

Applications of high-speed digital pulse acquisition and software-defined electronics (SDE) in advanced nuclear teaching laboratories

F. D. Becchetti, N. Damron, and R. O. Torres-Isea

Citation: *American Journal of Physics* **88**, 70 (2020); doi: 10.1119/1.5125128

View online: <https://doi.org/10.1119/1.5125128>

View Table of Contents: <https://aapt.scitation.org/toc/ajp/88/1>

Published by the [American Association of Physics Teachers](#)

ARTICLES YOU MAY BE INTERESTED IN

[Electric field lines of relativistically moving point charges](#)

American Journal of Physics **88**, 5 (2020); <https://doi.org/10.1119/10.0000189>

[Why are complex numbers needed in quantum mechanics? Some answers for the introductory level](#)

American Journal of Physics **88**, 39 (2020); <https://doi.org/10.1119/10.0000258>

[Reflections in a moving mirror](#)

American Journal of Physics **88**, 46 (2020); <https://doi.org/10.1119/10.0000172>

[Celestial Calculations: A Gentle Introduction to Computational Astronomy](#)

American Journal of Physics **88**, 81 (2020); <https://doi.org/10.1119/10.0000284>

[Using Kepler's laws and Rutherford scattering to chart the seven gravity assists in the epic sunward journey of the Parker Solar Probe](#)

American Journal of Physics **88**, 11 (2020); <https://doi.org/10.1119/10.0000145>

[Determination of permeability using a classic Darcy water column](#)

American Journal of Physics **88**, 20 (2020); <https://doi.org/10.1119/10.0000296>



Learn about the newest **AAPT** member benefit



INSTRUCTIONAL LABORATORIES AND DEMONSTRATIONS

The downloaded PDF for any paper in this section contains all the papers in this section.

John Essick, *Editor*

Department of Physics, Reed College, Portland, OR 97202

Articles in this section deal with new ideas and techniques for instructional laboratory exercises, for demonstrations and for equipment that can be used in either. Although these facets of instruction also appear in regular articles, this section is for papers that primarily focus on equipment, materials and how they are used in instruction.

Manuscripts should be submitted using the web-based system that can be accessed via the *American Journal of Physics* home page, <http://web.mit.edu/rhprice/www>, and will be forwarded to the ADN editor for consideration.

Applications of high-speed digital pulse acquisition and software-defined electronics (SDE) in advanced nuclear teaching laboratories

F. D. Becchetti, N. Damron, and R. O. Torres-Isea

Department of Physics, University of Michigan, Ann Arbor, Michigan 48109

(Received 14 December 2018; accepted 8 August 2019)

There is a new generation of high-speed programmable pulse digitizers available now from several vendors at modest cost. These digitizers in tandem with on-board or post-processing software combine to produce a Software-Defined Electronics (SDE) system that can be effectively used in several advanced physics teaching lab experiments. In particular, as we will demonstrate, they are particularly well suited for nuclear-physics related experiments, often replacing many analog electronics modules. Appropriate on-board SDE can generate full or partial integrals of the pulses, pulse-shape characterization (PSD) data, coincidence signal indication, fast timing, or other information. Likewise, external PC-based SDE post-processing software can readily be developed and applied by undergraduate students or instructors using one of several different software languages available: MATLAB, PYTHON, LABVIEW, ROOT, BASIC, etc. As demonstrated here, an SDE-based system is a cost-effective substitute for many dedicated NIM or CAMAC electronics modules as this requires only a single digitizer module and a computer. A single digitizer with SDE is easily adapted for use in many different experiments. Applications of various high- and low-speed digitizers with SDE for many other types of physics teaching lab experiments will also be discussed. © 2020 American Association of Physics Teachers.

<https://doi.org/10.1119/1.5125128>

I. INTRODUCTION

Progress in modern, fast electronics has made many new modes of physics data acquisition possible. In particular, high-speed pulse digitizers with embedded field-programmable gate arrays (FPGAs) are rapidly replacing more conventional discrete, single-function NIM and CAMAC modules in many physics research experiments. In addition, as costs for these digitizer units continue to fall, they become attractive and affordable for use in advanced physics teaching laboratories, and in particular for many nuclear experiments.¹⁻³ While NIM and CAMAC electronics, along with analog or digital oscilloscopes, are still useful as teaching tools for some initial experiments, the use of a pulse digitizer and related software-defined electronics (SDE), which students or instructors can program if appropriate, introduces students to modern state-of-the-art physics research instrumentation and methods.

The on-board FPGAs and basic SDE software typically included with these digitizers can be used to initially display the pulse waveforms, gating signals, and timing signals in

digital storage oscilloscope mode (DSO), and then display the pulse integral spectra as well as basic pulse-shape information. This information can be recorded and analyzed directly or the digitizer module can be programmed to produce other outputs suitable for user-written SDE. The experiments described here use primarily student- or instructor-written SDE, with vendor-supplied basic SDE used mainly to monitor the real-time data acquisition. Alternately, the digitizer vendors or their user groups often can supply the necessary post-processing SDE software. As we wish to demonstrate, adapting pulse digitizers and related SDE for data acquisition and analysis in physics teaching labs (and in particular advanced nuclear labs) has the following advantages over past methods using arrays of modular electronics and instruments:

- (1) As noted, it introduces students (and in some cases also instructors) to modern state-of-the-art experimental physics research instrumentation and methods.
- (2) Students can be actively involved in programming the digitizers via the on-board FPGAs including specifying the type of data output. This requires student

understanding of the data needed, e.g., pulse height, pulse integral, time of flight, pulse-shape discrimination, coincident events, etc., and to set the appropriate digitizer parameters.

- (3) Students can apply and reinforce programming skills they have acquired, or readily learn what is needed, to create SDE to process and visualize the digitized pulse data.
- (4) The data can be acquired in a multi-parameter mode allowing for highly educational 2D or 3D data visualization, especially for coincident events. This is not easy to achieve with conventional modular electronics.
- (5) One digitizer can be used for several different types of experiments, often eliminating many of the specialized electronics modules needed for specific experiments (e.g., NIM or CAMAC modules and their power-supply crates). There can be considerable cost savings, in many cases, if one is buying new equipment. In addition, the designs of many NIM and CAMAC modules are old and contain internal components that are difficult to replace.
- (6) Several digitizers can be configured for a single experiment. Thus, depending on the number of detection systems available, several groups of students can simultaneously perform the identical experiment, e.g., for a studio type lecture-lab course.
- (7) Likewise, most digitizers can also be used as a high-speed SDE digital storage oscilloscope (DSO) replacing a conventional or digital oscilloscope. The multi-channel DSO mode is very instructive both for observing detector signals prior to digitization and facilitating the determination of FPGA parameters.
- (8) Better and more extensive data can usually be obtained compared to older methods. One can then often use radioactive sources with less strength and in particular low-cost US-NRC “exempt” quantity sources.
- (9) Most digitizers can be operated remotely, via Ethernet connections or other protocols. This permits students to run and control experiments remotely, permitting extended run times such as overnight runs or runs between lab periods. This again often facilitates the use of low intensity, US-NRC “exempt” radioactive sources or if using strong sources, can minimize students’ exposure to nuclear radiation.
- (10) The same SDE modules can usually be shared among several similar digitizers.
- (11) The digitizers readily permit open-ended student-originated experiments, including undergraduate honors theses and independent research and, also, graduate-level research, without need to purchase new, specialized electronics modules.

Although many suitable commercial digitizers exist and many new ones are being introduced on a regular basis, we will describe in more detail the use of the CAEN DT5790N module,¹ which is particularly well suited for advanced nuclear-physics teaching labs (Fig. 1). This unit consists of a two-channel high-speed digitizer (12 bits, 250 MS/s) conveniently combined with two programmable short-protected HV detector power supplies (4 kV max, 4 W each, SHV outputs) together with 12V 100 mA and 24V 50 mA low voltage supplies (DB9 outputs) suitable for preamps, etc. Like many digitizers, it can be controlled remotely, in this case via built-in Ethernet, USB, or fiber-optic interfaces. This permits

students to run and monitor experiments over long running periods beyond a normal lab session, especially when using low-intensity radioactive sources.

Another suitable stand-alone commercial digitizer with four 12-bit, 250 MS/s inputs (however no kV HV supplies) is the XIA PIXIE-Net.² Both the CAEN and XIA digitizers are modestly priced (\$4k–6k depending on options) and include extensive, user-friendly on-line operating manuals,^{1,2} which facilitate operation and programming the on-board FPGAs including specifying the format of the digitized output data. Both CAEN and XIA provide good technical support and have a large user base as both produce digitizers are widely used in physics research. As with the CAEN digitizers, XIA provides a number of SDE programs in C or LABVIEW⁴ that can be used with their digitizers.

Our particular CAEN unit (DT5790N) has two negative HV outputs suitable for most scintillator+photo-multiplier (PMT) detectors but can also be used to power other nuclear detectors using negative bias voltage. (The DT5790P model has positive HV supplies.) A lower-cost version (<\$5k) of this digitizer is available (CAEN DT5720) without the built-in power supplies but otherwise is similar and software compatible with the DT5790 for use with SDE. Like the XIA digitizer, the 5720 unit also is available as a four-channel digitizer.¹ Although CAEN makes a digitizer-based teaching module utilizing a DT5720 digitizer (CAEN SP5600¹), this module is primarily designed for experiments demonstrating the use of small-area silicon PMTs. Hence, it has more limited applications than the DT5790/5720 and XIA PIXIE-Net digitizers described here. Likewise, there are many other companies producing research-grade high-speed digitizers but most of these at this writing have more than four channels and usually require a special power-supply crate, which makes them expensive for teaching purposes.

We have configured a DT5790N digitizer together with suitable custom SDE software, mostly written by an undergraduate student (ND), for several important nuclear physics experiments typically done in advanced physics teaching labs:^{5–10} Gamma-Gamma Coincidence (Na-22 and Co-60 sources), Compton Scattering, Muon and Nuclear Decay Lifetimes, and Fission-Neutron Time-of-Flight, with additional experiments planned. Students or instructors, even those with minimal programming experience, can generate the needed SDE as part of a given experiment. Sample SDE programs used here, as noted mostly written in Python by a senior University of Michigan (UM) undergraduate student (co-author N.D.), and related information can be found in Ref. 43. However, the CAEN DT5720 unit is compatible with CAEN’s CoPASS vendor-supplied SDE software¹ so that is another option. (It is anticipated that CAEN will soon



Fig. 1. Picture of the CAEN DT5790N digital pulse analyzer/HV supply unit showing inputs and outputs (12 V and 24 V DB9 outputs in back, not shown).

have a version of CoPASS SDE software useable with the DT5790 digitizers.)

Most of the experiments described here are normally performed with modular NIM or CAMAC electronics and 1D multi-channel analyzers (MCAs). Such experiments are described in several commercial lab manuals,⁵⁻⁷ in various on-line advanced lab write ups (e.g., Refs. 8 and 9), or in relevant textbooks.¹⁰ The underlying nuclear physics being studied is covered in detail in many physics' textbooks.¹¹⁻¹⁴ Other physics teaching lab applications that can utilize lower-speed low-cost digitizers and SDE will also be described. A specific experiment developed with one particular low-cost modest-speed digitizer (Red Pitaya³) will be used as an example.

II. HIGH-SPEED DIGITIZED PULSE INFORMATION

In this paper, we primarily discuss applications of high-speed digitizers such as the CAEN and XIA units, which we define as having greater than 200 MS/s sampling rate. Such speed, which corresponds to <5 ns resolution, is often needed for nuclear experiments,⁵⁻¹⁰ including particle time-of-flight, fast coincidence measurements, pulse-shape discrimination (PSD), and other experiments using scintillators, solid-state Si or Ge detectors.

The digitizer unit used here (CAEN DT5790N¹) was purchased with FPGA software options appropriate for identifying and generating charged ion, neutron, electron, muon, or gamma-ray signals from one or two scintillators, either organic plastic or organic liquid scintillators, inorganic scintillators, as well as various solid-state or gaseous detectors. The XIA PIXIE-Net digitizer has similar options.² The CAEN 5790N digitizer has suitable remote-controlled HV detector power supplies, and includes two on-board programmable FPGAs to do specific user-defined or CAEN-software defined on-board SDE pulse analysis (Fig. 2). Again, the latter also applies to the XIA digitizer.

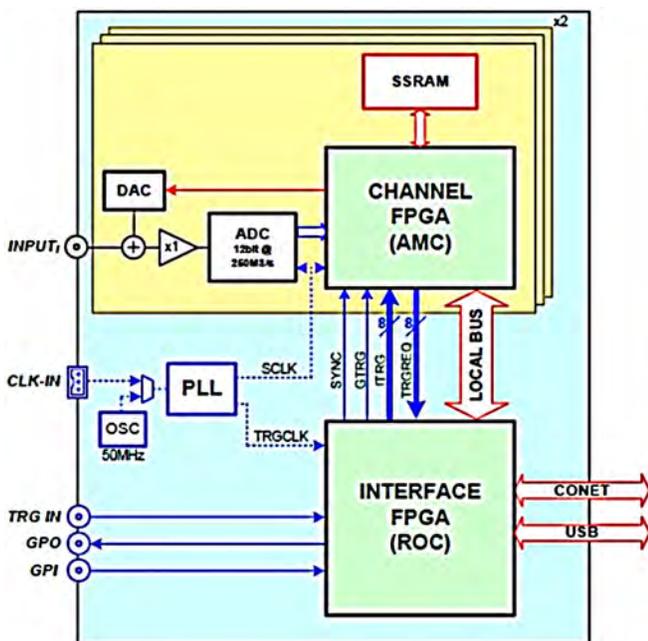


Fig. 2. On-board module layout of the CAEN DT5790 digitizer for one of the two channels (HV and preamp power supply circuits not shown). DAC = digital-to-analog converter; ADC = analog-to-digital converter; PLL = phase-locked loop; FPGA = field-programmable gated array.

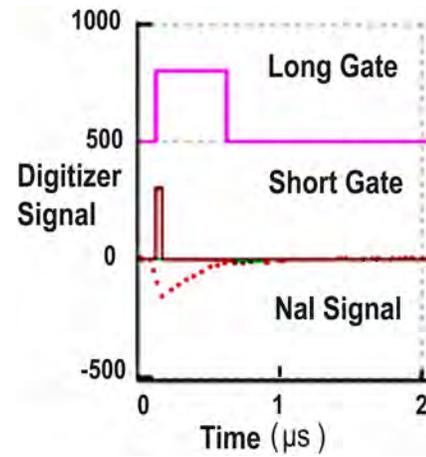


Fig. 3. Pulse digitization display (SDE DSO mode) showing the long (Q long) and short (Q short) pulse-integral gates used for a typical NaI(Tl) detector gamma-ray signal (PMT anode).

Examples of digitized pulse data from a scintillator PMT anode pulse observed using the DT5790N digitizer's SDE DSO mode are shown in Fig. 3 and Sec. III. In most plastic scintillators, there is no pulse-shape discrimination (PSD) but one can then use the time-stamped data from the digitizer (Fig. 4) to determine the signal timing (in this unit to 4 ns accuracy). The fast timing can be used between two scintillator signals to separate gamma rays from neutrons (or charged ions) via their time-of-flight (ToF) as determined with a user-defined SDE Time-to-Digital Converter (SDE TDC) in post processing or, e.g., using CAEN's CoPASS software if using the DT5720 unit. Similarly, with a SDE 2D-Multi-channel Analyzer (SDE 2D MCA), as an example for gamma-gamma coincidence measurements (see below), one typically just integrates the digitized pulses from two detectors with coincidence/non-coincidence signals generated in the on-board FPGA or alternately generated in SDE in post-processing using the time-stamped list data (Fig. 4). However, pulse pile-up rejection (PUR) often must also be

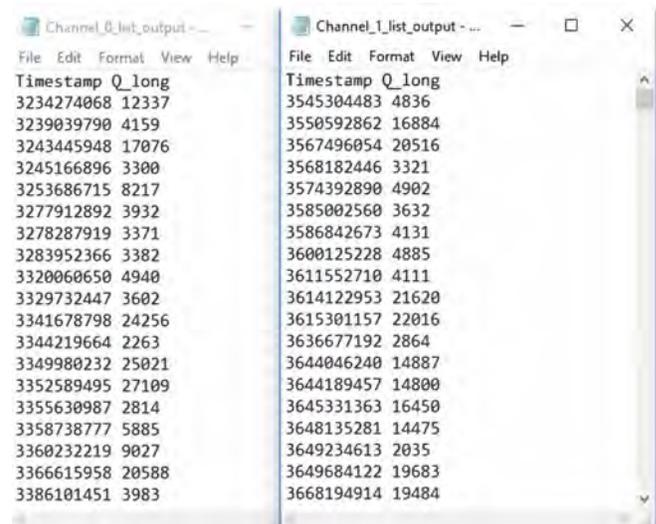


Fig. 4. Sample of list-mode data files generated in the data acquisition computer for two detectors having coincident gamma-ray signals. Shown are the time stamps (in 4 ns intervals) and the integrated scintillator pulse values (Q long, Fig. 3).

employed when using intense sources to avoid false coincident events (see below). As noted, the CAEN DT5720 and XIA units operate in a similar fashion but lack the built-in HV supplies. Likewise, the XIA PIXIE-Net unit has similar user-defined software options for displaying and generating digitized pulse information via the on-board FPGA² and has many of the same features of the CAEN units.

Most fast pulse digitizers have a limited input pulse voltage maximum, in the case of the DT5790 and DT5720 units, $\pm 2\text{ V}$ maximum and, for the XIA unit, $\pm 3.5\text{ V}$. However, it is sometimes desirable to obtain optimal signal-to-noise performance from a PMT (or other detector) and thus run these detectors at their specified optimum operating voltage. This may produce signals outside the range of the digitizer. In such cases precision GHz-bandwidth $50\ \Omega$ in-line attenuators¹⁵ can be used, and often are needed, to match the digitizer pulse voltage input limits. This is often preferable to lowering the PMT voltage well beyond the specified operating voltage to reduce the input signal voltage.

Below we illustrate the use of SDE software as applied to advanced undergraduate teaching-laboratory nuclear experiments with the key SDE modules indicated.

