreted out. Ultimately, it was decided to scrap the threading modification altogether. This was decided in light of time constraints and low projected benefit given the limits on single detector effective poll rate.

PROGRAM OVERVIEW

Once the digiBASE-Es are connected to a controlling computer, CraneWow and the associated Perl scripts provide basic set-up and data processing for an array of digiBASE-Es. What follows is an overview of how to use the programs and what their functions are.

CraneWow

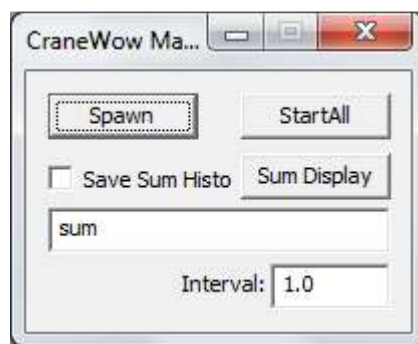


Figure 5 - CraneWow Master Box.

Upon running CraneWow, the user will see a small window with three options (Figure 5). Selecting the check box to “save sum histo” will cause summed histograms of all detectors to be saved on-the-fly, in addition to any individual on-the-fly histograms. Selecting “spawn” will spawn a main CraneWow detector window.

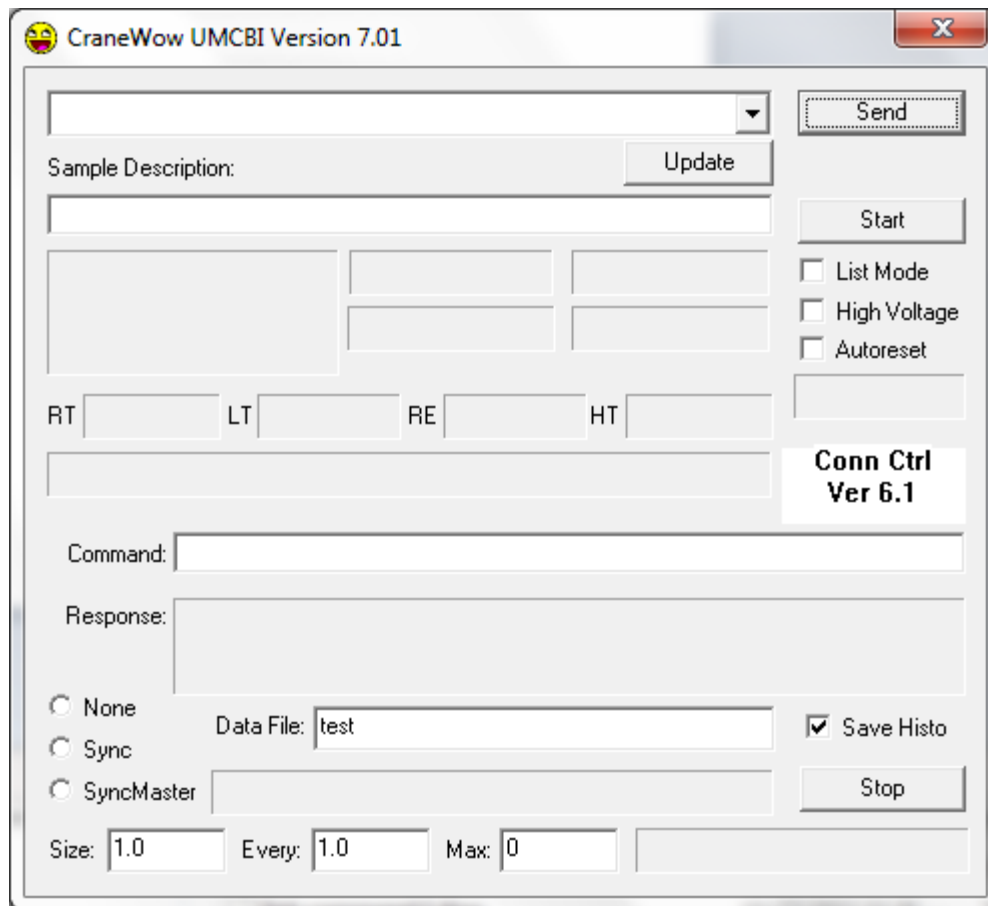


Figure 6 - CraneWow Detector Window, unselected.

There are many options and indicators in the CraneWow detector window (Figure 6). The first step is to choose a detector from the drop-down at the top. The user will see in this drop-down a full detector list identical to the one recently created with MCB Configuration. Once a detector has been selected, information including the serial number and status will appear in the window (Figure 7).

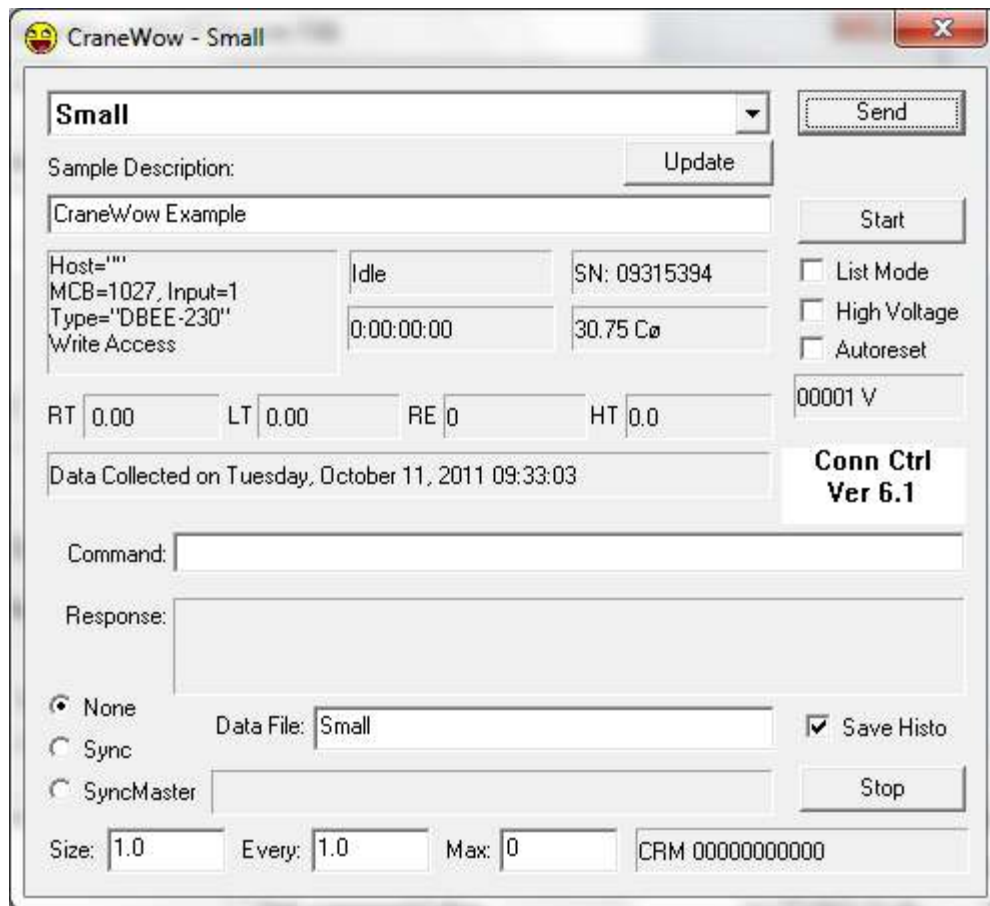


Figure 7 - CraneWow Detector Window, selected.

Figure 7 shows the detector window for a digiBASE-E titled “Small”. The list mode and high voltage are both off, as indicated by the check boxes to the right. One must make sure the detector is in list mode and has high voltage turned ON with the associated check boxes. The program will automatically suggest a file name beginner based on the detector name. If another filename is desired, it can be entered in the box given. The description for the data being written can be edited by changing the “Sample Description” and hitting SEND. Note that when this is done, any commands typed into the command window will also be sent. A selection of these commands is included in

the CraneWow User Manual, and a complete list is available in the ORTEC digiBASE-E manual.[6]