III. EXPERIMENTS WITH HIGH-SPEED DIGITIZERS AND SDE

A. Na-22 coincident positron-annihilation radiation (SDE 2D MCA and TDC)

The proton-rich radioactive nucleus Na-22 (2.6 year half-life) preferentially decays to the stable nucleus Ne-22 first via positron emission or electron capture to the 1.27 MeV first excited state in Ne-22 (see Ref. 6, Expt. 13). This state then decays via gamma emission to the Ne-22 ground state. The positrons emitted then annihilate in the source by combining with an atomic electron, producing primarily two back-to-back 511 keV gamma rays in coincidence.¹⁰

The set up for detecting the two coincident positron-decay gammas typically consists of two scintillator+PMT detectors.⁵⁻¹⁰ Fast coincidence timing usually is best done using plastic scintillators, but owing to their low effective atomic number Z , as they are organic, they do not produce photo-peaks at well-defined energies,¹⁰ in this case, 511 keV. Instead they produce non-distinct Compton spectra extending out to the Compton maximum energy (about 341 keV). NaI(Tl) scintillators are thus preferred to measure gamma energy spectra as these produce full-energy photo-peaks. They are considerably slower (about 10 times slower) than plastic for coincidence timing^{10,16,17} but their response time is adequate for this experiment.

We have used two NaI(Tl) detectors with the CAEN DT5790N digitizer and a student-written SDE 2D multi-channel analyzer (SDE 2D MCA) module. The latter can confirm that the two Na-22 positron-annihilation gammas occur in fast coincidence, each with energy 511 keV, and occur back-to-back, i.e., 180° apart. This is required to conserve momentum as the positron annihilates with an electron in the source while both are nearly at rest in the plastic source material. In this case one can use the time-stamped NaI(Tl) scintillator pulse integrals (Fig. 4) with SDE time-to-digital converter (TDC) and 2D MCA modules in post-processing to identify both coincident and non-coincident gammas and determine their energy spectra. Detectors placed at 90° and other angles relative to each other can then verify that the two

coincident gammas are emitted back-to-back (180° apart). Post-processing SDE software written in PYTHON (by UM undergraduate student ND) was used to generate 2D MCA plots, where the coincident gamma spectra along with random coincident events can be displayed (Fig. 5). As shown in Fig. 5, the coincident gamma photo-peaks are determined to each be 511 keV. However, we note that coincident Compton scattering of the 511 keV gammas in the NaI(Tl) detectors also is observed (Fig. 5) along with some random coincident events as pile-up rejection was not implemented. Such 2D data visualization, which can be highly educational, requires multi-parameter coincident data acquisition and display (SDE 2D MCA), not easily done with conventional electronics and single-channel (1D) pulse-height analyzers.

B. Application: PET medical imaging

This experiment serves to introduce students to a highly effective physics-based modality in medical imaging,

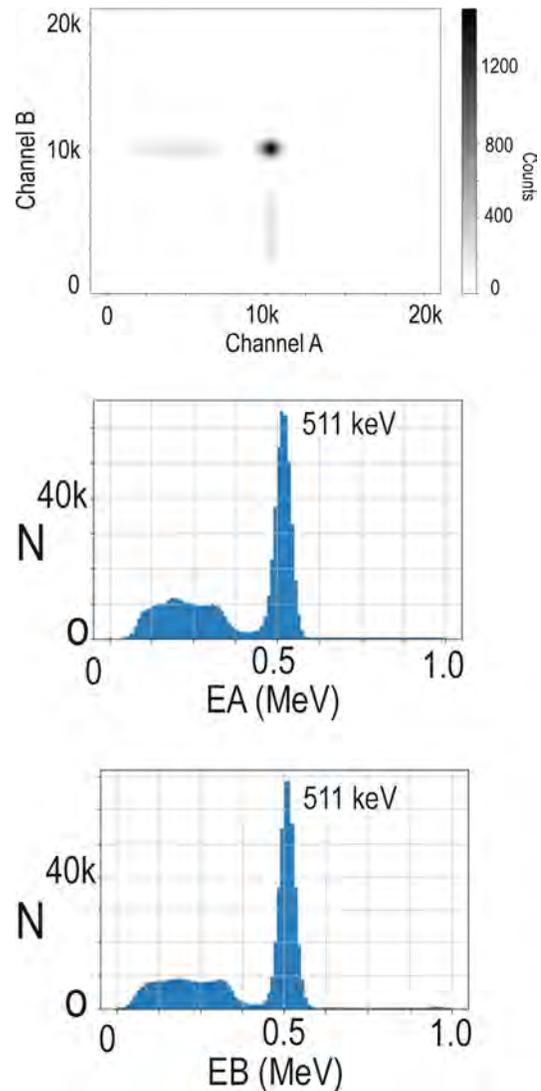


Fig. 5. Top: 2D coincident Na-22 NaI(Tl) gamma-ray digitized pulse integral spectra in two detectors, denoted as detector A and detector B, generated using student-written post-processing Python 2D MCA SDE. Bottom: The projected SDE NaI(Tl) calibrated energy spectra (SDE 1D MCA) in the corresponding detectors for the two coincident Na-22 annihilation gamma rays detected 180° apart.

positron-emission tomography (PET). As seen in Fig. 5, a positron-emitting source can readily be identified via the detection of coincident 511 keV annihilation gamma rays, as these provide a unique signature. Using suitably collimated detectors, the location of the annihilation, which could be a cancer tumor, can be located to a few mm, e.g., 2–5 mm depending on the radioisotope employed¹⁸ (see also: Wikipedia “Positron emission tomography” and references cited there). PET imaging uses short-lived positron-emitting radioisotopes such as C-11, N-13, O-15, F-18, Ga-68, Zr-89, and Rb-82 incorporated into special radio-pharmaceuticals. The latter accumulate in one or more organs in the body having specific medical problems. The PET images then provide both static images as well as physiological information, e.g., the uptake rate of the radio-pharmaceutical.¹⁸ This can facilitate discovery and treatment of cancer tumors at an early stage, detection and analysis of cardiovascular and neurological disease, etc.¹⁸ This imaging modality has low background as it requires, as illustrated above, the detection of two coincident 511 keV positron-annihilation gamma rays. Hence very low amounts of radioactive material with minimal radiation dose to a patient are needed to obtain a good image.¹⁸

C. Co-60 gamma-gamma coincidence (SDE 2D MCA)

The radioactive nucleus Co-60 beta decays with a half-life of 5.26 years to excited states of Ni-60 (see Ref. 6, Expt. 19) The $J^\pi = 4^+$, $E_x = 2.51$ MeV Ni-60 excited state (recall J is the spin and π is the parity of the level) decays sequentially in coincidence with the decay of the $J^\pi = 2^+$, 1.33 MeV excited state to the $J^\pi = 0^+$ ground state of Ni-60. This produces two prominent coincident gamma rays at $E = 1.17$ MeV and 1.33 MeV (see Ref. 6, Expt. 19). Coincident gamma-ray studies have proved especially important in studying and identifying nuclear levels, especially in deformed nuclei where the level spacing, often up to $J^\pi > 50^+$ levels (!), can be used to deduce nuclear moments of inertia.^{19,20} Most of these experiments now use very large arrays of gamma-ray detectors together with high-speed pulse digitizers.²¹ Such experiments have revealed some dramatic and unexpected results regarding nuclei.^{19–21} Gamma-gamma coincidence is an important nuclear experimental technique studied in most advanced teaching labs¹⁰ and is readily demonstrated with the SDE system described here.

The setup for this experiment again consists of two NaI(Tl) scintillators, denoted as detectors A and B, and PMTs (used to generate gamma-ray energy-related signals E_A and E_B) set at specific angles relative to a Co-60 source. In the present case a modest, hence low-cost, US-NRC exempt 1.9 μ Ci Co-60 source was used together with two lead-shielded 2 in. long by 2 in. dia. NaI (Tl) detectors. These were set 180° apart and 3–4 cm each from the source, similar to the setup used for the Na-22 coincidence experiment (above). The coincidence timing was checked using the DSO mode in the digitizer together with a Na-22 source, although in some cases this can also be done using a split signal from a Phillips Scientific Model 417 high-speed pulser (Phillips Scientific, Mahwah N.J.) to simulate the detector PMT anode pulses. The PMT HV to each detector was adjusted so the detector gains were approximately matched, but in any case, the energy spectra derived from the PMT anode pulse integral, using a 400 ns pulse-integral gate (long gate, Fig. 3), could be determined in units of MeV using

Na-22 and other known gamma-ray energy calibration sources. The on-board digitizer coincidence timing window was then set (40 ns wide) to ensure primarily fast coincidence data were recorded along with any random coincidences also falling within that window.

Random, overlapping coincidences, as observed using the digitizer’s two-channel DSO mode, initially proved to be an issue, and it was necessary to implement the user-defined on-board pulse pile-up rejection (PUR) SDE supplied by CAEN. This proved effective in eliminating most double pulses within each detector’s pulse-integral 400 ns wide gate window. PUR then allows one to count at relatively high rates (a few kHz) with minimal random coincidences or single-detector pulse pile-up in the spectra. Excessive low-energy events can be removed with the user-defined threshold settings in the FPGA, which specify the lowest voltage signals to be further digitized. This permits students to use close detector geometry and employ exempt-level radioactive sources (here a 1.9 μ Ci Co-60 source) to obtain sufficient coincidence data (Figs. 6 and 7). Since the sequential decays in Ni-60 occur within picoseconds (see Ref. 6, Expt. 19), the two gamma rays in the decay to the ground state (g.s.) appear to occur simultaneously in the NaI(Tl) detectors. However, there is still a small number of random events in the NaI(Tl) coincidence window, which appear as coincident events in the 2D and gated spectra (Figs. 6 and 7) but are from two different nuclear decay events that yet occur within the coincidence timing window.

A PYTHON SDE 2D-MCA routine was written (again by UM student ND) to display both coincidence gamma rays and any random coincidences together with gamma rays outside of the coincidence window. A 2D plot (E_A vs E_B) of the detected gamma-ray energies (online color) is shown in Fig. 6. and confirms that the 1.17 MeV and 1.33 MeV gamma rays occur in coincidence, as allowed by the selection rules for gamma-ray decay for the levels involved (see Ref. 12, Chap. 10). Both the photo-peaks, along with the corresponding Compton spectra for these gamma rays, are observed. There also is indication of some 1.33 MeV and 1.17 MeV events due to decays from different nuclei that nonetheless

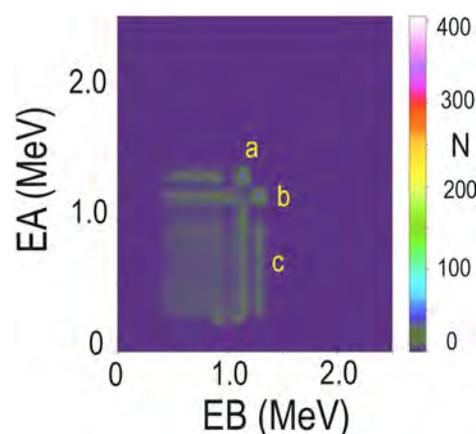


Fig. 6. A 2D energy spectrum (E_A vs E_B) of pulse-integrated signals from a Co-60 source observed with two NaI(Tl) gamma detectors (denoted A and B) run in coincidence, 180° apart (SDE 2D MCA). Two strong Ni-60 gamma decay groups are indicated in coincidence in separate detectors, at 1.17 MeV and 1.33 MeV. Group a: 1.33 MeV gamma in detector A in coincidence with 1.17 MeV gamma in detector B; group b: 1.17 MeV gamma in detector A in coincidence with 1.33 MeV group in detector B; group c: Typical coincident Compton events in the detectors.

occur within the coincidence window. Projections onto the energy axes of one detector showing all coincident events as well as only 1.33 MeV gammas coincident with 1.17 MeV gammas are shown in Fig. 7.

As another part of this experiment students can measure the angular correlation between the two gamma rays, which is related to the nuclear spins and multipolarity of the levels. The correlation for the decay gammas from the two Ni-60 levels is fairly uniform versus angle, however (see Ref. 6, Expt. 19).

D. Random coincidences and dead-time measurements (DSO mode)

The digitizer SDE DSO mode allows students to observe random coincidences between two detectors, i.e., events in two detectors not in precise time coincidence. This can be studied in more detail by changing the user-defined coincidence timing gate width (see above). Likewise, pulse pile-up in the detectors can be studied using different source-detector distances, hence count rates, or a random-pulse generator and observing the errors introduced in the pulse integrals or related effects (see Ref. 10, Chap. 5).

E. Compton scattering (SDE TDC, 1D- and 2D-MCA)

Compton scattering is another important advanced nuclear-physics lab experiment that requires a fast coincidence measurement, in this case between a scattered gamma ray and the recoil electron (see Ref. 10, Sec. 2.7.2; Ref. 6, Expt. 10). This is readily done with a two-channel fast digitizer and vendor-supplied, student- or instructor-developed

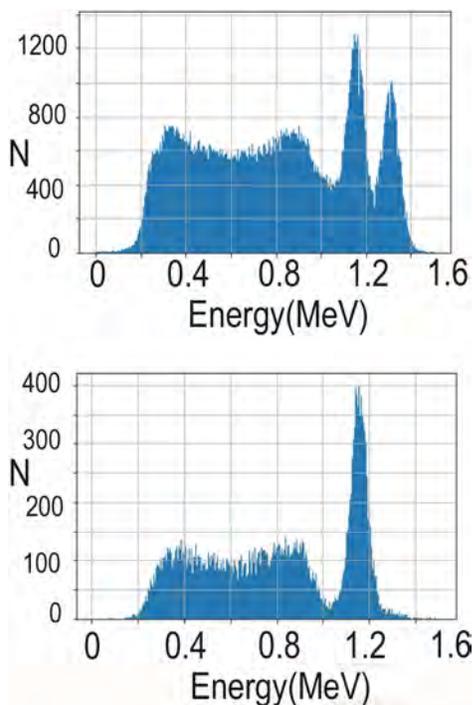


Fig. 7. Projected coincident Ni-60 gamma-decay spectra from a Co-60 source (Fig. 6) observed in one of the NaI(Tl) detectors (SDE 1D MCA). Top: All coincident events showing that the 1.17 MeV and 1.33 MeV gamma rays are in coincidence. Bottom: Projected spectrum (SDE 1D MCA) in the same detector but requiring coincidence with a 1.33 MeV gamma in the second detector, confirming the 1.17 MeV gamma is part of a fast gamma decay sequence between nuclear levels in Ni-60.

SDE TDC, 1D-, and 2D-MCA modules. A standard Compton scattering setup can be used,⁵⁻⁹ with a plastic scintillator and PMT serving as the recoil-electron detector and a NaI(Tl) with PMT serving as the detector for the Compton-scattered gamma ray (Fig. 8). The coincidence timing window and energy calibrations are done as previously described. SDE post-processing software, which can be student-written as done here, is then used to create SDE 1D- and 2D-MCAs, a SDE TDC and other modules for visualization of coincident and non-coincident scattered gamma rays and recoil electrons. In this case, there is a direct correlation between these energies and the gamma-ray scattering angle via the well-known Compton relation.¹¹⁻¹⁴ The latter requires that the summed energy be equal to the incident gamma ray energy, in this case 662 keV (Cs-137). After calibrating the detectors with gamma-ray sources, data were obtained over several gamma scattering angles in the range from 30° to 90° (Fig. 9). This was done with (roughly) 2 in. × 2 in. diameter plastic and NaI(Tl) detectors using a 288 μCi collimated Cs-137 source. This is a strong, non-exempt source, but if longer running periods are available the experiment can be done with close-in geometry and exempt sources of a few μCi (see Ref. 6, Expt. 19). Likewise, four-channel digitizers (CAEN DT5720 and XIA PIXIE-Net) would permit measurements at several gamma-ray scattering angles simultaneously, thus greatly expediting measurements.

Two-dimensional coincidence data spectra (SDE 2D MCA) summed over the gamma scattering angles measured are shown in Fig. 10. As predicted by the Compton equation, the scattered gamma ray and recoil electron energies are correlated and sum to 662 keV (within uncertainties) at all the gamma-ray scattering angles measured. As shown, this can be readily verified and visualized with appropriate SDE modules using the digitized coincident output data from a digitizer.

F. Related experiments utilizing multi-channel digitizers

As an additional part of the above experiment, students can also measure the gamma-ray Compton scattering probability versus scattering angle and verify the prediction of the Klein-Nishina theory derived from quantum electrodynamics (see Ref. 12, p. 200). This measurement, however, requires further information, such as the acceptance and detection efficiency of the two detectors as a function of the gamma ray or recoil electron energy.^{10,16,17}

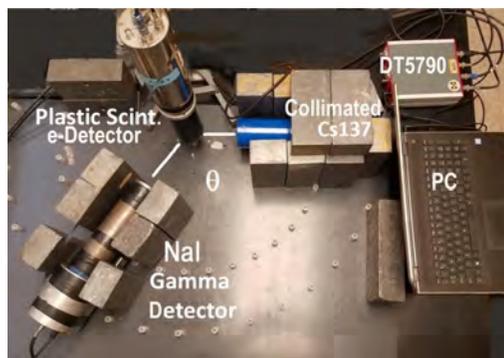


Fig. 8. Compton scattering setup as used with the CAEN DT5790N digitizer/HV supply and a collimated 0.288 mCi Cs-137 gamma source (Refs. 5-9).

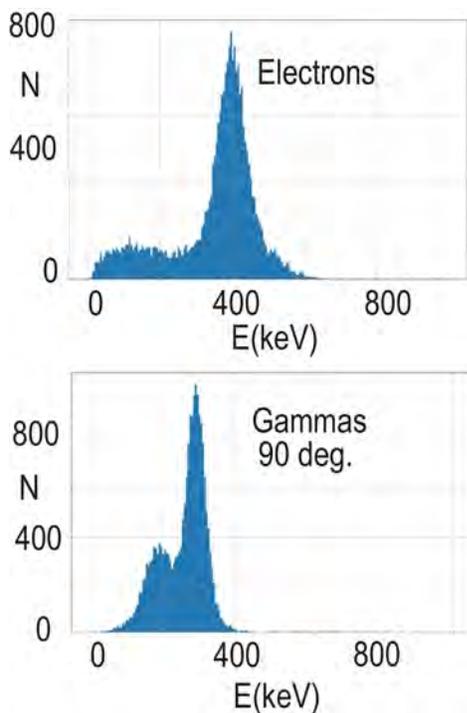


Fig. 9. Electrons (top) detected in a plastic scintillator (Fig. 8) in coincident with gamma rays (bottom) scattered at 90° and detected with a NaI(Tl) scintillator, as measured using 1D MCA SDE.