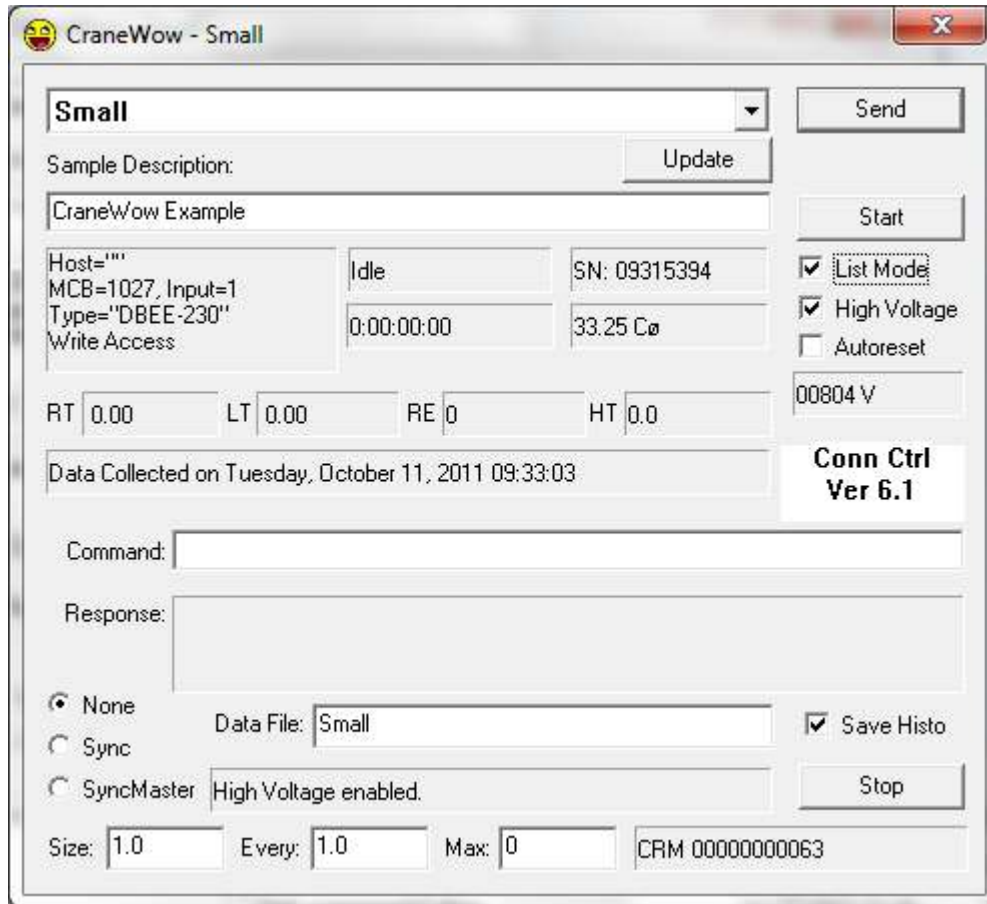


Figure 8 - CraneWow Detector Dialog, List Mode and High Voltage engaged.

If Sync gating is desired, selecting an option other than NONE from the radio buttons on the bottom left side will enable it. More than one SyncMaster will cause uncertain synchronization and possible loss of data as the digiBASE-Es may not start correctly.

CraneWow can generate text histograms on-the-fly for immediate viewing and verification of data. The length and frequency of these histograms can be adjusted via

the “size” and “every” fields, respectively, with the example in Figure 8 recording one-second-long histograms each second. Note that uncertain timing will result from settings requiring more than one overlapping histogram at a time. Thus, the user should not have “size” set to less than half the value of “every” before beginning. If no on-the-fly histograms are desired, the user can uncheck the “Save Histo” box at the bottom right. “Max” will set the number of seconds (as measured by the computer) to gather data. The default of “0” will have the program collect data until manually stopped via the “Stop” button in the bottom right corner. If a high count rate is expected, the “allow resets” check box can be checked after starting data acquisition. This will allow recovery if the digiBASE-E encounters a higher throughput than it can handle.

Once all settings have been made, clicking the “spawn” button again on the master box will spawn a new detector window, allowing the user to configure the next detector. Once all detectors are set, they may be started one-by-one via the “start” button on each window, or by the “StartAll” button on the master dialog.

The realtime and livetime are displayed in the “RT” and “LT” boxes, respectively. These can be compared to determine the dead time of the detector as it runs. The “HT” box, standing for “Histogram Time,” can be found to the right. This box should be very close to the realtime as displayed. The RT and LT boxes are updated through direct query of the digiBASE-E. However, the “HT” box is updated as the data stream is retrieved, directly from the realtime time stamps as they are read and interpreted. If this value lags behind significantly (more than one half of one second), this can indicate an imminent digiBASE-E crash. The board temperature is displayed, in

Celsius, in the BT box. The maximum operating temperature of the digiBASE-E, according to the manufacturer, is 60 °C ambient (though the board temperature may be somewhat higher than this). This field was added to check the health of the device in hot weather, with the intention of stopping collection and turning off high voltage if the values became too high.