In these and other measurements, having a four-channel digitizer (CAEN DT5720 or XIA PIXIE-Net) with four detectors running simultaneously can be advantageous. Likewise, four detectors would expedite measuring the angular correlation in gamma-gamma coincident decays (e.g., Co-60; Ref. 6, Expt. 19), but also for measuring the angular distribution in alpha-particle Rutherford scattering (see Ref. 6, Expt. 15), and other experiments.⁵⁻¹⁰ This could permit these experiments to be done in a single lab period in many cases yet still using US- NRC exempt low-intensity radioactive sources.⁵⁻⁹

G. Muon lifetime (SDE TDC and 1D MCA)

Another important experiment is measurement of the cosmic-ray muon lifetime (see Ref. 11, Sec. 9.5).²² This experiment illustrates Einstein's theory of relativity, e.g.,

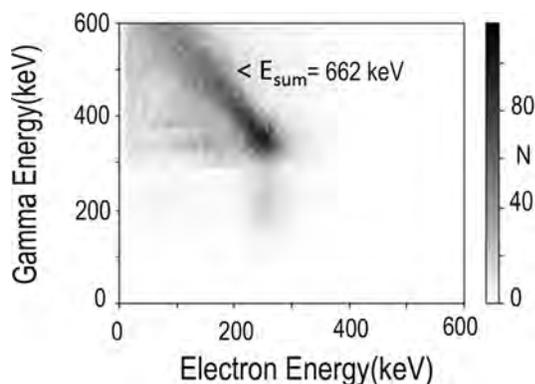


Fig. 10. SDE-generated 2D plot (2D MCA SDE) of coincident recoil electron and Compton-scattered gamma energies (Fig. 9) summed over several gamma-ray scattering angles, 30° to 90° .

time dilation in our reference frame and length contraction in the muon reference frame given that muons are produced in the upper atmosphere and would otherwise decay before reaching the Earth. The experiment can introduce students to topics such as DNA damage from cosmic rays, DNA repair via double strand DNA, and the time scale for evolution (see, e.g., Wikipedia and references cited there). It also introduces students to heavy leptons and the Standard Model of physics.¹¹⁻¹⁴

In this version of the experiment, a simplified version of a standard setup (e.g., Ref. 8) can be used as the needed data can be obtained from a single PMT detector.²² The original muon decay detector used by our students⁸ consists of a large drum of oil scintillator with two large (5 in.) PMTs and associated electronics to trigger on any incident muon stopped in the scintillator. One PMT signal is used to trigger on the incident muon and a second PMT to trigger on the muon decay signal with a fast time measurement between them. This can now be done using only one PMT with just the digitizer/HV supply unit and suitable post-processing SDE TDC. The only requirement is that a wide digitizing window is specified ($>$ a few μ s) such that both the incident and decay pulses are observed together, then digitized and recorded as event-mode data (12 bits, 250 MS/s). The post-processing SDE can then be written to locate the incident and decay signals (Fig. 11) and generate the time observed for the muon to decay when stopped via SDE TDC and 1D MCA modules. As noted, most digitizers can be operated remotely or in stand-alone mode, and data accumulated beyond the normal lab period to obtain good statistics (as done here). The mean lifetime of the muon when stopped (neglecting some small QM effects⁵⁻⁹) after correcting for random decay pulses (a constant background in the decay spectrum) can then be determined accurately (Fig. 12).

Since one large scintillator can accommodate several PMTs (as in the UM setup noted above in Ref. 8) and only a single digitizer is needed to acquire the needed PMT data, several students can be running the muon lifetime experiment using different PMTs at the same time, if appropriate.

H. Fission-neutron time of flight (SDE TDC and 1D MCA)

This experiment is done with a sealed, depleted oil-well-logging Cf-252 spontaneous fission source, as described in Ref. 23. The SDE version replaces many analog electronics modules (NIM or CAMAC) with the CAEN two-channel

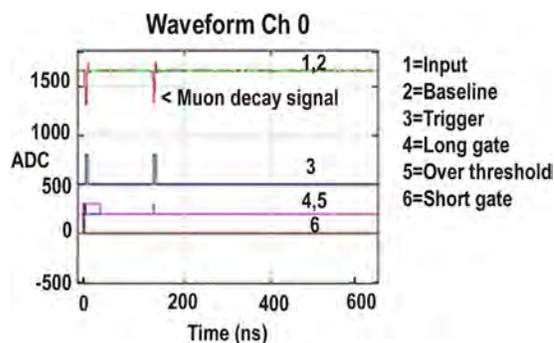


Fig. 11. Typical incident cosmic-ray muon (see Ref. 11, Sec. 9.5) and associated stopped-muon decay pulses as recorded in a digitizer data acquisition SDE DSO window.

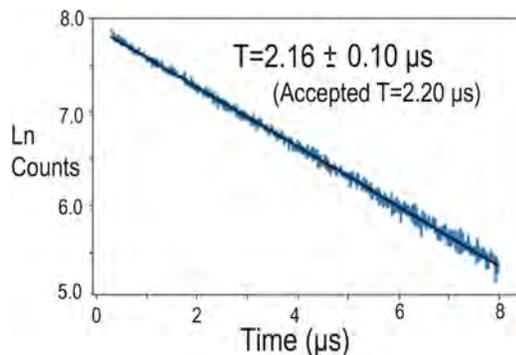


Fig. 12. SDE 1D-TDC time spectrum of stopped-muon decay data after background subtraction (uncorrected for small QM capture effects) indicating the fitted mean lifetime with uncertainty compared with the accepted intrinsic lifetime value (see Ref. 11, Sec. 9.5).

digitizer/HV supply operating in time-stamped list mode. In this mode, the signal from the start and stop plastic scintillators is recorded in the PC as a list of pulse integrals stored with a time stamp in 4 ns intervals (Fig. 4). SDE post-processing modules (in Python) written by one of us (ND) replacing analog coincidence and time-to-digital conversion (TDC) modules were then used to determine the time-of-flight (ToF) of the detected neutrons and gamma rays at the far detector. The scintillators were placed about 1 m apart (Fig. 13). The bottom scintillator is triggered by prompt gammas released in the spontaneous fission of Cf-252.²³ Typical time-of-flight data observed in the far detector as generated by SDE TDC post-processing software are shown in Fig. 14. The corresponding neutron energy spectrum

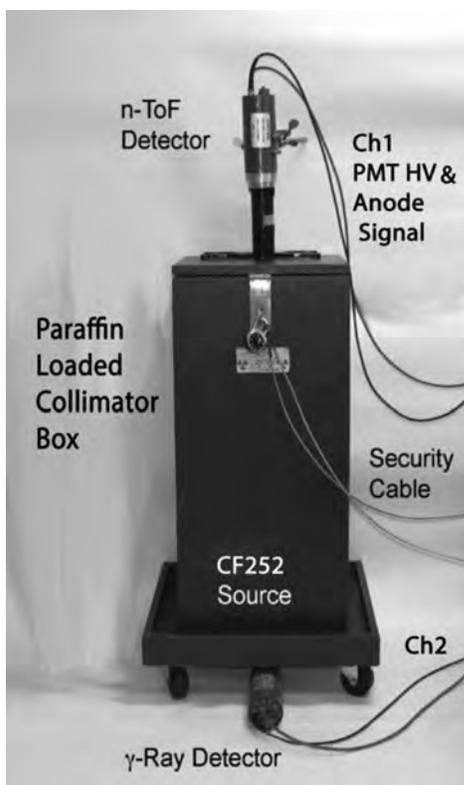


Fig. 13. Cf-252 fission neutron ToF setup (Ref. 23) with the plastic scintillator PMT HV and anode signals indicated for the two DT5790N digitizer channels.

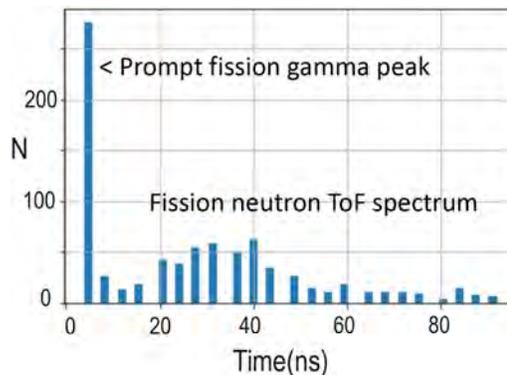


Fig. 14. Cf-252 fission ToF spectrum using a SDE TDC with the time-stamped scintillator data (Fig. 4). The zero-time value has been slightly offset for clarity.

deduced from the SDE TDC ToF spectrum (SDE 1D MCA), assuming non-relativistic kinematics since velocity $v \ll c$, is shown in Fig. 15.

As seen in Fig. 14, the prompt gammas that trigger the far detector occur well before the fission neutrons as, of course, the gammas travel at the speed of light c , whereas the neutrons are travelling at only a fraction of c with a kinetic energy of a few MeV.²³ While this particular digitizer has limited time resolution (4 ns intervals), the resulting ToF spectrum with a (roughly) 1 m flight path is sufficient to analyze the spectrum in terms of the expected Watt fission-neutron energy spectrum (Fig. 15) or other models.^{23,24} However, depending on the age of the source, and especially if a depleted well-logging source is used, there can be some daughter isotopes in the source that produce high-energy fission neutrons and these can distort the ToF data²⁵ from that expected for a Watt fission-neutron spectrum (Fig. 15). As with other experiments, the data were accumulated over a long run-period using the event-mode data output option of the CAEN digitizer. This allowed us to use a depleted source within our US-NRC possession limits for Cf-252 (100 uCi).

The SDE TDC using time-stamped data even at 4 ns intervals can nonetheless yield timing resolution between a set of two detector pulses to 1 ns or less. This is possible if a large number of events are analyzed to give time distributions with accurate *mean* timing values^{10,16,17} as the latter can be much better than the timing resolution for a single pair of events. The typical plastic scintillators used in this

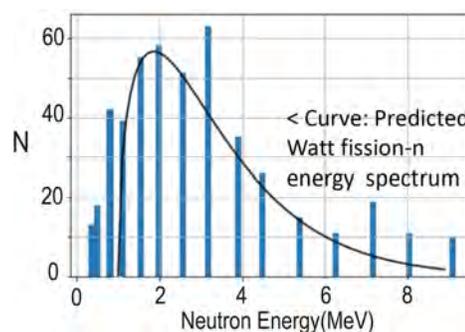


Fig. 15. Fission neutron energy spectrum (SDE 1D MCA) deduced from the ToF SDE TDC spectrum (Fig. 14). The curve shown is the predicted Watt energy spectrum for Cf-252 (Refs. 23–25). However, depending on the age of the Cf-252 source, the observed neutron energy spectrum can be slightly distorted due to fission of daughter nuclei (Ref. 25).

experiment do not allow for pulse-shape discrimination so the on-board vendor-supplied PSD SDE or student-developed PSD SDE cannot be used. But for the fission experiment, except for some delayed gammas, most gamma rays can be separated via their ToF using a user-written SDE TDC (Fig. 14). However, the use of certain liquid, crystalline or newer plastic scintillators which have PSD capability^{26,27} could be utilized with on-board or user-developed SDE replacing the CAMAC modules used in Ref. 23 for PSD. In particular, in an organic liquid scintillator, neutrons can be separated from gamma-ray signals by using a pulse-shape identification (PSD) parameter based on the long and short integrated pulse gates (Fig. 3) together with the FPGA SDE software supplied by CAEN as an option¹ and similar software from XIA.² Electrons, muons (e.g., cosmic-ray muons), and charged particles can similarly be identified if needed using PSD in some cases.^{26,27}

I. SDE MCA and MCS applications

The simplest application of a pulse digitizer with vendor- or user-supplied SDE is for experiments requiring just 1D multi-channel analyzer (MCA) or multi-channel scaler timing (MCS) functionality. Examples of experiments for a SDE MCA are gamma-ray, beta-, and alpha-particle energy spectroscopy.^{5-9,28-30} In this case, the vendor-supplied SDEs using the on-board FPGAs can often be used to give integrals of the digitized energy-related pulses from appropriate nuclear detectors. Energy spectra, with user-defined low-energy thresholds and PUR employed as needed, can then be generated, displayed, and recorded online.¹ Alternately one can apply user-written post-processing SDE to the event-mode data list sent to the host PC (Fig. 4). Conveniently, most digitizers have built-in base-line stabilization and corrections for zero offset available to optimize the energy spectra. A typical solid-state silicon detector will usually require a preamplifier, which will produce slow charge-to-voltage signals^{5,6,10} that can still be post-processed and calibrated with suitable SDE to give energy spectra (SDE MCA). These preamps also often have fast timing output signals.^{5,6,10} The latter can be appropriate for high-speed digitizing and, via pulse integration, produce useable energy spectra.

Digitizing HPGe gamma data^{6,7,10,16,17} can be more problematic due to the limited resolution of modest-cost digitizers as used here (12 bit, i.e., 4096 channels for the CAEN DT5790, DT5720, and XIA PIXIE-Net units^{1,2}). In this case for high-resolution spectroscopy requiring only 1D spectra acquisition,²⁸⁻³⁰ it may be more appropriate to use a modern, compact high-resolution (8000 channels or more) digitizer or dedicated MCA unit. As an example, the Amptek 8000D MCA unit uses a modest speed 100 MS/s, 16-bit pulse digitizer and a dedicated software³¹ to provide MCA and MCS functionality suitable for many experiments not requiring multi-parameter data acquisition. A similar MCA/MCS single-channel unit but lower resolution, the Avicenna RX1200 uses a 12-bit, 200 MS/s digitizer³² together with an on-board FPGA to allow custom pulse processing. Due to the relatively low resolution data (a few %) from NaI(Tl) and plastic scintillators obtained from typical PMT anode or dynode signals, adequate MCA gamma-ray or particle spectra (see above) can usually be obtained for detectors with 12-bit digitizers, including the DT5790, DT5720, XIA PIXIE-Net, and other units such as the Avicenna digital MCA.³²⁻³⁶

MCS functionality, for example, to measure nuclear decay lifetimes (e.g., see Ref. 6, Expt. 14), is often readily available via time-stamped data (Fig. 4). Thresholds for the pulse inputs or pulse integrals can be set by users (including students as appropriate) using the on-board FPGAs and SDE DSO display, or in post-processing SDE, to select specific nuclear-decay pulses. Using the list-mode data and appropriate SDE in post-processing, one can then accurately measure nuclear decay lifetimes. Given the high sampling rates (e.g., on the order of MS/s), short lifetimes can be accurately measured, such as the 2.6 min Ba-137m 662 keV gamma-decay half-life using a Cs-137/Ba-137m generator.^{5,6,28,29}

IV. ALTERNATE, LOWER-COST DIGITIZERS, DSOs AND SDE APPLICATIONS

The cost of pulse digitizers will in general depend on the digitized pulse-bit resolution, the sampling rate, and additional features. In teaching-lab applications not requiring high-speed digitization, significant savings can be obtained using a lower-cost digitizer with lower than 200 MHz/s sampling rate. One such unit is the Red Pitaya StemLabs 125–14 digitizer.³⁶ The latter is well suited for SDE as it uses a FPGA with a two-channel 125 MS/s, 14-bit digitizer. Unlike the CAEN DT5790 units, this and most other digitizers require external HV and preamp detector power supplies often with an external digitizer power supply or instrumentation rack also needed. Likewise, low-cost programmable controller modules such as the Arduino UNO with external sensors have been employed for teaching,³⁷ but again these require separate power supplies, preamp circuitry, etc. Such controllers, digitizers, and FPGA units come with various degrees of functionality in the vendor-supplied or user-group SDE, including technical support, but many can be programmed to generate list-mode data that can be post-processed by students with custom SDE.³⁶⁻³⁸

We are presently evaluating several such units and related custom or other SDE with encouraging results (below). Many low-cost digitizers appear well suited for several introductory and advanced atomic, molecular, and other physics teaching-lab experiments not requiring high-speed or high-resolution pulse digitization.^{8,9,39} In particular, together with several undergraduate students, an experiment on chaotic systems using a Red Pitaya digitizer^{3,36} has been developed. Here again, the digitizer data are readily visualized using student-, instructor-, or other SDE-software to display various important features of chaotic motion in multi-parameter space. Details of the experiment and SDE software will be presented elsewhere.

Another type of digitizer, which often can be utilized for data acquisition in an advanced laboratory including many of the experiments described here, is a digital storage oscilloscope (SDE DSO).⁴⁰ These often have provisions for exporting digitized multi-channel data to a PC in a form suitable for implementation of post-processing user-defined or other SDE. Some external detector power supplies may still be needed and most DSOs are not as versatile in terms of programming on-board calculations compared with a digitizer having on-board user-accessible FPGAs.

V. IMPLEMENTATION

While the experiments described here were done using SDE software primarily written by a senior undergraduate

student (ND), further evaluation by a wider range of students will be needed before wide-spread implementation in our advanced lab curriculum. This will determine what aspects of the SDE software programming will be required of the students and for what particular experiments this will be appropriate. In any case implementation of various SDE software programs, including vendor- or instructor-supplied SDE software developed for the CAEN two-channel (DT5790,DT5720), CAEN four-channel (DT5720), XIA PIXIE-Net four-channel, and Red Pitaya digitizers, is initially planned. This will provide our students with an introduction to modern high-speed pulse digitization and SDE. Although the present work utilized primarily a DT5790N digitizer with built-in HV supplies, the CAEN DT5720 digitizer, which lacks the HV power supplies, has the advantage that it can use the extensive CAEN-supplied CoMPASS SDE software¹ and is available as a four-channel unit. The DT5720 digitizer and CoMPASS SDE software are being evaluated at both UM and the University of Illinois for nuclear physics teaching lab purposes.⁴¹ Also, the CoMPASS SDE software is being modified by CAEN for use with the DT5790 digitizers.

As indicated, other research-level high-speed digitizers also may be adaptable for teaching applications with appropriate SDE (e.g., see Refs. 34 and 35). Likewise, there are several modest-speed low-cost digitizers, e.g., see Ref. 36, that utilize vendor- or user-supplied SDE that can be effectively employed in other types of teaching labs (atomic and solid-state physics, optics, and so on) not requiring high-speed, high-resolution digitizing or fast coincident signal timing.

VI. CONCLUSIONS

Modern high-speed (>200 MS/s) (Refs. 1 and 2) and many modest-speed digitizers (e.g., see Ref. 3), as demonstrated here, can effectively be used with appropriate vendor-supplied on-board, or user-written post-processing software-defined electronics (SDE) for advanced physics teaching labs, and in particular nuclear-physics experiments. The needed SDE can be student- or instructor-generated if appropriate using one of many common programming languages. The digitizer with suitable SDE can replace much of the conventional electronics and power supplies (e.g., NIM and CAMAC modules and power bins) used in many of these experiments. Multi-parameter data can readily be acquired and displayed allowing for highly visual multi-dimensional presentation and analysis of the data. Pulse digitizers and SDE introduce students to modern state-of-the-art physics data acquisition, signal processing, data visualization, and analysis as also used in many other areas of present-day scientific research.

ACKNOWLEDGMENTS

This work was supported in part by REU supplement to NSF grant PHY 14-01242. This paper is based mostly on the work presented by one of us (N.D.) at the October 2018 Fall APS meeting (Ref. 42).

¹CAEN Technologies, Staten Island N.Y. 10305, <www.caen.it>.

²XIA LLC, Hayward CA 94544, <www.xia.com>.

³Red Pitaya, <www.redpitaya.com>.

⁴John Essick, *Hands-On Introduction to LabVIEW for Scientists and Engineers*, 4th ed. (Oxford U.P., New York, 2019).

⁵Jerome L. Duggan, *Laboratory Investigations in Nuclear Science* (The Nucleus, Oak Ridge, TN, 1988).

⁶*Experiments in Nuclear Science, AN34*, 3rd ed. (Ortec-Ametek, Inc., Oak Ridge, TN, 1987), <www.ortec-online.com>.

⁷*Laboratory Manual: Nuclear Science Experiments with Digital Electronics* (Mirion Technologies, Hourseheads, NY, 2017), <www.canberra.com>.

⁸*Physics 441/442 Advanced Laboratory Notes* (University of Michigan, Michigan, 2018), <<http://instructor.physics.lsa.umich.edu/adv-labs>>.

⁹*Advanced Physics Laboratory Manual*, edited by J. W. Hammer (University of Notre Dame, Indiana, 2008), <www3.nd.edu/~wzech/advancedphysicslab.html>.

¹⁰W. R. Leo, *Techniques for Nuclear and Particle Physics Experiments*, 2nd, revised ed. (Springer-Verlag, Berlin, 1994).

¹¹Jeff C. Bryan, *Introduction to Nuclear Science*, 3rd. ed. (CRC Press, Boca Raton, FL, 2018).

¹²Kenneth S. Krane, *Introductory Nuclear Physics*, 3rd. ed. (John Wiley and Sons, N.Y. N.Y., 1988).

¹³Ernest M. Henley and Alejandro Gracia, *Subatomic Physics*, 3rd. ed. (World Scientific, Singapore, 2007).

¹⁴A. Das and T. Ferbel, *Introduction to Nuclear and Particle Physics*, 2nd. ed. (World Scientific, Singapore, 2003).

¹⁵Fairview Microwave, Lewisville, TX 75056.

¹⁶Glenn F. Knoll, *Radiation Detection and Measurement*, 4th ed. (John Wiley, N.Y., 2011).

¹⁷Nicholas Tsoulfanidis and Sheldon Landsberger, *Measurement and Detection of Radiation*, 3rd ed. (CRC Press, Boca Raton, FL, 2011).

¹⁸Gopal B. Saha, *Basics of PET Imaging*, 2nd ed. (Springer, N.Y., 2010).

¹⁹M. J. A. de Voigt, J. Dudek, and Z. Szymański, "High-spin phenomena in atomic nuclei," *Rev. Mod. Phys.* **55**, 949–1046 (1983).

²⁰David Ward and Paul Fallon, "High spin properties of atomic nuclei," *Adv. Nucl. Phys.* **26**, 167–291 (2001).

²¹D. Weisshaar *et al.*, "The performance of the γ -ray tracking array GRETINA for γ -ray spectroscopy with fast beams of rare isotopes," *Nucl. Instrum. Meth. Phys. Res. A* **847**, 187–198 (2017).

²²"Muon lifetime," TeachSpin, Buffalo, N.Y., <<https://www.teachspin.com/>>.

²³F. D. Becchetti, M. Febraro, R. Torres-Isea, M. Ojaruega, and L. Baum, "²⁵²Cf fission-neutron spectrum using a simplified time-of-flight setup: An advanced teaching laboratory experiment," *Am. J. Phys.* **81**, 112–119 (2013).

²⁴F. H. Frohner, "Evaluation of ²⁵²Cf prompt fission neutron data from 0 to 20 MeV by Watt spectrum fit," *Nucl. Sci. Eng.* **106**, 345–352 (1990).

²⁵F. D. Becchetti, "Comment on '²⁵²Cf fission-neutron spectrum using a simplified time-of-flight setup: An advanced teaching laboratory experiment [Am. J. Phys. **81**, 112–119 (2013)]'," *Am. J. Phys.* **82**, 206 (2014).

²⁶F. D. Becchetti, R. S. Raymond, R. O. Torres-Isea, A. Di Fulvio, S. D. Clarke, S. A. Pozzi, and M. Febraro, "Recent developments in deuterated scintillators for neutron measurements at low-energy accelerators," *Nucl. Instrum. Meth. Phys. Res. A* **874**, 72–78 (2017).

²⁷N. P. Zaitseva *et al.*, "Recent developments in plastic scintillators with pulse shape discrimination," *Nucl. Instrum. Meth. Phys. Res. A* **889**, 97–104 (2018).

²⁸Sidney A. Katz and Jeff C. Bryan, *Experiments in Nuclear Science* (CRC Press, Boca Raton, FL, 2011).

²⁹Randolph S. Peterson, *Experimental γ Ray Spectroscopy and Investigations of Environmental Radioactivity* (Spectrum Techniques, Oak Ridge, TN, 1996).

³⁰F. D. Becchetti, M. Febraro, J. Riggins, and R. O. Torres-Isea, "A multi-functional apparatus for α and β spectroscopy utilizing a permanent ring-magnet β spectrometer," *Am. J. Phys.* **84**, 883–893 (2016).

³¹MCA-8000D, Amptek, Inc., Bedford, MA 01730.

³²RX1200 Digital Spectrometer, Avicenna Instruments LLC, Corvallis, OR 97330.

³³UCS-30 and ICS PCI Computer Spectrometers, Spectrum Techniques LLC, Oak Ridge, TN 37830.

³⁴Analog Devices EVAL-AD9680, <www.analog.com>.

³⁵Signatec-Dynamic Signals LLC, Lockport, IL 60441.

³⁶StemLab 125-14 by RedPitaya, <www.redpitaya.com>.

³⁷C. M. Lavelle, "Gamma ray spectroscopy with the Arduino UNO," *Am. J. Phys.* **86**, 384–394 (2018).

- ³⁸Geza Kolumban, “Software defined electronics: A revolutionary change in design and teaching paradigm of RF radio communication systems,” *ICT Express* **1**, 44–54 (2015).
- ³⁹Adrian C. Melissinos and Jim Napolitano, *Experiments in Modern Physics*, 2nd ed. (Academic Press, NY, 2013).
- ⁴⁰Jason Engbrecht and Nathaniel Hillson, “A flexible positron spectrometer for the undergraduate laboratory,” *Am. J. Phys.* **86**, 549–558 (2018).

- ⁴¹Prof. A. Di Fulvio, University of Illinois, Department of Nuclear Engineering (private communication, 2019).
- ⁴²Nathan Damron, Frederick D. Becchetti, and Ramon O Torres-Isea, “Applications of digital pulse acquisition systems and software defined electronics in advanced teaching labs,” *Bull. Am. Phys. Soc.* **63**, 12 (2018); <<http://meetings.aps.org/link/BAPS.2018.HAW.HA.80>>.
- ⁴³See supplementary material at <https://doi.org/10.1119/1.5125128> for digitizer and SDE-related information including sample SDE programs.



Electric Fly

When this device is connected to a Wimshurst machine or other electrostatic generator, the arms begin to revolve with the points facing to the rear. The presence of the highly-charged points results in a locally-large electric field, and charge flies off the points. This gives a kick to the apparatus, and it revolves. A nice demonstration: easily built, entertaining as well as being scientifically interesting and not often used today. This apparatus is at the physics department at the University of Utah. (Picture and text by Thomas B. Greenslade, Jr., Kenyon College)

Applications of High-speed Digital Pulse Acquisition and Software-Defined Electronics (SDE) in Advanced Nuclear Teaching Laboratories
F.D. Becchetti, N. Damron, and R. O. Torres-Isea,
Dept of Physics, U. of Michigan, Ann Arbor MI 48109

Since acceptance of our paper there have been several new developments that should be noted:

1) The CAEN CoMPASS DAQ/SDE software is now available for the DT5790 250 MS/s 12 bit series of digitizers. This allows one to more easily program the FPGA in the digitizer to perform specific functions such as coincidence measurements. A student-written set of instructions and SDE routines (written by UM undergrad Jesse Viola in Python) are given below. As noted in the paper, the DT5720 digitizer (a lower cost 250 MS/s 12 bit digitizer w/o detector power supplies) also can utilize ComPASS.

2) We mention the Red Pitaya STEMLAB 125-14 (125 MS/s 14 bit) digitizer as a lower-speed lower-cost alternate to the CAEN and XIA 250 MS/s 12 bit digitizers for certain experiments not requiring high-speed digitizing. An Am. J. Phys article by Vretenar et al. (just published in the December 2019 issue), which was accepted at same time as our SDE article but unknown to us, describes application of the Red Pitaya STEMLAB 125-14 digitizer for a nuclear lifetime measurement. However, as indicated in our paper, the 8 nsec sampling is too coarse for many other nuclear experiments compared with the 4 nsec sampling interval of the more expensive CAEN and XIA digitizers. In many experiments one needs to accurately integrate a fast pulse, e.g. from a PMT anode, to get good energy resolution, and this requires a high sampling rate (i.e. many short sampling intervals). In this case the number of digitizing bits is less important and 12 bits is usually adequate. Fortunately, Red Pitaya in the near future will be releasing a 250 MS/s 12 bit digitizer (below) which will make it more useful for many more nuclear experiments than the 125-14 bit Red Pitaya (check their website for updated information).

NEWS

16.08.2019

[Back to list](#)

NEW STEMLAB 250-12 PRODUCT DEVELOPMENT

At Red Pitaya we heavily relate to our community's opinion and this is no different when we are launching new products. We listen to the needs and wants of our users and tailor our products to best fit these needs. At Embedded World in February we announced the development of a completely new STEMLab product with improved analog front-end performance. The new STEMLab 250-12 provides more real-time processing capabilities due to the bigger FPGA. Some preliminary information on key performance is available below. The product prototypes are currently in production and will be tested at our premises soon. The product is supposed to be released in late 2019/early 2020.

We are always happy to hear your opinions and this time is no exception. If you have any ideas or proposals let us know and we will make sure to pass it on to our development team!

	Property	Value	Comment
General	Sampling rate	250 Msps	
	Number of channels	2 inputs, 2 outputs,	4 BNC connectors on

3. In addition to references cited in the paper, there are many good references on digital signal processing in general:

- a) Steven W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing* (California Technical Publishing, 1997)
- b) Richard G. Lyons, *Understanding Digital Signal Processing, 4th ed.*
- c) Richard G. Lyons, et al. , *The Essential Guide to Digital Signal Processing*
- d) *Digital Signal Processing, 2nd ed.* (Schaum's Outlines,2012)
- e) A.V. Oppenheim and R.W. Schafer , *Discrete-Time Signal Processing, 3rd ed.* (Pearson, 2014)
- f) A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing* (Pearson,2015)
- g) J. G. Proakis et al. , *Digital Signal Processing, 4th ed.*

A. Additional notes on CAEN and XIA table-top high-speed digitizers for nuclear-physics and other teaching applications.

In the AJP paper we evaluated and utilized primarily the CAEN DT5790N digitizer, but we now also have CAEN 5720 and XIA LLC Pixie-net digitizers under student evaluation. These are all 2-4 channel digitizers of comparable costs (4-6 k\$) and have vendor-supplied on board or PC SDE software available. Some of these table-top digitizers (DT5790 as well as lower-speed DT5780) also have detector HV supplies as well as preamp supplies so are more stand-alone units. They can output time stamped list-mode data, set for coincident events if desired, for post-processing with vendor-supplied or custom SDE software in a PC. CAEN supplies SDE software (CoMPASS and MCASq..) to facilitate post-processing, and likewise XIA (see documentation at their websites). CAEN's CoMPASS software is a user- friendly GUI interface for programming the digitizer to generate time-stamped list mode coincident or non-coincident data suitable for post-processing with SDE software.

The trade off in low-cost digitizers is sampling rate versus bit resolution. This dictates which digitizer is suitable for certain experiments. Scintillators do not have high pulse height resolution i.e. energy resolution (typically a few % FWHM) but can be very fast. Hence a 12 bit, 250 MS/s digitizer works well giving pulses digitized to 4 nsec time steps suitable for ToF measurements, fast coincidence or PSD. However, 12 bits (4096 channels) may be too little for Hi-res SiSB and HP Ge detectors (but normally adequate for teaching). A 14 bit but slower (100 MS/s) DT 5780 or XIA equivalent or RedPitaya STEMLAB 125-14 can be used, but will

have limited timing accuracy (8-10 nsec), hence less PSD, ToF and other capability. However, the DT5780 and similar units support the CAEN DPP-PHA MCA software which is well suited for high-resolution HPGe/Si spectroscopy (if needed). DPP-PHA is not supported (as of now) on the CAEN 12 bits digitizers (DT5790, DT5720, etc). High bit resolution, high sample rate digitizers while available are much more expensive (x2-x3 12 bit digitizers).

In the attached pages we list some of the presently available (Aug 2019) CAEN, XIA and Red Pitaya table-top digitizer specs and software (incl. firmware) available including SDE software. Although CoMPASS GUI software is now available for DT5790/80 digitizers, the present CoMPASS manual primarily describes CoMPASS applications for a multi-channel, research-level 5730 series digitizer which has features not present in the DT5720,5790 and 5780 digitizers. (e.g. CFD timing). See attached info. on CoMPASS as well as related XIA SDE software.

Likewise, while there are no application manuals per se at moment for the Pixie-Net digitizer, it can use much of the same XIA SDE software developed for the Pixie-4e digitizer board (see attached info.).

As noted in the paper, with both CAEN, XIA and many other digitizers, various data files can be specified for storing the data in a PC, either the full digitized waveforms or data pre-processed on board in the digitizer to give a list of time-stamped integrated pulses, PSD, pulse integrals, etc. including coincident data if specified. (See CAEN manual on setting coincident gates e.g. from user guide on setting coincidences, below, or in the CoMPASS manual, and similar XIA documentation).

However, unless one has the needed technical expertise to implement vendor-supplied research-level SDE software, it is probably preferable to use custom experiment-specific PC-based software using data stored in list mode. The post-processing SDE software can then be written in Python, Root, Matlab, LabView, etc. and as shown in the paper, this is generally adequate for most teaching purposes. [Some samples of student-written post-process SDE software (Python) are included at the end of this material].

In general, then, most post-processing SDE teaching software will utilize the list mode time-stamped data that can be generated with the digitizers, either 1 or 2 single channels, often with coincidence set between two or more channels. Once transferred to a PC or MAC, these files (see sample in paper and below) can be analyzed with custom-SDE programs written for specific experiments using

standard data analysis or other software (as shown in the sample programs). Hence as indicated, Root, MatLab as well as Python, and other DAQ/Analysis software packages including CAEN CoMPASS in many cases can be used for post processing, graphical display, and analysis of the data.

B. Setting DT5790 registers and 2 channel coincidence (with DPP-PSD software). We include this for completeness, but it is now possible to do this via a GUI interface using the latest version of CAEN CoMPASS software (now available for both the DT5790, DT5720 as well as other CAEN digitizers) :

How to configure the registers - DT5790 with DPP-PSD

The register configuration for DT5790 is the same as for the 720 series (refer to Sec. **How to configure the registers - 720 and 751 series with DPP-PSD**), apart for the "Trigger Validation Mask" register, where address 0x8188 corresponds to channel 0, and address 0x818C corresponds to channel 1. Moreover bit[2] must be used for channel 0 and bit[3] must be used for channel 1. Refer to **[RD10]** for additional details.

Register name	Address	Register bits	Options
Trigger Validation Mask	0x8188 (ch0) 0x818C (ch1)	[2]	0 TRG_REQ[0] not propagated 1 TRG_REQ[0] propagated
		[3]	0 TRG_REQ[1] not propagated 1 TRG_REQ[1] propagated
		Operation Mask [9:8]	00 OR 01 AND 10 MAJORITY
		Majority Level [12:10]	value for majority
		[27:13]	0 (Reserved)
		External Trigger [30] Software Trigger [31]	0 EXT TRG not propagated 1 EXT TRG propagated 0 SOFT TRG not propagated 1 SOFT TRG propagated

Examples for DT5790

Coincidence between channel 0 and channel 1 (ch0 & ch1)

Channel 0 and channel 1 acquire data within 40 ns of coincidence window:

```
# Syntax:
# GENERIC_WRITE 0x<address> 0x<data> 0x<mask>

GENERIC_WRITE 0x8000 0x4 0x4
#
GENERIC_WRITE 0x1080 0x40000 0xC0000
GENERIC_WRITE 0x1180 0x40000 0xC0000
#
GENERIC_WRITE 0x1070 0x5 0xFF
GENERIC_WRITE 0x1170 0x5 0xFF
#
GENERIC_WRITE 0x106C 0x9 0xFF
GENERIC_WRITE 0x116C 0x9 0xFF
#
GENERIC_WRITE 0x8188 0x10C 0xFFFFFFFF
GENERIC_WRITE 0x818C 0x10C 0xFFFFFFFF
```

C. CAEN table-top digitizers and software presently available, late 2019 (check CAEN website for updates):

Digitizer families

CAEN Digitizers are platform independent instruments housing high speed (up to 5 GS/s) multichannel ADCs with local memory and FPGAs for real-time data processing.