Perl Post-Processing Scripts

A number of scripts are provided with CraneWow to facilitate post-processing of the list mode data. For ease of use, these scripts should be copied into and run from the folder containing the DAT file, but they can technically be run from any location. These should be run using the command line, as they require arguments.

ticker.pl

USAGE: perl ticker.pl input.dat output.csv

The ticker.pl script will generate a second-by-second count rate data stream and deposit it in a .csv file. This file can be opened by Notepad or Excel, and contains ordered pairs of “time,count rate” in counts per second. This information can be used to identify times when count rates increased or were depressed due to the detection environment. Once these time spans are located, histograms can be generated using histogram.pl.

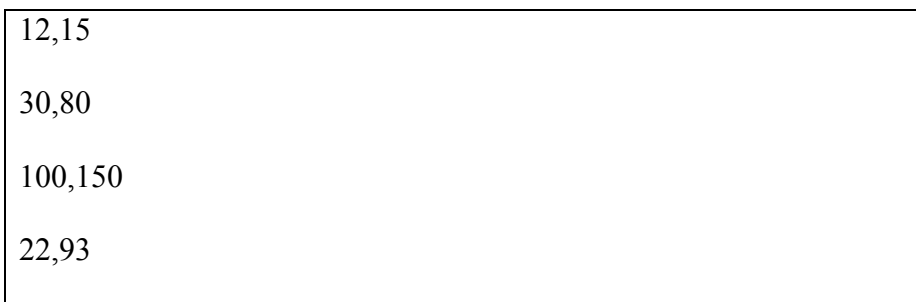
histogram.pl

USAGE: perl histogram.pl times.csv input.dat output

This script can be used to extract arbitrary time spans of spectral data from a .dat file. Prior to running, a text file must be generated with each line being “beginning,end”

in seconds for each histogram desired. In this way, the script can be run once to extract a large number of “interesting” spectra for use in Excel or any other program capable of reading comma-delineated data.

For example, if one wished to extract histograms from three different time spans from a single DAT file, one would create a text file in the format of Figure 9.



```
12,15
30,80
100,150
22,93
```

Figure 9 - Example histogram time spans file.

If this file is saved as `interesting.txt`, with the DAT file named `Monday.dat`, one would copy these files and the scripts into the same folder, then run the following command:

```
perl histogram.pl interesting.txt Monday.dat MondayHisto
```

This would create four histograms, named in the manner `MondayHisto(12,15).csv`. These could be opened in Excel or a text editor, and analyzed as needed.

dat2n42.pl

USAGE: `perl dat2n42.pl input1.dat input2.dat ...`

The N42 format is an XML-based data format standard defined by ANSII. While it does not have the capability for true list mode, it can accommodate a series of

time stamped, short duration spectra. Spectrum analysis programs are able to interpret these files, calculating count rates and summing across time spans. Another advantage is that all detectors' data is contained within the same file as separate spectra.

The `dat2n42.pl` script will take, as arguments, all the `.dat` files one wishes to combine. It is advisable to only include `.dat` files that are of the same general time span, and have different file names. However, time spans of any arbitrary length may be used. The “detectors” will be named as the file names.

Note that since N42 is XML, the files are text. This means that files can be quite large (uncompressed). A four-hour, three-detector file, for example, runs about 12 MB on disk. The script uses 1 second intervals as standard, but this can be changed within the script via the `$interval` variable. While running, the script will display the latest interval number recorded (by default, this is equivalent to seconds). The script output will have the same file name for each run. Thus, it is important to rename or move the output (`PleaseRenameMe.n42`) prior to running another conversion.

PROOF OF CONCEPT

Overview

The project that this software was initially written for involved the study of minimum detectable activity for a detector system attached to the spreader bar of a seaport crane. The cranes used to offload large container ships are fitted with spreader bars, used to lift the containers. These large devices are able to resize to the different container dimensions. The spreader bars affix to latch points on the containers and lift them from the ship, placing them on trucks to be sent throughout the yard. As the spreader bar is in contact with the container for several seconds, it represents a possible scanning point for radioactive contraband. Our group traveled to the Port of Tacoma, putting several detectors onto a spreader bar at Husky Terminal as it unloaded containers from a docked ship.

The detectors were packed into “pods”, mounted to the top and inside of the spreader bar. Included in each pod was a generous amount of potassium chloride, having the advantage of supplying a potassium-40 peak. This peak, due to the naturally-occurring potassium-40, gave a reference point to the gain stabilization function of the digiBASE-E, which was crucial over the long collection times of the operation. Gain stabilization allowed the avoidance of spectral drift that can occur over long periods. Three of these detector pods were attached to a spreader bar on a crane at Husky Terminal at the Port of Tacoma.