Available in different form factors, VME, NIM and Desktop, they are the result of all the experiences collected by CAEN in signal processing and data acquisition systems and have been designed driven by physics and related applications as Medical Imaging, Homeland Security and Environmental Monitoring.

 <p>720 Digitizer Family 12-bit @ 250 MS/s 2 / 4 / 8 Channels</p>	 <p>724 Digitizer Family 14-bit @ 100 MS/s 2 / 4 / 8 Channels</p>	 <p>725 Digitizer family 14-bit @ 250 MS/s 8 / 16 Channels</p>
 <p>730 Digitizer Family 14-bit @ 500 MS/s 8 / 16 Channels</p>	 <p>740 Digitizer Family 12-bit @ 62.5 MS/s 32 / 64 Channels</p>	 <p>742 Digitizer Family 12-bit @ 5 GS/s (Switched Capacitor) 16+1 / 32+2 Channels</p>
 <p>743 Digitizer Family 12-bit @ 3.2 GS/s (Switched Capacitor) 8 / 16 Channels</p>	 <p>751 Digitizer Family 10-bit @ 1-2 GS/s 4-2 / 8-4 Channels</p>	 <p>761 Digitizer Family 10-bit @ 4 GS/s 1 / 2 Channels</p>

Note: DT5790 (12 bits) and DT5780 (14 bits) have built-in detector and preamp supplies. As noted CAEN CoMPASS SDE software is now available for most of the DT series.

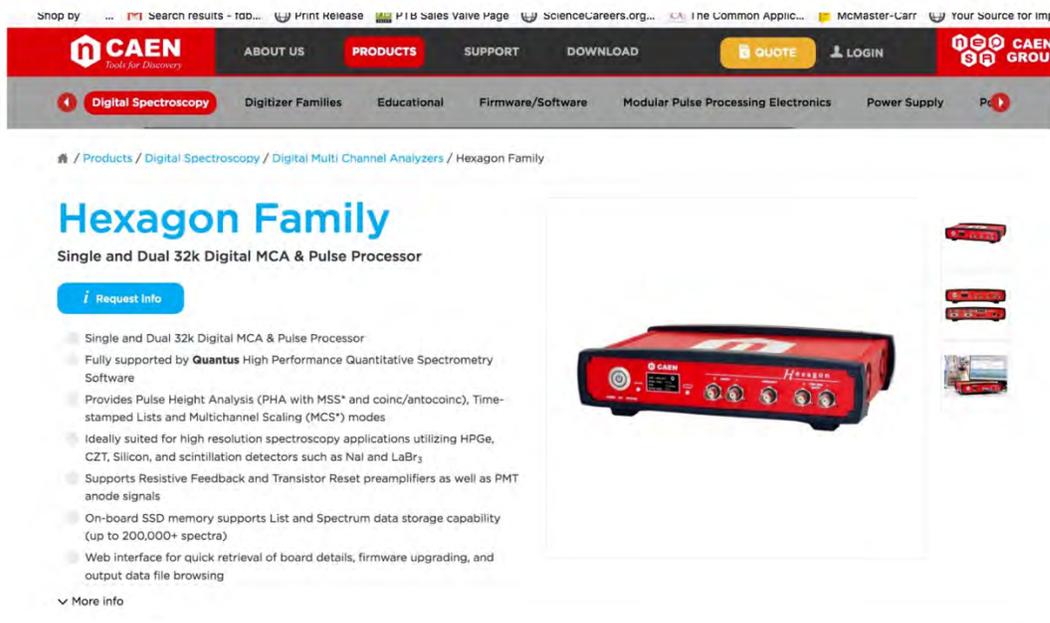
The DT5790N digitizer is the one we mainly used in the Am.J. Phys. SDE paper: 12 bits (4096 chnls), 250 MS/s (4 ns/sample). It supports DPP-PSD software and, now, CoPASS.

The DT5720 comes with various options. We are evaluating the 4 channel version, 12 bits, 250 MS/s which presently supports both DPP-PSD and CoPASS GUI software. It lacks HV supplies tho, but CoPASS simplifies creation of list-mode data files including coincidence data. It is < 4k\$ vs 6-7 k\$ for the DT5790.

The DT5780 is a 14 bit, 100 MS/s digitizer best suited for high resolution HPGe and Si spectroscopy and supports DPP-PHA SDE software. Usually 12 bits is adequate for teaching labs (with a >200 MS/s rate needed for ToF and other applications). It has 2 chnls. with HV/LV power supplies like the DT5790. The digitizer is similar to that used in the RedPitaya STEMLAB 125-14 digitizer.

Another 14 bit 100 MS/s desktop digitizer but w/o HV/LV supplies is the DT5724 Available with 4 chnls. This is similar to the 2 channel Red Pitaya STEMLAB 125-14 digitizer/

This is a new CAEN 32K channel digitizer for high-resolution spectroscopy.



The screenshot shows the CAEN website's product page for the Hexagon Family. The navigation bar includes 'ABOUT US', 'PRODUCTS', 'SUPPORT', 'DOWNLOAD', 'QUOTE', and 'LOGIN'. The breadcrumb trail is: Home / Products / Digital Spectroscopy / Digital Multi Channel Analyzers / Hexagon Family. The main heading is 'Hexagon Family' with the subtitle 'Single and Dual 32k Digital MCA & Pulse Processor'. A 'Request Info' button is present. The product features are listed as follows:

- Single and Dual 32k Digital MCA & Pulse Processor
- Fully supported by **Quantus** High Performance Quantitative Spectrometry Software
- Provides Pulse Height Analysis (PHA with MSS* and coinc/anticoinc), Time-stamped Lists and Multichannel Scaling (MCS*) modes
- Ideally suited for high resolution spectroscopy applications utilizing HPGe, CZT, Silicon, and scintillation detectors such as NaI and LaBr₃
- Supports Resistive Feedback and Transistor Reset preamplifiers as well as PMT anode signals
- On-board SSD memory supports List and Spectrum data storage capability (up to 200,000+ spectra)
- Web interface for quick retrieval of board details, firmware upgrading, and output data file browsing

Below the text is a large image of the red Hexagon Family digitizer unit, with a vertical strip of smaller images showing different views or models of the device.



CAEN has made a big effort in developing algorithms for the Digital Pulse Processing (DPP).

The user can install a DPP algorithm on the FPGA of the digitizer (firmware upgrade), run it on-line and implement new acquisition methods that go beyond the simple waveform recording.

Firmware Upgrade

720 Digitizer Family



Firmware



Waveform Recording Firmware

Time windowed waveform recording
Dead-timeless acquisition
Multi-buffer memories
Multi-board synchronization



Pulse Shape Discrimination

Digitizer running DPP-PSD firmware becomes a digital replacement of dual gate QDC, discriminator and gate generator.

Software



CAEN Digitizer readout application



Digitizer Software for Signal Inspection and waveform recording



Multiparametric DAQ Software for Physics Applications

Features

ICH

WV

TS

QDC

PSD

ZS

ICH

WV

TS

QDC

PSD

ZS

Feature Acronyms Legend

- ICH** (**I**ndependent **c**hannels): the channels can trigger and acquire data independently and can be correlated using a user-defined (anti) coincidence logic criteria
- WV** (**W**aveforms): acquisition of a programmable number of ADC samples (raw waveforms)
- TS** (**T**imestamp): trigger coarse time stamp with low resolution (10-20 ns)
- QDC** (**C**harge to **D**igital **C**onverter): gated integrator (charge)
- PSD** (**P**ulse **S**hape **D**iscrimination): particle discrimination via double gated integration
- ZS** (**Z**ero **S**uppression): acquisition of a programmable number of ADC samples with zero suppression

🏠 / Products / Firmware/Software / Software / Readout Software / CoMPASS

CoMPASS ★ New

Multiparametric DAQ Software for Physics Applications

[Manuals](#) [Downloads](#)

- ☐ Software for simultaneous DPP acquisition, including [DPP-PHA](#), [DPP-PSD](#) and [DPP-QDC](#) (Coming Soon)
- ☐ Multi-board management
- ☐ Synchronization of multiple boards even from different families. Time Correlation between different channels
- ☐ Simultaneous plot of waveforms, energy, time, PSD, and TOF spectra
- ☐ Energy calibration
- ☐ Digital Constant Fraction Discrimination for fine time stamp interpolation (pico second intrinsic resolution)
- ☐ Selectable filters on energy, time and PSD
- ☐ Advanced data saving options:
 - ☐ Data from board is recorded for the whole acquisition run
 - ☐ Time ordered recording of channels acquired data (list mode)



▼ More info

Overview

Overview

CAEN Multi-Parameter Spectroscopy Software (CoMPASS) is the DAQ software from CAEN able to implement a Multiparametric Data Acquisition for Physics Applications: the detectors can be connected directly to the digitizers/MCAs inputs and the software acquires energy, timing, and PSD spectra at the same time.

CoMPASS software has been designed as a user-friendly interface to manage the acquisition with all the CAEN DPP algorithms. It allows an easy setting of the acquisition parameters and to display up to six different plots and histograms at the same time.

CoMPASS can manage multiple boards and allows an easy synchronization of multiboard systems. Among the most important features, CoMPASS allows to implement event correlation between different channels (in hardware and/or software), apply energy, PSD and time selections, calculate and show the acquisition statistics (trigger rates, data throughput, percentage of discarded events due to the selections, etc...), perform a basic mathematical analysis of the recorded spectra (ROI selection, background subtraction, peak fitting, etc), save the output data files (raw data, lists, waveforms, spectra) and use the saved files to run offline with different processing parameters.

The user familiar with the ROOT Analysis Framework, CoMPASS provides also the possibility of saving the output files (lists waveforms and spectra) in the ROOT TTree format for an easy post processing with user made analysis code. CoMPASS software supports CAEN x720, x724, x725, x730, x751 digitizer families running the DPP-PSD and DPP-PHA, and the x781 MCA family.

Support for x740D with DPP-QDC, to the CAEN [DT5780](#) MCA and to the [DT5790](#) Analyzer is coming soon.

Applications

- Nuclear spectroscopy
- Clover detectors
- HPGe, Silicon Drift Detectors
- Neutron physics with scintillation detectors
- Multiple boards synchronization
- Homeland security
- Precise Timing for Time of Flight measurements

Related Products



DT5720
4 Channel 12 bit 250 MS/s Digitizer



DT5724
4 Channel 14 bit 100 MS/s Digitizer



DT5725
8 Channel 14-bit 250 MS/s Digitizer

Related Firmware



DPP-PHA
Digital Pulse Processing for the Pulse Height Analysis



DPP-PSD
Digital Pulse Processing for Charge Integration and Pulse Shape Discrimination

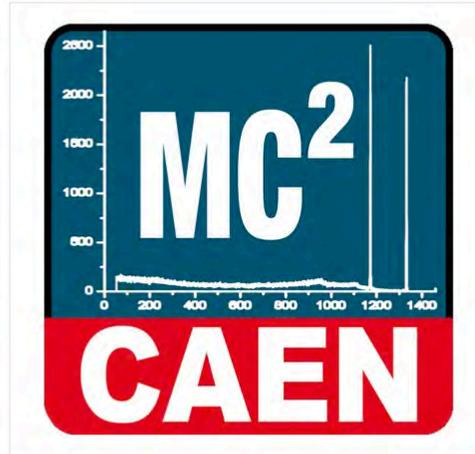
MC² Analyzer ★ New

Graphical software tool for digitizers running DPP-PHA firmware

📖 Manuals

↓ Downloads

- Designed for:
 - 724, 725 and 730 Digitizer families
 - 770/780/781 Digital MCA families, Hexagon and Gamma stream
- Trapezoidal filter replacing shaping amplifier and peak sensing ADC
- Online baseline restoration and ballistic effect correction
- Online pile-up correction for live-time measurement
- PHA and time-stamped list mode available
- Full setting of all the relevant DPP-PHA parameters and power supplies for DT5780, Hexagon and Gamma stream
- Complete simultaneous control of multiple boards
- Basic mathematical analysis on collected spectra (peak search, background subtraction, peak fitting, etc.)



∨ More info

Overview

Overview

MC² Analyzer is a software specifically designed to manage CAEN Digital MCAs (780/781 families, Hexagon, DT5770, and Gamma stream) as well as CAEN digitizers running DPP-PHA firmware, like 724, 725 and 730 families.

The DPP-PHA firmware implements a digital trapezoidal filter on the input pulse, which replaces the traditional analog chain of shaping amplifier and peak sensing ADC. The MCA is therefore directly connected to the charge sensitive preamplifier, with no need of additional devices. The PHA algorithm is able to perform online baseline restoration, ballistic effect corrections, and to manage the pile-up for the live time information. PHA and time-stamped list acquisition modes are available.

MC² Analyzer software allows the user to program the relevant DPP-PHA parameters, to manage the HV channels configuration (x780, Hexagon and Gamma stream only), to collect the spectra and perform basic mathematical analysis, like energy calibration, peak search, background subtraction, peak fitting, etc.

The software is designed with multi-channel and multi-board capabilities: it can handle several boards and manage the data acquisition from each of them at the same time.

Supported Operating Systems:	Windows (32-64 bit), Linux (32-64 bit)
Language:	C#
Supported Products:	VME, NIM and Desktop Digitizers
Supported Comm. Interface:	USB, Optical Link, VME
Additional Software Required:	CAENDigitizer, CAENComm, CAENVMELib, Gnuplot (Linux only)

Applications

- Nuclear spectroscopy
- HPGe, silicon drift, silicon strip detectors
- Slow scintillation detectors like NaI(Tl)
- Anti-compton shielding
- Homeland security
- Environmental survey
- Ion beam analysis
- Nuclear medicine

D. XIA PIXIE-Net digitizer

PIXIE-Net



Instruments That Advance The Art

PIXIE-Net

4 channel 250 MHz Desktop Digital Spectrometer

FEATURES

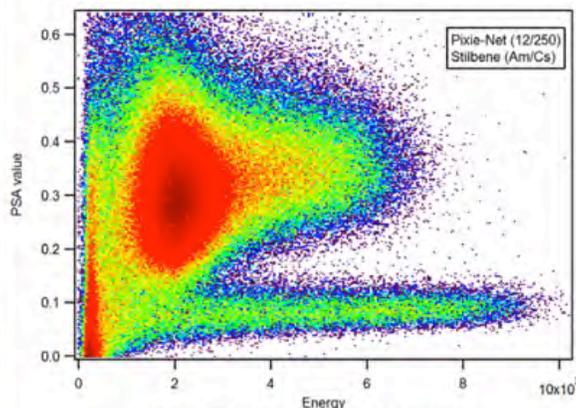
- Compact standalone detector readout electronics.
- 4 channels, 12 bit, 250 MSPS pulse processor.
- 32K on-board MCA spectrum per channel.
- Waveform capture and Pulse Shape Analysis.
- Sub-nanosec Timing Resolution.
- Embedded Linux platform.
- Front and back panel digital I/O signals.
- Gbit Ethernet, local USB storage, web interface.
- Approx. 8x8x14cm, 1 kg, 18 W
- Low cost



OVERVIEW

The PIXIE-Net is a multi-channel digital spectrometer for radiation detectors in desktop format with integrated networking and USB resources. Designed for high-precision coincidence gamma-ray spectroscopy using HPGe detectors, scintillators, or silicon detectors, the PIXIE-Net offers not only waveform digitization and capture but also pulse height measurements, on-board energy histograms, time stamping and constant fraction timing, pileup inspection, external gating and online pulse shape analysis. Besides nuclear spectroscopy, the PIXIE-Net can be used for neutron/gamma discrimination, time-of-flight measurements, and coincidence/anti-coincidence measurements.

The PIXIE-Net combines an embedded Linux system with XIA's digital pulse processing developed on the PIXIE PXI spectrometer series. Incoming signals are digitized at 250 MSPS with 12 bit ADCs. Using a ZYNQ All Programmable SoC with programmable logic (FPGA) and dual-core ARM[®] processor on the same chip, the ADC data stream is captured and processed in the FPGA section and the ARM processor writes the results to a removable USB drive or makes them available on a web server. Multiple units can share signals through GPIO.



Pulse shape analysis (alpha/gamma discrimination) using crystal Stilbene

APPLICATIONS

- Scintillator, HPGe, Si detectors
- Real-time Pulse-Shape Discrimination
- Clover detectors
- Distributed Data Acquisition
- Homeland Security
- OEM systems
- Remote monitoring

XIA LLC

www.xia.com

sales@xia.com Tel: +1-510-401-5760

31057 Genstar Road, Hayward, CA 94544 USA

SPECIFICATIONS

Front/Rear Panel I/O

- 4 analog signal inputs. Impedance 50 Ω or 2 k Ω .
- 10 digital inputs / outputs for triggers or veto signals
- 5V power supply and control for active PMT bases
- Simple pulser for system check and diagnostics
- RJ45 Ethernet
- USB type A (for peripherals)
- USB type micro-B (for UART control from PC)

Zynq “System on Chip” Platform

- Dual-core ARM[®] Cortex™-A9 with Xilinx 7-series FPGA section
- Gigabit Ethernet, 1GB RAM, USB2.0, 16GB SD
- PMOD expansion port (for Wi-Fi, GPS, GPIO, I2C, SDI, LEDs)
- Linux based on Ubuntu 15
- Local file storage
- IEEE 1588 timing compatible

Digital Controls

- Analog gain and digital gain adjustment
- Energy filter (0.048 - 63.4 μ s).
- Trigger filter and threshold
- Waveform length and delay
- Event acceptance: Hit pattern, external gating, coincidence window, input delays

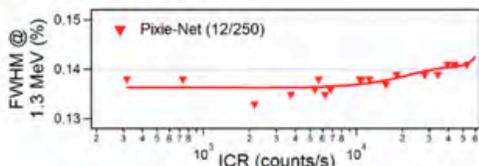
Pulse Processing

- Signal digitized at 250 MSPS, 12 bit. (inquire for higher rate/precision, or more channels)
- Energy filter operated at 125 MHz.
- Real time pulse shape analysis (charge integration)
- Real time constant fraction timing

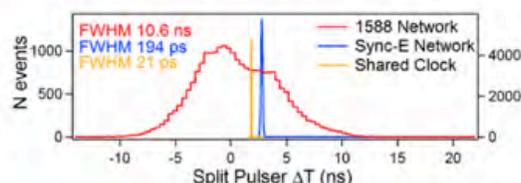
Data Reported

- Energy spectra (32K per channel).
- List mode data (energies, timestamps, and waveforms).
- Run statistics.