Results

The first day in Tacoma, a background test was taken, as well as tests moving the crane up and down, as well as over the water. Figure 10 shows the count rate for three detectors attached to the spreader bar of the port crane.

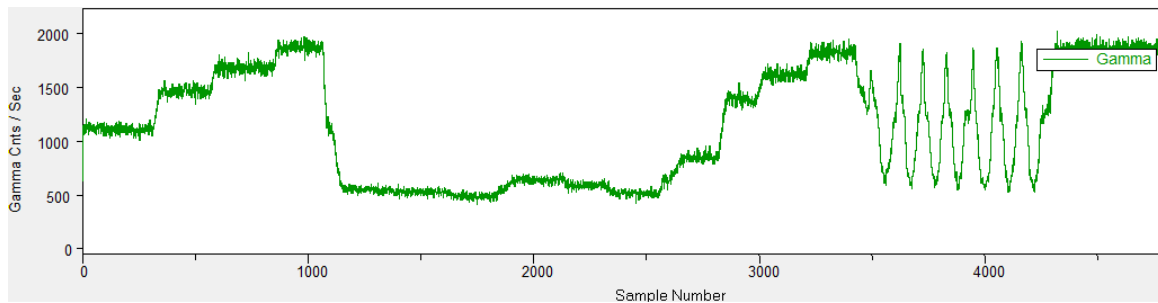


Figure 10 - Sample count-rate graph from Tacoma trip.

There are clear steps visible where the spreader bar was moved from position to position. The “sawtooth” pattern in the 3500-4300 region is a series of simulated cargo grabs, with the spreader bar moved in an arc over the water, then back over the asphalt beneath the crane. Count rates were much higher over the asphalt, and much lower over the open water. The “steplike” regions from 0-1000 and 2600-3500 correspond to leaving the spreader bar at particular heights over the asphalt for some time; a similar pattern over the open water can be seen from 1800-2600. A small sample of a potassium salt table salt substitute, included to aid gain stabilization of the system, provided another interesting data example as shown in Figure 11.

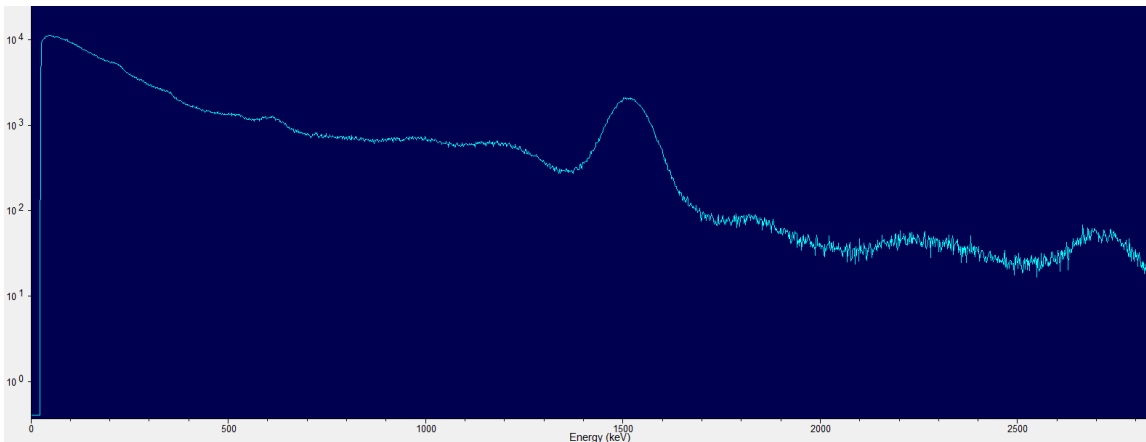


Figure 11 - Sample radiation spectrum from Tacoma trip.

The spectrum in Figure 11, roughly a half hour long, was pulled from the data of a single detector for a time when the crane was over the water. Clearly visible in the center of the spectrum is the peak from the naturally-occurring potassium-40 included with our potassium salt. This peak also served as a handy reference point for gain stabilization on the crystals. Frequently, the K-40 peak can be seen in the background spectrum, from natural concentrations of potassium in the ground. However, the equipment, ship, and containers might have prevented seeing it at times during the operation. This would compromise the gain stabilization function of the digiBASE-E, which avoids spectral drift by automatically adjusting the gain in order to keep a particular peak centered around the same channels.