PERFORMANCE



Energy resolution as a function of input count rate.



Timing resolution within a Pixie-Net (shared clocks) and between two Pixie-Nets (network synchronization).

SOFTWARE

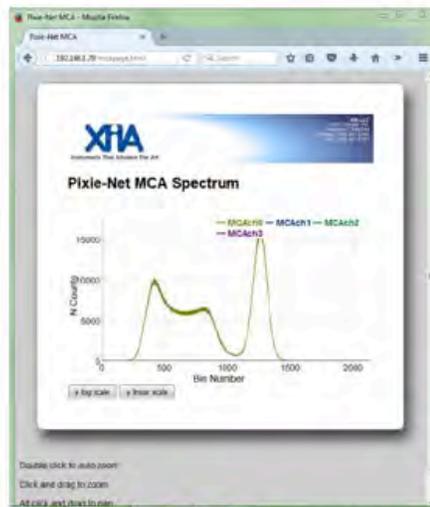
The Pixie-Net runs a customized version of Ubuntu 15. Many applications from the Linux universe are installed (gcc, ftp, http, ssh, etc); more can be added by the user. A number of simple programs read results and waveforms from the FPGA section, format and store them to file, and make them available on a web server. Settings and results can be managed through a web interface provided by XIA, or with plug-ins for common acquisition/analysis tools (LabVIEW, MATLAB, Igor Pro, etc). All parameters can be saved to disk for easy switching between applications.

The Pixie-Net is a standalone data acquisition system, eliminating the need for a host PC and software drivers to control and readout a USB or PXI peripheral. Data acquisition can be set up, started/stopped, and monitored over the network, with data saved to local storage or shared over the network.

Options for operation include:

- Remote login to the embedded Linux system via a UART/SSH terminal program or via Remote Desktop
- Monitor MCA spectra and run statistics via a web page hosted by the Pixie-Net Linux system
- Save data to mounted network drive or local USB drive
- Download MCA spectra, run statistics, and list mode data files from web server
- Integrate web API into front end GUI built on analysis software

Multiple Pixie-Net units can form a distributed data acquisition system synchronized over the data network. Each Pixie-Net can locally collect data with timestamps synchronized to sub-microsecond (or even sub-nanosecond) precision using IEEE 1588 clock synchronization or synchronous Ethernet, implemented in the Zynq Ethernet controller or a secondary Ethernet PHY.



XIA LLC

www.xia.com

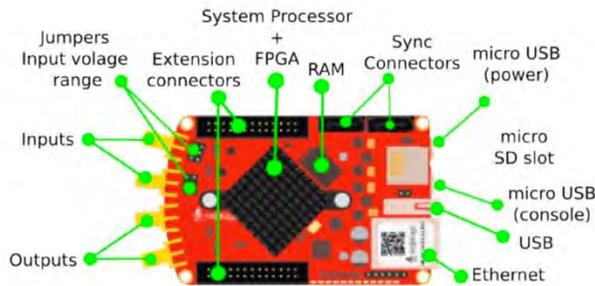
sales@xia.com Tel: +1-510-401-5760

31057 Genstar Road, Hayward, CA 94544 USA

E. Red Pitaya STEMLAB 125-14 Digitizer:

STEMLab 125-14 (originally Red Pitaya v1.1) vs. STEMLab 125-10

STEMLab is available in two versions which both offer the same functions and features, the difference is in the technical specification of high-frequency inputs and outputs, RAM capacity and some other differences (find more info in the comparison table below). They are addressed to target different groups and / or needs. Where STEMLab 125-14 has 14bit input / output channels for highly accurate measurement results in professional environments, STEMLab 125-10 has 10bit input / output channels and is perfect for universities, students and makers.



	STEMLAB 125-10	STEMLAB 125-14
Basic	Processor	Processor DUAL CORE ARM CORTEX A9 DUAL CORE ARM CORTEX A9
	FPGA	FPGA Xilinx Zynq 7010 SOC Xilinx Zynq 7010 SOC
	RAM	256MB (2Gb)
	System memory	Micro SD up to 32GB
	Console connection	USB to serial converter required
	Power connector	Micro USB
	Power consumption	1.5A

	STEMLAB 125-10	STEMLAB 125-14
Connectivity	Ethernet	10bit
	USB	USB 2.0
	WIFI	requires WIFI dongle
	Synchronisation	-

F. Testing digitizers and software using fast pulser, delays, and attenuators.

As noted in the text, fast scintillator PMT signals can conveniently be simulated using a Phillips 417 battery-powered fast negative signal pulser. This signal can be used to check the operation of the digitizer and signal processing software provided its amplitude, via suitable GHz attenuators if needed, is kept below the input range of the digitizer (typically 1-2 v). Coincident signals can be simulated by using a 50 ohm signal splitter (as available from many vendors) and splitting the pulser signal into two signals. One can also use a simple BNC tee but the

pulses may have "ringing" due to impedance matching issues as the pulser is designed for a 50 ohm load. ToF spectra can then be simulated by using a set of appropriate 50 ohm cables of various lengths or a passive variable delay module to add delay in one signal path. The latter set up can also be used to calibrate a SDE TDC if the cable delays are known. (One can often find passive delay boxes on eBay..or make one using LEMO cables. Good quality switches are important though, e.g. slide switches are not as reliable as toggle switches). Phillips scientific makes suitable, passive delay and attenuator modules (below).

Phillips Scientific NIM Pocket Pulser **MODEL 417**

FEATURES

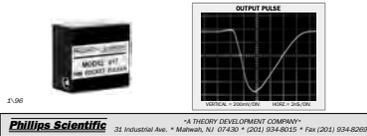
- * Current Switching NIM Output Pulse
- * Operates Direct Coupled into 50 Ohm Load
- * Long Battery Life - Three Years
- * 60Sec Pulse Width at 10 KHz Rate
- * Small Size - Low Cost

DESCRIPTION

The NIM pocket pulser is a compact, lightweight battery operated pulse generator. It provides a convenient and portable way of simulating fast negative photomultiplier pulses or NIM logic pulses, to easily check cables and electronics in the laboratory or remote experimental sites. The model 417 generates a negative pulse resembling a photomultiplier with the risetime of 1.5nsec and width of 60nsec at a 10 KHz rate. An output is present when a load of 10K ohms or less is connected. With no load the pulser goes to a standby condition with a battery life in excess of three years. The fully compatible fast NIM output is typically 16mA (800Volts across 50ohms), with no DC offset present. The current switching output will not be damaged when testing shorter cables.

SPECIFICATIONS

Output Rise and Falltimes : Typically 1.5nSec Rise/Time, 5nSec Fall/Time.
Repetition Rate : Typically 10 KHz.
Amplitude : Typically 800mVolts for resistive loads of 50 ohms to 10K ohms.
Connector Type : LEMO, female.
Battery Life : Over 1000 hours for continuous use. Over three years for standby (no load).
Battery Type : 3 Volt Lithium coin battery, comes with Panasonic number BR2325 or BR2320H. The Radio Shack equivalent part number is BR2320H (catalog no. 23-163).
Dimensions : 1.375" x 1.250" x .675. (3.5 cm x 3.18 cm x 1.72 cm)
Care and Service : No serviceable parts other than battery replacement; The positive side of the battery faces out.



1/06
Phillips Scientific "A THEORY DEVELOPMENT COMPANY"
 31 Industrial Ave. • Mahwah, NJ 07430 • (201) 934-8010 • Fax (201) 934-8269



GHz attenuator and 50 ohm splitter

Phillips precision delay and attenuator modules:

Phillips Scientific Quad Rotary Attenuator **NIM MODEL 804**

DESCRIPTION

The Phillips Scientific Model 804 is a quad, 50 ohm, rotary PI attenuator, packaged in a single width NIM module. It is configured with 12 positions, ten calibrated steps from .1 to 1.0 attenuation in .1 increments, and an OFF position. An additional open position has been included to allow the user to add another value of attenuation if desired. The 804 employs a high quality sealed switch that maintains uniform performance in the presence of dust or other contaminants.

SPECIFICATIONS

Attenuation Range and Steps :

Position	Gain
1	.1
2	.2
3	.3
4	.4
5	.5
6	.6
7	.7
8	.8
9	.9
10	1.00
11	OPEN
12	OFF

Attenuation Accuracy : ±0.5% of setting.
Impedance : 50 ohms ±1% for all positions except 11.
Frequency Range : DC to 1000 MHz, 3db point.
Voltage Standing Wave : 1.05:1 to 250 MHz, 1.10:1 to 500 MHz, 1.20:1 to 1 GHz.
Average Power : 0.5 Watt.
Indexing : 30° Continuous rotation.
Operating Temp. Range : -20° C to +75° C.
Connector Types : BNC connectors are standard; LEMO or SMA female are optional. Specify when ordering.
Packaging : Standard single width NIM module in accordance with TID-20893 and Section ND-S24.
Quality Control : 36 hour cycled burn-in with switched power cycles.



4/96
Phillips Scientific "A THEORY DEVELOPMENT COMPANY"
 31 Industrial Ave. • Mahwah, NJ 07430 • (201) 934-8010 • Fax (201) 934-8269

Phillips Scientific Dual NanoSecond Delay Module **NIM MODEL 792**

FEATURES

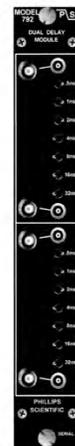
- * Two Independent Delay Sections
- * Total Delay of 127nSec in .5nSec Steps
- * Monotonicity is Guaranteed for Each Section
- * Both Lemo and BNC Connectors
- * Available in CAMAC Packaging Model 7192

DESCRIPTION

The Model 792 uses coaxial cable to provide an accurate and stable means of generating delays to 127nSec in a single width NIM module. Since coaxial cable is employed, no power is required and both linear or logic signals may be delayed with excellent fidelity. The configuration of two identical channels allows two signals to be delayed independently, often necessary for coincidence applications. Each channel has seven switches which select the calibrated delays of .5, 1.2, 4, 8, 16 and 32nSec. The total delay is equal to the sum of the switch settings.

GENERAL PERFORMANCE

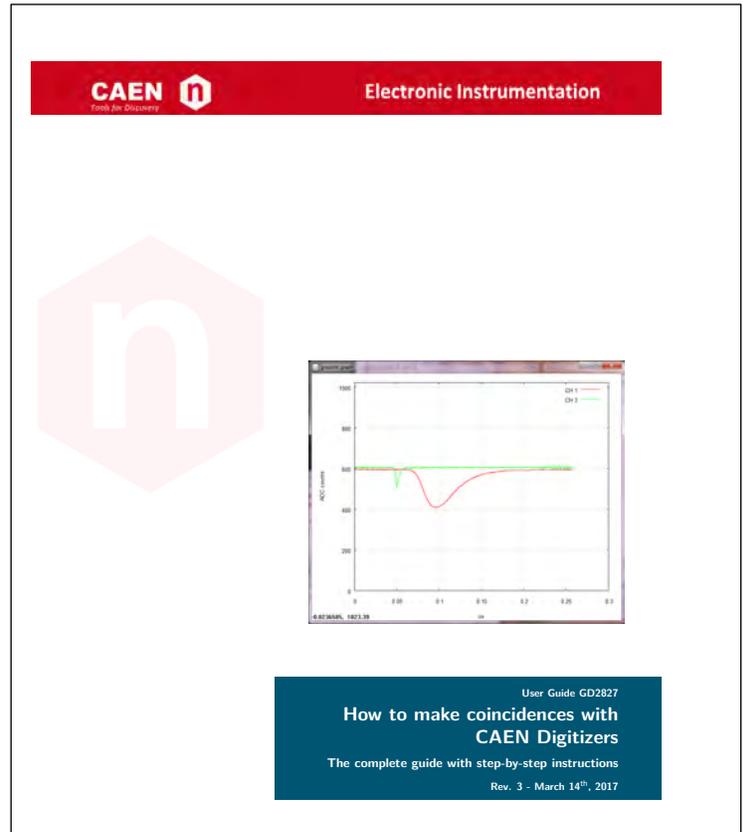
Input and Output Characteristics : 50 ohm impedance throughout; Accepts signals to ±350 Volts; Input and output ports are interchangeable; LEMO and BNC connectors on both ports.
Delay Controls : Seven toggle switches per channel select .5, 1.2, 4, 8, 16 and 32nSec. The delay equals the sum of the switch settings.
Delay Range : Selectable from 0 to 63.5nSec in .5nSec steps; 127nSec with both channels cascaded.
Delay Accuracy : Better than ±50pSec or ±.5%, whichever is greater, for each delay switch inserted.
Insertion Delay : 1.5nSec; ±100pSec per section, with all switches in the OFF position.
Risetime and Falltime : $t_r = t_f = .775nSec + (.20pSec \times \text{X delay settings})$, from 10% to 90%.
Maximum KSWR : 1.02 : 1 to 100MHz, 1.05 : 1 to 250MHz, 1.10 : 1 to 500MHz.
Maximum Attenuation : Loss @ 1 (1 db) + (# of switches IN) + (.03 db X delay setting) @ 100MHz.
Operating Temperature : 20° C to +85° C ambient.
Packaging : Standard single width NIM module in accordance with TID-20893 (Rev).



7/96
Phillips Scientific "A THEORY DEVELOPMENT COMPANY"
 31 Industrial Ave. • Mahwah, NJ 07430 • (201) 934-8010 • Fax (201) 934-8269

F. CAEN CoPASS and related software information

As noted, CAEN provides extensive documentaion on using the CAEN DPP-PSD /DPP-PHD and CoPASS GUI software, as well as documentation for setting up coincidence measurements, and other features. Updated documentation is available on the CAEN web site. XIA provides similar documentation for XIA Pixie-Net applications and SDE. However as noted the present documentation for CoPASS refers to the 5730 digitizers which have features *not* present in the desk- top models.



Related CAEN Software available (check CAEN website for updates):

Firmware/Software

Firmwares, Library and Software developed for CAEN Hardware

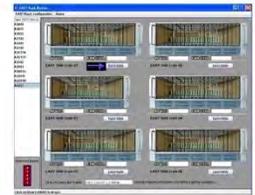
Software



Configuration Tools



Control Software

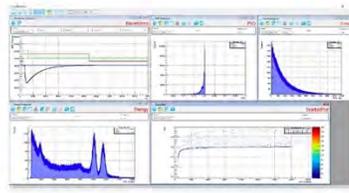


EASY Configuration files

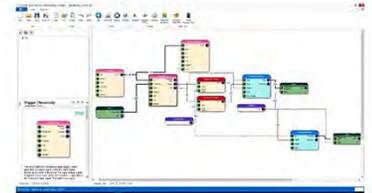


EPICS

EPICS Input/Output Controller for SYx527 Systems and Power Supply Modules



Readout Software



Open FPGA Tools



LabVIEW Instrument Drivers

LabVIEW™ Instrument Driver for CAEN NIM, Desktop and Rack mounting Power supplies

G. Sample student-written SDE programs

As SDE programs are developed and implemented at UM we plan to post samples of these at the publication section of our research website:

<http://research.physics.lsa.umich.edu/twinsol/>

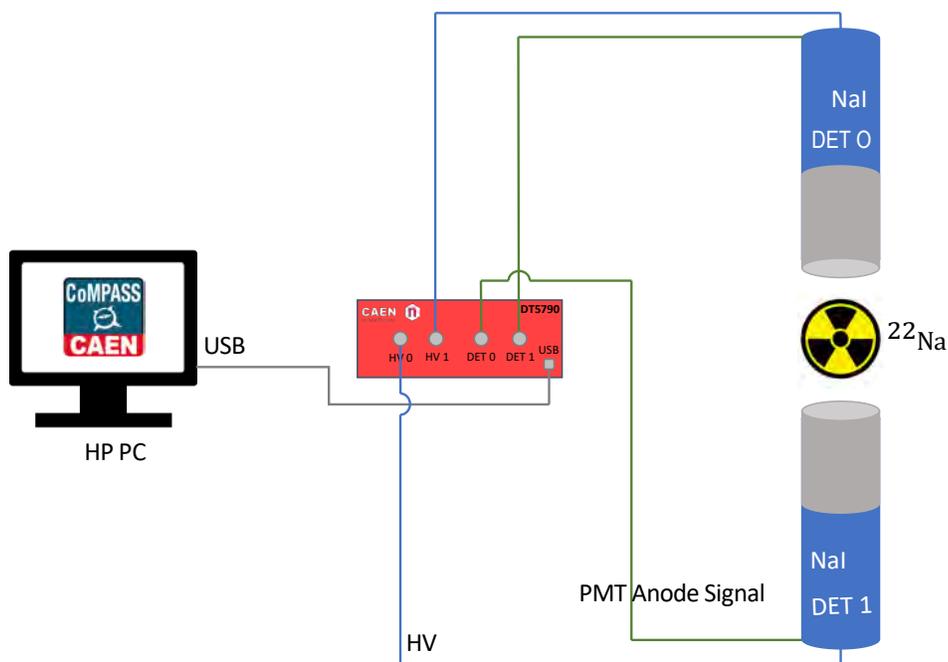
Likewise, Prof. Angela DiFulvio , graduate student Fang Ming and others at the U. Illinois have written SDE for the DT 5720 and other digitizers useful for teaching. The SDE programs (mostly in ROOT) are posted at their GitLab website: <https://gitlab.engr.illinois.edu/nml/pals.git> or via email : difulvio@illinois.edu

Attached are samples of student-written SDE programs (by UM students Nathan Damron and Jesse Viola) using Python and the Pandas library. Those written by Mr. Damron were used to acquire most of the data shown in the AJP paper. Nathan was an EE UM undergrad and therefore had the expertise needed to configure the digitizer and wrote the PC-based Python SDE programs used for various experiments. His programs include energy calibrations and other features not necessarily needed in a basic post-processing program. Jesse is an LSA undergraduate and has written a simpler but adequate 2D coincidence acquisition and display program, again in PYTHON.