CONCLUSION

The original objective of this project was to create a program to run the ORTEC digiBASE-Es in list mode. The program needed to perform basic set up of the devices, including putting them into list mode and handling the high voltage settings. The program also needed to provide status updates as to the live time of the current acquisition, the progress of the acquisition, and to provide some way to verify correct operation. CraneWow provides these things through its main window interface, with easy to use check boxes and status texts. The optional text histograms that can be created during operation allow for a quick check to see that the spectrum being acquired is reasonable for the situation, while the more complete DAT file being generated provides the data required for a complete analysis. The Perl scripts developed allow that DAT file to be processed into more human-readable data.

The software was tested in the field at the Port of Tacoma, and the data gathered is being analyzed as a PhD project. This analysis should provide some insight into the minimum detectable activities reasonably achievable with a crane-mounted cargo scanner. The post-processing scripts, written in Perl, are aiding this analysis by collating data from the raw data files, creating count rate information and spectral data in human-readable formats.

Much has been learned about the process of creating software and the process of conducting research from this project. The difficulties associated with learning a new system, even when the languages involved are known, can easily overwhelm the process, causing delays. The software can now be used to collect data for other projects. The

quick setup of the array into any configuration allows it to be used for many passive scanning applications. One possibility is to create portal monitors, which are used to scan objects and creatures at a chokepoint. A future project in the works is an agricultural portal monitor; a model of a system that could possibly be used to inspect livestock after some radiological event. The livestock would be herded through a chute and radioactive contamination could be assessed. If the system itself is portable and flexible enough, an event such as Fukushima would not result in such large-scale loss of valuable food resources. The work presented here successfully resulted in a software package that can be used for further projects.

REFERENCES

- [1] A.K. Hara, C.D. Johnson, R.L. MacCarty, T.J. Welch, C.H. McCollough, W.S. Harmsen, CT Colonography: Single-versus Multi-Detector Row Imaging, *Radiology*, 219 (2001) 461.
- [2] ORTEC Homepage, Available from <<http://www.ortec-online.com/>>, accessed 2011.
- [3] M. Grypp, Tacoma Spreader Bar, 2011.
- [4] F.H. Attix, Introduction to radiological physics and radiation dosimetry, Wiley, New York, 1986.
- [5] G.F. Knoll, Radiation detection and measurement, 4th ed., John Wiley, Hoboken, N.J., 2010.
- [6] digiBASE-E User's Manual, ORTEC 2009.
- [7] E. Reber, L. Blackwood, A. Edwards, J. Jewell, K. Rohde, E. Seabury, J. Klinger, Idaho explosives detection system, *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 241 (2005) 738-742.
- [8] D. Upp, R. Keyser, Performance of a car-mounted neutron and gamma-ray monitoring system for illicit material detection, INMM 45th Annual Meeting 2009.
- [9] VeriTainer (web site), VeriTainer, Available from <www.veritainer.com>, accessed 2011.
- [10] R. Redus, M. Alioto, D. Sperry, T. Pantazis, VeriTainer radiation detector for intermodal shipping containers, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 579 (2007) 384-387.
- [11] VeriTainer, System Capability Analysis, 2009, available from <<http://www.veritainer.com/images/VeriSpreaderCraneMountedSolution.pdf>>, accessed 2011.
- [12] Customs and Border Patrol, Available from <cbp.gov>, accessed 2011.
- [13] ORTEC Connections Programmer's Toolkit, ORTEC, 2004.
- [14] Perl, Available from <<http://www.perl.org/>>, accessed 2011.
- [15] L. Wall, T. Christiansen, J. Orwant, Programming Perl, 3rd ed., O'Reilly, Cambridge Mass., 2000.

- [16] ANSI N42.42: Data format standard for radiation detectors used for Homeland Security, NIST, Available from <http://physics.nist.gov/Divisions/Div846/Gp4/ANSIN4242/xml.html>, accessed 2011.

APPENDIX A DEFINITIONS

Word

A “word” in the programming sense is, essentially, a packet of binary data. It is a logical chunk of information that contains something useful, such as a character, a number, or a bunch of flags and switches. Words are frequently some number of bytes large, where a byte is eight bits (a bit being a 1 or 0). In the case of the digiBASE-E’s data stream, each word is four bytes long.

Each channel in the traditional “radiation spectrum” sense is sent to the computer as a word. The order the words are sent tells the computer which channel it is, while the number inside the word tells the computer how many counts have been recorded in that energy bin.

In the case of list mode, these same words are used, but have different data inside. The zeroth word tells the computer how many subsequent words contain useful data. Then, that many words afterward each have either a regular timestamp (to tell the computer the real and live time) or a channel number with a smaller, more precise timestamp, indicating a count in that energy bin.

Raw Data

The data words coming in from the digiBASE-E in list mode make up a stream of “raw data” that is written to a file. Other than a short header containing information such as the start data and calibration coefficients, the data file consists entirely of this raw data stream, sequentially recorded as it comes from the device. This data consists of the four-byte words, of which there are four types:

The sync time stamps contain a count of sync pulses as they come in from the sync master device. These pulses are generated ten times per second, and the count resets every eight minutes. The word also contains a smaller “prescale” time stamp (in 20 ns ticks), which is added to the latest real time stamp to obtain precise timing information for the sync pulse.

The live time stamps arrive just after each real time stamp, and contain only the live time given in 10 ms ticks. This means that for a live time stamp of one hundred, one minute of live time has passed. Comparing this value to the preceding real time stamp can give the dead time and live time percentage for the detector.

The real time stamps arrive each 10 ms, incrementing by one each time. This is the word to which all others are compared in order to generate important information such as live time percentage and precise timing information. Sync and ADC words contain smaller time stamps given in 20 ns ticks that must be added to this one to obtain their precise time.

The ADC words contain a channel number for the detection event. They also contain the same type of prescale stamp as the sync time stamp word. It is these words that make up the spectral data when added up over a period of time.

When combined with the calibration data available in the header, a spectrum for a particular time period can be obtained by adding up the hits represented by the ADC words over a particular period of time.

File Formats

.dat

The DAT file format (for data) contains the raw data stream from the digiBASE-E after a small header. The header includes calibration data and a short comment, as well as information about the device and the start time of the acquisition. The size of these files will be roughly proportional to the count rate and acquisition time.

Excel, Maestro, and other programs cannot open the DAT files. The file must be processed before a more human-usable format can be obtained. This can be done with the included Perl scripts.

.csv

CSV (for comma-separated variable) is an extension used for a wide variety of related file formats. While many small variations exist depending on the needs of a particular project, the common thread is a text file, with the use of commas to separate related values, and newlines to separate entries. Having a known number of values per line in a known order can provide a basic “spreadsheet” file format that is easily opened by Excel. The Perl scripts included with CraneWow use this format for input and output.

.n42

N42 is a file format used by the U.S. government and commercial companies as a standard for radiation spectrum files. As this is an XML format, it is also text-based. XML (for Extensible Markup Language) is a set of guidelines for defining data entries and their contents, much like HTML does for web pages. The specifics of which entities are valid and what the data contained within might mean are handled by the program. One of the advantages of the N42 format is that it can approximate a list mode stream

with a series of short spectra. It can also contain several detectors' worth of data in one file. One of the Perl scripts can convert and collate several DAT files into one N42 file, with the only loss of information being combining every second's worth of detection events into spectra.

List Mode

List mode is a digiBASE-E feature which differs from pulse-height analysis mode in that every event is given its own data entry. Rather than adding up the hits in each energy bin, list mode leaves such processing for the program or user. Instead, the events seen by the detector are simply noted, along with time information. This data is kept as a continual stream, with everything gathered since the last pull transmitted upon request by the controlling computer.

PHA Mode

Pulse-height analysis is one of the most familiar sights for those working with radiation detection and identification. In this mode, the digiBASE-E keeps a long list of numbers, each corresponding to an energy bin. When data is acquired for a certain span of time, each of these channels will contain the number of hits for a particular energy range during the acquisition time.

Sync Mode

The digiBASE-E list mode has an option known as sync mode, used to help correct possible timer divergence among the devices. Each device is connected in a loop, and one device is nominated as the sync master. The sync master generates a pulse

every 100 ms, which is used by the digiBASE-Es as a signal to insert a sync time stamp into the data stream.

Real time

The real time is the amount of time the device has spent acquiring. In the case of an active acquisition, it is the time since acquisition began. In the case of a spectrum or data file, it is the total time spanned by the file.

Live time

Live time will always be shorter than real time for a particular acquisition or spectrum. This is due to the way in which detectors are “dead” for a short time after detecting an event. During the time in which the detector is dead, it cannot detect additional events. Thus, the live time is equal to the real time less the dead time.