The Python programs use Pandas library and generate post- SDE software for Na22 and C060 gamma-gamma decay, as well as Compton scattering. The SDE software processes list-mode data files set with or w/o coincidence requirement, as appropriate, with subsequent analysis, display and gating of the 1D and 2Ddata.

CAEN DT5790 ^{22}Na Gamma-Gamma Coincidence Experiment

General gamma-gamma coincidence setup:



Software/hardware

- Windows 10
- CAEN CoMPASS 1.3 and DPP-PSD
- CAEN DT5790 digitizer with USB driver

CoMPASS Settings

Acquisition Mode:

- Acquisition modeⁱ – List only
- Timed run (s) – 10
- List saving – Save filtered data, CSV file format

Settingsⁱⁱ:

- Use all factory settings (see below for list of factory settings)

Time Selection:

- Correlation – Paired AND
- Correlation window (μs) – 0.1

ⁱ 'List only' does not allow waveforms to be visualized.

ⁱⁱ Settings are all equivalent for CH0 and CH1.

Sample Data File

```
TIMETAG;ENERGY;ENERGYSHORT;FLAGS  
538540000;95;21;0x0  
685136000;98;17;0x0  
2218124000;161;23;0x0  
2541228000;102;17;0x0  
2672752000;337;46;0x0  
...
```

²²Na Gamma-Gamma Coincidence Experiment Procedure

The setup for this experiment consisted of two NaI(Tl) detectors connected to the CAEN DT5790 digitizer. The 1 μ Ci Na-22 source was placed between the two detectors at an equivalent distance from each detector. In the initial setup, the Na-22 sample was about 3 cm from each detector, and the detectors were 180° apart. Each 2 in. by 2 in. NaI scintillator, PMT and base was connected to a high-voltage output at approximately 1500 V [To match gains, Detector 1 (CH1) was set at approximately 100 V higher than Detector 0 (CH0)]. Using the CAEN CoMPASS software, a coincidence window was set at 0.1 μ s, thus filtering out data points that were outside of the time window. The data file produced (.csv file) includes timestamps, energy long and short pulse integration, and flags. Visualization software written in Python (see below for programs) was then used to create histograms to represent the energy long pulse integration values. Using gating in the Python program, the energy peak can be separated from lower energy coincidences in order to have an accurate count (N) of high-energy gamma coincidence events. In this experiment, the energy peak can be calibrated to 511 keV, which is the energy of positron annihilation gamma rays released from the decay of Na-22. To confirm that the positron annihilation gamma rays are released 180° apart, the two detectors were moved to 45° apart. This change resulted in the predicted effect of decreasing the number of high-energy coincidence events (see Fig. 2. Left). Increasing the distance between the detectors also decreases the number of coincidence events, including the random coincidences (see Fig. 2. Right, for each detector 6 cm away from the source). The decrease in coincidence events, and radioactive intensity in general for a point source, can be described by the inverse-square law: $I \propto \frac{1}{r^2}$.

Figures:

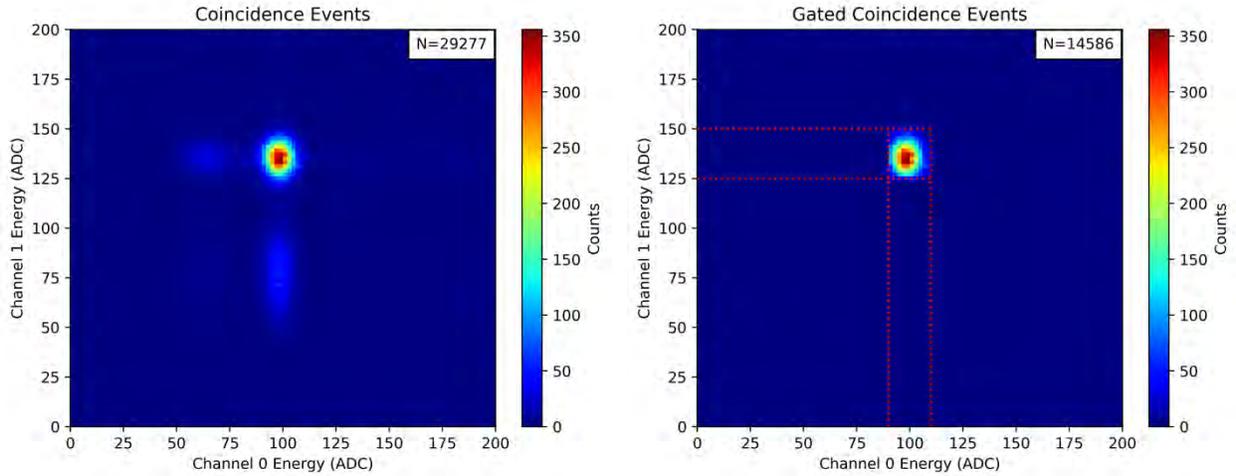


Fig. 1. Left: 2D histogram Na-22 gamma-gamma coincidence events with two detectors (CH0 and CH1) 180° apart in a 100 s interval. Each detector is placed about 3 cm from the Na-22 source. Right: Gated version of the Fig. 1. Left ($90 < \text{CH0 Energy} < 110$, $125 < \text{CH1 Energy} < 150$). The number of events, N , is shown in the top right of each figure.

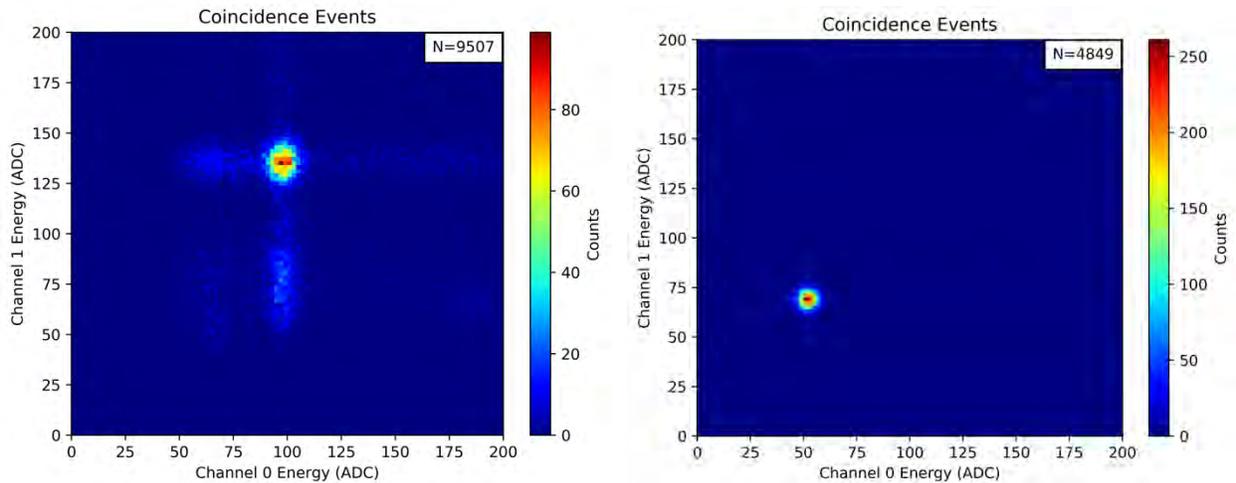


Fig. 2. Left: 2D histogram Na-22 gamma-gamma coincidence events with two detectors (CH0 and CH1) 45° apart in a 100 s interval. Each detector is placed about 3 cm from the Na-22 source. Right: 2D histogram Na-22 gamma-gamma coincidence events with two detectors (CH0 and CH1) 180° apart in a 100 s interval. Each detector is now placed about 9 cm from the Na-22 source, and attenuation is added.ⁱⁱⁱ The number of events, N , is shown in the top right of each figure.

ⁱⁱⁱ Channel (0 and 1) energy scale is approx. half of Fig. 2 Left channel (0 and 1) energy scale due to the use of a 50Ω splitter.

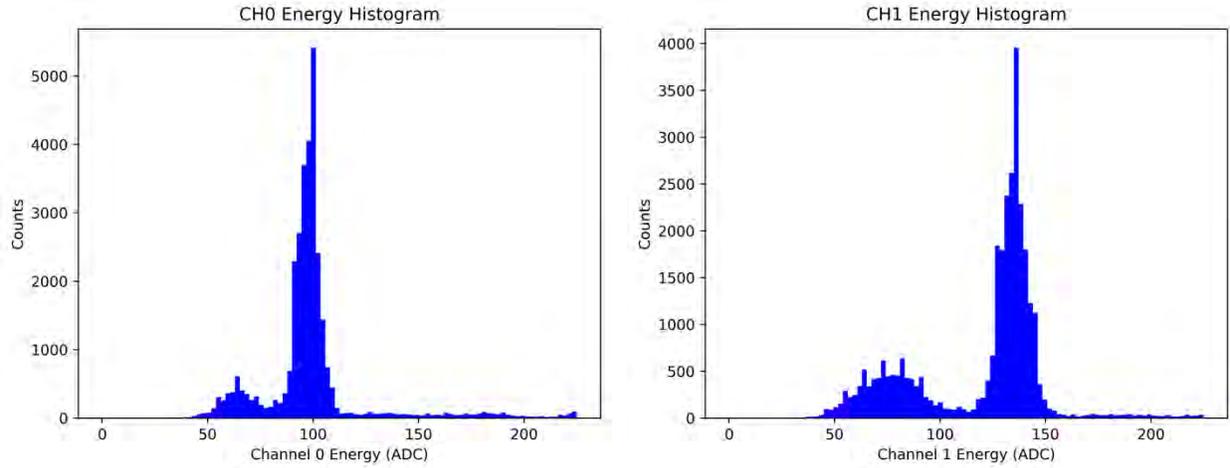


Fig. 3. Uncalibrated energy spectra of Na-22 gamma rays for each detector (CH0 and CH1) 180° apart in a 100 s interval. Each detector is placed about 3 cm from the Na-22 source (see Fig. 1).

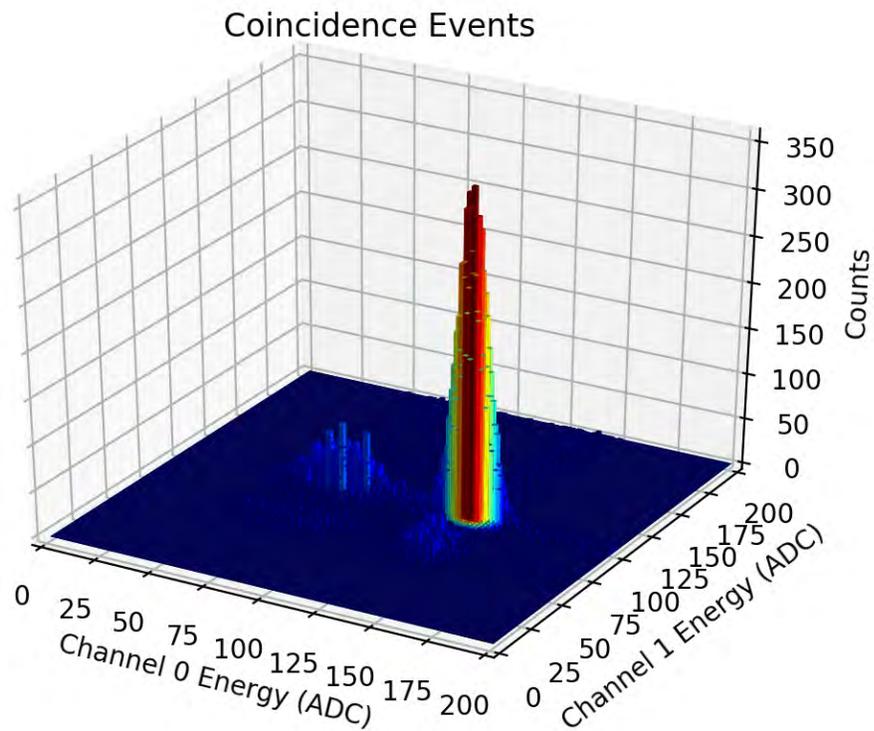


Fig. 4. 3D histogram of Na-22 gamma-gamma coincidence events with two detectors (CH0 and CH1) 180° apart in a 100 s interval. Each detector is placed about 3 cm from the Na-22 source (see Fig. 1).

Python Programs

1D Energy Histogram:

```
#Initialization
import matplotlib.pyplot as plt
import numpy as np
import csv

#Array
x = []

#.csv Reader
with open('FILE_NAME.csv') as csvfile:
    csv_reader = csv.reader(csvfile, delimiter=';')
    for lines in csv_reader:
        x.append(lines[1])

#Delete Label
del x[0]

#Create Float Array
x1 = np.array(x)
x1 = np.asfarray(x1, float)

#1D Histogram (change range to fit data)
plt.title('Energy Histogram')
plt.xlabel('Channel 0 Energy (ADC)')
plt.ylabel('Counts')
plt.hist(x1, bins=100, range=[0, 225], facecolor='blue')

#Save Histogram
plt.savefig('CH#_Energy_Histogram_ADC.png', dpi=500, bbox_inches='tight')

plt.show()
```

2D & 3D Energy Histograms:

```
#Initialization
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
import numpy as np
import pandas as pd
import csv

#Gate Input (keV) (X0 < X1, Y0 < Y1) (change gating to fit data)
X0 = 0
X1 = 225
Y0 = 0
Y1 = 225

#Calibration Input (keV) (change calibration to fit data)
E = 1.0

#Bins (Resolution)
B = 100

#Arrays
x = []
y = []

#CH0 .csv Reader
with open('FILE_NAME0.csv') as csvfile:
    csv_reader = csv.reader(csvfile, delimiter=';')
    for lines in csv_reader:
        x.append(lines[1])

#CH1 .csv Reader
with open('FILE_NAME1.csv') as csvfile:
    csv_reader = csv.reader(csvfile, delimiter=';')
    for lines in csv_reader:
        y.append(lines[1])

#Delete Label
del x[0]
del y[0]

#Create Float Arrays
x1 = np.array(x)
x1 = np.asfarray(x1, float)

y1 = np.array(y)
y1 = np.asfarray(y1, float)

#Calibration
x2 = x1 * E
y2 = y1 * E

#Create Data Frames
```

```

colsx = ["Energy 0"]
dx2 = pd.DataFrame(x2, columns=colsx)

colsy = ["Energy 1"]
dy2 = pd.DataFrame(y2, columns=colsy)

#Merge Data Frames
DF = pd.concat([dx2, dy2], axis=1)

#Gated Data Frames
DF1 = DF[DF["Energy 0"] >= X0]
DF2 = DF1[DF1["Energy 0"] <= X1]

DF3 = DF2[DF2["Energy 1"] >= Y0]
DF4 = DF3[DF3["Energy 1"] <= Y1]

#Split Data Frame
x4 = DF4[["Energy 0"]]
y4 = DF4[["Energy 1"]]

#Create Arrays
x5 = x4.values
x5 = x5.transpose()
x5 = x5.flatten()
x6 = np.array(x5)
x6 = np.asfarray(x6, float)

y5 = y4.values
y5 = y5.transpose()
y5 = y5.flatten()
y6 = np.array(y5)
y6 = np.asfarray(y6, float)

#Events
A = len(x6)

#Graph Limits
G = Y1 / (225 * E)
H = X1 / (225 * E)

#2D Histogram
plt.title('Coincidence Events')
plt.xlabel('Channel 0 Energy (keV)')
plt.ylabel('Channel 1 Energy (keV)')
plt.hist2d(x2, y2, bins=B, range=[[0, 225 * E], [0, 225 * E]],
cmap=plt.cm.jet)
plt.colorbar(label='Counts')

#Save 2D Histogram
plt.savefig('Coincidence_Events_keV.png', dpi=500, bbox_inches='tight')

#Gated 2D Histogram
plt.title('Gated Coincidence Events')

```

```

plt.xlabel('Channel 0 Energy (keV)')
plt.ylabel('Channel 1 Energy (keV)')
plt.axvline(x=X0, c='r', ymax=G, linestyle=':')
plt.axvline(x=X1, c='r', ymax=G, linestyle=':')
plt.axhline(y=Y0, c='r', xmax=H, linestyle=':')
plt.axhline(y=Y1, c='r', xmax=H, linestyle=':')
plt.hist2d(x6, y6, bins=B, range=[[0, 225 * E], [0, 225 * E]],
cmap=plt.cm.jet)
plt.colorbar(label='Counts')

#Event Text
plt.text(0.98,.98,"N={}".format(A), bbox={'facecolor':'w','pad':5},
        ha="right", va="top", transform=plt.gca().transAxes )

#Save Gated 2D Histogram
plt.savefig('Gated_Coincidence_Events_keV.png', dpi=500, bbox_inches='tight')

#Gated 3D Histogram
xAmplitudes = x6
yAmplitudes = y6

x7 = np.array(xAmplitudes)
y7 = np.array(yAmplitudes)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

hist, xedges, yedges = np.histogram2d(x7, y7, bins=(B, B), range=[[0, 225 *
E], [0, 225 * E]])
xpos, ypos = np.meshgrid(xedges[:-1]+xedges[1:], yedges[:-1]+yedges[1:])

xpos = xpos.flatten()/2.
ypos = ypos.flatten()/2.
zpos = np.zeros_like (xpos)

dx = xedges [1] - xedges [0]
dy = yedges [1] - yedges [0]
dz = hist.flatten()

cmap = plt.cm.jet
max_height = np.max(dz)
min_height = np.min(dz)

rgba = [cmap((k-min_height)/max_height) for k in dz]

ax.bar3d(xpos, ypos, zpos, dx, dy, dz, color=rgba, zsort='average')
plt.title('Gated Coincidence Events')
plt.xlabel('Channel 0 Energy (keV)')
plt.ylabel('Channel 1 Energy (keV)')
ax.set_zlabel('Counts')
ax.set_xlim3d(0, 225 * E)
ax.set_ylim3d(0, 225 * E)