TCP/IP

TCP/IP is the protocol suite used for most communications on the Internet today, as well as local, smaller networks. An IP address identifies a particular machine on the network.

DHCP

DHCP is a method for assigning IP addresses to machines on a network. When the machine first connects, it will send out a signal requesting an IP address from a DHCP server.

RJ-45 / 8P8C

This is the type of connector used by Ethernet connections. It resembles a phone plug, albeit wider. While the technical term is 8P8C, the term RJ-45 is more commonly

(mis)used to refer to this type of connector. This is the means by which the digiBASE-E gets its power supply and communicates with the controlling computer.

Power over Ethernet

Ethernet cabling has eight wires available for use. Not all of these must be used for data transmission. By using two of these wires to transmit power, extra cabling can be eliminated, reducing set-up time and complication. The digiBASE-E uses PoE exclusively as a power source.

Radiation

Particles emitted from atoms as part of a decay process are known as radiation. There are several types, including alpha, beta, gamma, and neutron radiation. For the purposes of this project, gamma radiation is the most relevant.

Gamma rays and X-rays are both types of photons (light) that are emitted from atoms as the result of energy level changes by subatomic particles. The only distinction lies in the origin of the photon: gamma rays come only from the nucleus as it de-excites. Photons are indirectly ionizing radiation, as they are not themselves charged. They must physically collide with something in order to cause secondary, ionized particles before they can be detected. In the practical case of radiation detection, these are electrons.

Scintillator

A scintillator, such as sodium iodide (doped with thallium), can be used to detect gamma radiation and can provide information about the energy of the photon detected. When combined with an appropriate dopant, sites with different energy states are added

to the possible sites electrons can occupy in the material. When struck with ionizing radiation, the electrons will jump to a higher site.

These electrons quickly decay back into their base energy state, re-releasing the energy as a collection of light that propagates through the crystal. This light is detected by electronics which amplify the signal into one receivable by a data collection device. The intensity of the light indicates the amount of energy originally dumped by the incoming radiation particle.

NaI(Tl) is a common scintillator, chosen for its relatively inexpensive and durable nature. Other detector materials include certain scintillator plastics, which are quite cheap but have low resolution, and more expensive and fragile semiconductor crystals such as high-purity germanium. These can offer higher resolution, but at the cost of the apparatus required to keep them running. Many such systems require a liquid nitrogen coolant system.

ADC Word

This is a term used to refer to the data words containing spectral information during list mode operation of the digiBASE-E. It comes from Analog to Digital Converter, which is a device that translates the analog signal from the radiation detector into a digital signal processable by a computer.

Spectrum

The photons being detected by the NaI crystals all have different energies. These energies are organized by the digiBASE-E into channels, or energy bins, each of which represents a small energy range. A radiation spectrum is generated by adding up the hits

in each energy bin over time. This information is then displayed as an energy vs counts graph, with thin bars or dots indicating the data points.

This type of plot is frequently called a “spectrum”, and is one of the most familiar presentations of radiation information. This information can be used to identify radioactive sources and help gauge their intensity. A peak in the data, centered on a particular energy bin, can indicate particular isotopes as their emissions are frequently exact and known. The count rate can be compared to a similar situation with a source of known strength, and the strength of the unknown source can be estimated.

Perl

Perl is a high-level programming language, frequently used for small scripts that parse large volumes of text data. Perl is used as a backbone for many websites, generating pages and other output. As a scripting language, it can be used for simple but time-consuming tasks without needing a lot of the set-up overhead required for a language such as C/C++ or Java. In the case of CraneWow, Perl is used for all the post-processing.

APPENDIX B

USER MANUAL

The CraneWow User's Manual is included with this thesis. Electronically, it is offered as CraneWowUserManual.pdf. This document provides step by step guides to set up and use the digiBASE-E array with the CraneWow software, as well as tips and troubleshooting help.

APPENDIX C

PROGRAMMING MANUAL

The CraneWow Programming Manual is also included with this thesis. Electronically, it is offered as CraneWowProgrammerManual.pdf. This document provides a more in-depth explanation of the digiBASE-E data stream, along with commentary and help regarding the functions and variables used by the program's code. This is intended to help future programmers and maintainers of the software in their duties.

VITA

Name: Gentry Charles Hearn

Address: Department of Nuclear Engineering
Texas A&M University
3133 TAMU
College Station, TX 77843-3133

Email Address: ghearn@riacc.com

Education: B.S., Computer Science, Tarleton State University, 2008
B.S., Physics, Tarleton State University, 2008