```

```
#Save Gated 3D Histogram  
plt.savefig('3D_Gated_Coincidence_Events_keV.png', dpi=500,  
bbox_inches='tight')  
  
plt.show()
```

CoMPASS Factory Settings^{iv}

Input

- Record length – 992 ns
- Pre-trigger – 100 ns
- Polarity – negative
- Ns baseline – 128 samples
- Fixed BLR – 0
- DC Offset – 20.0 %
- Input dynamic – 2.0 Vpp

Discriminator

- Threshold – 200 lsb
- Trigger holdoff – 496 ns

QDC

- Energy coarse gain – 160 fC/LSB
- Long gate – 300 ns
- Short gate – 80 ns
- Pre-gate – 48 ns

Spectra

- Energy N channels – 4096
- PSD N channels – 4096
- Time intervals N channels – 8192
- Time intervals T_{min} – 0.000 μs
- Time intervals T_{max} – 1000.000 μs
- Start/stop Δt T_{min} – 10000 ns
- Start/stop Δt T_{max} – 10000 ns
- 2D Energy N channels – 512
- 2D PSD N channels – 512

Rejections

- PSD low cut – 0.000
- PSD high cut – 0.000
- Time intervals low cut – 0 ns
- Time intervals high cut – 0 ns

Energy Calibration

- SET CALIBRATION VALUES

Sync/Trig

- Start mode – Software (asynchronous)
- TRIG OUT/GPO mode – Level 0
- Start delay – 0 ns
- Channel time offset – 0.000 ns

Miscellaneous

- Label – CH
- FPIO type – NIM
- Rate optimization – 1023

^{iv} Not including settings changed as needed in ²²Na gamma-gamma coincidence experiment

Compton SDE Program (Python by N. Damron):

```
# -*- coding: utf-8 -*-
"""
Created on Thu Jul 26 14:14:09 2018

@author: ndamron
"""

# -*- coding: utf-8 -*-
"""
Created on Mon Jul 9 17:00:42 2018

@author: nd246
"""

import matplotlib.pyplot as plt
from matplotlib import cm
import pandas as pd
import os

#Change directory
cwd = os.getcwd()
cwd

os.chdir(" *** data directory path *** ")

#open calibration files in Documents folder
cal_0 = pd.read_csv(' *** calibration file for 1st detector ***
', sep='\s+', header=None, skiprows=6, dtype = int)
cal_1 = pd.read_csv(' *** calibration file for other detector
*** t', sep='\s+', header=None, skiprows=6, dtype = int)

fig, ax = plt.subplots()

# the histogram of the data
n_0, bins_0, patches_0 = ax.hist(cal_0.as_matrix([1]), bins =
1000)
n_1, bins_1, patches_1 = ax.hist(cal_1.as_matrix([1]), bins =
1000)

#find peaks for calibration scaling
m_0 = max(n_0)
m_1 = max(n_1)
```

```

p_0 = [i for i, j in enumerate(n_0) if j == m_0]
p_1 = [i for i, j in enumerate(n_1) if j == m_1]
peak_0 = bins_0[p_0]
peak_1 = bins_1[p_1]

#scaling functions
def scale_0(t):
    return t * 1150/peak_0
    #return t

def scale_1(t):
    return t * 570/peak_1

#open file in Documents folder
#compton files for each angle
df0_0 = pd.read_csv('Cs137_30_ls_0.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df0_1 = pd.read_csv('Cs137_30_ls_1.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df1_0 = pd.read_csv('Cs137_40_ls_0.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df1_1 = pd.read_csv('Cs137_40_ls_1.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df2_0 = pd.read_csv('Cs137_50_ls_0.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df2_1 = pd.read_csv('Cs137_50_ls_1.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df3_0 = pd.read_csv('Cs137_60_ls_0.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df3_1 = pd.read_csv('Cs137_60_ls_1.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df4_0 = pd.read_csv('Cs137_70_ls_0.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df4_1 = pd.read_csv('Cs137_70_ls_1.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df5_0 = pd.read_csv('Cs137_80_ls_0.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df5_1 = pd.read_csv('Cs137_80_ls_1.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df6_0 = pd.read_csv('Cs137_90_ls_0.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
df6_1 = pd.read_csv('Cs137_90_ls_1.dat', sep='\s+', header=None,
skiprows=6, dtype = int)
#df7_0 = pd.read_csv('Cs137_100_ls_0.dat', sep='\s+',
header=None, skiprows=6, dtype = int)

```

```

#df7_1 = pd.read_csv('Cs137_100_ls_1.dat', sep='\s+',
header=None, skiprows=6, dtype = int)
#df8_0 = pd.read_csv('Cs137_110_ls_0.dat', sep='\s+',
header=None, skiprows=6, dtype = int)
#df8_1 = pd.read_csv('Cs137_110_ls_1.dat', sep='\s+',
header=None, skiprows=6, dtype = int)

#print(df2_0)
#print(df2_1)

#plt.figure()
data_0 = pd.DataFrame()
#df0_0[1].hist(bins =200)

data_0[0] = scale_0(df0_0[1])
data_0[1] = scale_1(df0_1[1])
#plt.figure()
#df0_0[1].hist(bins = 200)
#plt.savefig('Compton_30_0_single.png')
plt.figure()
#df0_1[1].hist(bins = 200)
data_0[1].hist(bins=200, range = (0, 700))
#plt.savefig('Compton_30_1_single.png')
#data_0 = data_0[data_0[1] < 14000]
#data_0 = data_0[data_0[0] < 14000]
#print(data_0)
#data_0.plot(x = 0, y = 1, kind = 'hexbin',cmap = cm.binary)
#plt.savefig('Compton_30_2D.png')

data_1 = pd.DataFrame()
data_1[0] = scale_0(df1_0[1])
data_1[1] = scale_1(df1_1[1])
#data_1 = data_1[data_1[1] < 14000]
#data_1 = data_1[data_1[0] < 14000]
#print(data_0)
#data_1.plot(x = 0, y = 1, kind = 'hexbin',cmap = cm.gist_ncar)

data_2 = pd.DataFrame()
data_2[0] = scale_0(df2_0[1])
data_2[1] = scale_1(df2_1[1])
#data_2 = data_2[data_2[1] < 14000]#gate off top
#data_2 = data_2[data_2[0] < 14000]#gate off top

```

```

#print(data_0)
#data_2.plot(x = 0, y = 1, kind = 'hexbin',cmap = cm.gist_ncar)

data_3 = pd.DataFrame()
data_3[0] = scale_0(df3_0[1])
data_3[1] = scale_1(df3_1[1])
#plt.figure()
#df3_0[1].hist(bins = 200)
#plt.savefig('Compton_60_0_single.png')
#plt.figure()
#df3_1[1].hist(bins = 200)
#plt.savefig('Compton_60_1_single.png')
#data_3 = data_3[data_3[1] < 14000]#gate off top
#data_3 = data_3[data_3[0] < 14000]#gate off top
#print(data_0)
#data_3.plot(x = 0, y = 1, kind = 'hexbin',cmap = cm.binary)
#plt.savefig('Compton_60_2D.png')

data_4 = pd.DataFrame()
data_4[0] = scale_0(df4_0[1])
data_4[1] = scale_1(df4_1[1])
#data_4 = data_4[data_4[1] < 14000]#gate off top
#data_4 = data_4[data_4[0] < 14000]#gate off top
data_4 = data_4[0:7000]
#print(data_4)
#data_4.plot(x = 0, y = 1, kind = 'hexbin',cmap = cm.gist_ncar)

data_5 = pd.DataFrame()
data_5[0] = scale_0(df5_0[1])
data_5[1] = scale_1(df5_1[1])
#data_5 = data_5[data_5[1] < 14000]#gate off top
#data_5 = data_5[data_5[0] < 14000]#gate off top
data_5 = data_5[0:8000]
#print(data_5)
#data_5.plot(x = 0, y = 1, kind = 'hexbin',cmap = cm.gist_ncar)

data_6 = pd.DataFrame()
data_6[0] = scale_0(df6_0[1])
data_6[1] = scale_1(df6_1[1])
data_6 = data_6[data_6[1] < 600]#gate off top
data_6 = data_6[data_6[0] < 500]#gate off top

df6_0[1] = scale_0(df6_0[1])
df6_1[1] = scale_1(df6_1[1])
#plt.figure()
#df6_0[1].hist(bins = 200,figsize = (18,12), range = (0,1000))

```

```

#plt.savefig('Compton_90_0_single.png')
plt.figure()
#df6_1[1].hist(bins = 200,figsize = (18,12), range = (0,1000))
#plt.savefig('Compton_90_1_single.png')
#data_6 = data_6[data_6[1] > 4000]
#data_6 = data_6[0:6000]
#print(data_6)
#data_6.plot(x = 0, y = 1, kind = 'hexbin',cmap = cm.binary)
#plt.savefig('Compton_90_2D.png')

#data_7 = pd.DataFrame()
#data_7[0] = df7_0[1]
#data_7[1] = df7_1[1]
#print(data_0)
#data_7.plot(x = 0, y = 1, kind = 'hexbin',cmap = cm.gist_ncar)

#data_8 = pd.DataFrame()
#data_8[0] = df8_0[1]
#data_8[1] = df8_1[1]
#print(data_0)
#data_8.plot(x = 0, y = 1, kind = 'hexbin',cmap = cm.gist_ncar)

#plot all sets overlapped
data = pd.DataFrame()
data[0] = data_0[0]
data[1] = data_0[1]
data = data.append(data_1, ignore_index = True)
data = data.append(data_2, ignore_index = True)
data = data.append(data_3, ignore_index = True)
data = data.append(data_4, ignore_index = True)
data = data.append(data_5, ignore_index = True)
data = data.append(data_6, ignore_index = True)
#data = data.append(data_7, ignore_index = True)
#data = data.append(data_8, ignore_index = True)
#data = data[data[0] < 10000]
#data = data[data[1] < 20000]
#data.plot(x = 0, y = 1,xlim = (0, 14000), ylim = (0, 14000),
kind = 'hexbin',cmap = cm.gist_ncar)
#plt.savefig('Compton_2D_color.png')
data.plot(x = 0, y = 1, xlim = (0,800), ylim = (0,800),kind =
'hexbin', gridsize = 1000, cmap = cm.binary, figsize = (18,12))
plt.savefig('Compton_2D_BW.png')
#plt.figure()

```

```

#df1_1[1].hist(bins = 200)

df1_0 = df0_0[df0_0[1] < 10000]
df1_1 = df0_1[df0_1[1] < 20000]

#df1_0[1].hist(bins = 100)
#plt.savefig('Compton30deg_plastic.png')
#plt.figure()
#df1_1[1].hist(bins = 100)
#plt.savefig('Compton30deg_NaI.png')

```

Na22-gamma -gamma coincidence (Python N. Damron):

```

# -*- coding: utf-8 -*-
"""
Created on Thu Aug  2 14:34:16 2018

@author: ndamron
"""

import matplotlib.pyplot as plt
from matplotlib import cm
import pandas as pd
import os

#Change directory
cwd = os.getcwd()
cwd

os.chdir(" *** data directory path *** ")

#open file in Documents folder
df_0l = pd.read_csv(' ***data file for 1st detector*** ',
sep='\s+', header=None, skiprows=6, dtype = int)
df_1l = pd.read_csv(' ***data file for other detector*** ',
sep='\s+', header=None, skiprows=6, dtype = int)
print(df_0l)

```

```

print(df_1l)

df_0l = df_0l[df_0l[1] < 20000]
df_1l = df_1l[df_1l[1] < 20000]
#df_1e[1].plot()
#df_0e[1].plot()

df_0l[1].hist(bins = 100, figsize = (18,12))
plt.savefig('Na-22_0_single.png')
plt.figure()
df_1l[1].hist(bins = 100, figsize = (18,12))
plt.savefig('Na-22_1_single.png')

dfl = pd.DataFrame()
dfl['channel 0 (bins)'] = df_0l[1]
dfl['channel 1 (bins)'] = df_1l[1]

#print(dfl)

#dfl.plot(x = 0, y = 1, kind = 'scatter')
#dfl['N (counts)'] = dfl.groupby(['channel 0 (bins)', 'channel 1
(bins)']).transform('count')
dfl.plot(x = 'channel 0 (bins)', y = 'channel 1 (bins)', kind =
'hexbin', cmap = cm.binary, gridsize = 1000, figsize = (18,12))

plt.savefig('Na-22_2D.png')

count_0 = df_0l[1].value_counts()
count_1 = df_1l[1].value_counts()

print(count_0)
print(count_1)

```

Co 60 gamma-gamma

```

# -*- coding: utf-8 -*-
"""
Created on Mon Nov 26 16:51:20 2018

```

```
@author: ndamron
''''
```

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from matplotlib import cm
import os

#Change directory
cwd = os.getcwd()
cwd

os.chdir(" *** data directory path *** ")

#open file in Documents folder
cal_0 = pd.read_csv(' *** calibration file for 1st detector ***
', sep='\s+', header=None, skiprows=6, dtype = int)
cal_1 = pd.read_csv(' *** calibration file for 2nd detector ***
', sep='\s+', header=None, skiprows=6, dtype = int)

cal_1 = cal_1[cal_1[1] < 60000]

#cal_0[1].hist(bins = 200)
#plt.figure()
#cal_1[1].hist(bins = 200)

counts_0 = cal_0[1].value_counts()
counts_1 = cal_1[1].value_counts()

peak_0 = counts_0.index[0]#calibration value for channel 1
peak_1 = counts_1.index[0]#calibration value for channel 1

print(peak_0)
print(peak_1)
#peak_0 = 511
#peak_1 = 511
def scale(t, channel):
    if channel == 0:
        return t * 1170 / peak_0
    if channel == 1:
        return t * 1170 / peak_1
    return t
```

```

data_0 = pd.read_csv(' ***data file for 1st detector*** ',
sep='\s+', header=None, skiprows=6, dtype = int)
data_1 = pd.read_csv(' ***data file for 2nd detector***',
sep='\s+', header=None, skiprows=6, dtype = int)

#gate at 500000
data_0 = data_0[:][:500000]
data_1 = data_1[:][:500000]

print(data_0)

data_0[1] = scale(data_0[1],0)
data_1[1] = scale(data_1[1],1)

new_data_0 = pd.DataFrame(np.linspace(0,0,num = len(data_1[0])))
new_data_1 = pd.DataFrame(np.linspace(0,0,num = len(data_1[0])))

new_data_0[1] = np.linspace(0,0,num = len(data_1[0]))
new_data_1[1] = np.linspace(0,0,num = len(data_1[0]))

#for index in data_0.index:
#    diff = data_0[0][index] - data_1[0][index]
#    if diff > 200:
#        print("channel 1 ts: ", data_0[0][index], "
channel 0 ts: ", data_1[0][index] )
#        data_0.drop(index)
#        data_0.reindex(np.linspace(0, len(data_0[0]), num =
len(data_0[0])))
#        print("channel 0 size: ", len(data_0[0]))
#    if diff < -200:
#        print("channel 1 ts: ", data_0[0][index], "
channel 0 ts: ", data_1[0][index] )
#        data_0.drop(index)
#        data_0.reindex(np.linspace(0, len(data_0[0]), num =
len(data_0[0])))
#        print("channel 1 size: ", len(data_1[0]))

i = 0
j = 0
data_0_size = len(data_0[0])
data_1_size = len(data_1[0])

data_1[1].hist(bins = 1000, range = (0, 1500))

```

```

plt.savefig('Co60_1D.png')

for index in data_0.index:
    if index + i > data_0_size - 1:
        break
    if index + j > data_1_size - 1:
        break

    diff = data_0[0][index + i] - data_1[0][index + j]
    print("Diff: ", diff)
    if np.abs(diff) < 200:
        if data_0[1][index+i] < 1400 and data_0[1][index+i] >
1220:
            new_data_0[0][index] = data_0[0][index+i]
            new_data_0[1][index] = data_0[1][index+i]
            new_data_1[0][index] = data_1[0][index+j]
            new_data_1[1][index] = data_1[1][index+j]
            #print(new_data_0)
            #print(new_data_1)
        else:
            print("bad")
            #print("data_0 index: ", data_0[0][index+i], "
index + 1: ", data_0[0][index+i+1])
            #print("data_1 index: ", data_1[0][index+j], "
index + 1: ", data_1[0][index+j+1])
            if diff > 0:
                j = j + 1
            if diff < 0:
                i = i + 1
            #print("data_0 index: ", data_0[0][index+i], "
index + 1: ", data_0[0][index+i+1])
            #print("data_1 index: ", data_1[0][index+j], "
index + 1: ", data_1[0][index+j+1])

#     if diff > 200:
#         print("channel 1 ts: ", data_0[0][index], "
channel 0 ts: ", data_1[0][index] )
#         data_0.drop(index)
#         data_0.reindex(np.linspace(0, len(data_0[0]), num =
len(data_0[0])))
#         print("channel 0 size: ", len(data_0[0]))
#     if diff < -200:
#         print("channel 1 ts: ", data_0[0][index], "
channel 0 ts: ", data_1[0][index] )

```

```

#         data_0.drop(index)
#         data_0.reindex(np.linspace(0, len(data_0[0]), num =
len(data_0[0])))
#         print("channel 1 size: ", len(data_1[0]))

#print("channel 0 length: ", len(data_0[0]), '\n', "channel 1
length: ", len(data_1[0]))
#print(data_0[0])
#data_0.drop(data_0.index[0])
#print("channel 0 length: ", len(data_0[0]), '\n', "channel 1
length: ", len(data_1[0]))
#print(data_0[0])
plt.figure()
new_data_0[1].hist(bins = 1000, range = (10, 1500))
plt.savefig('Co60_gate.png')

plt.figure()
new_data_1[1].hist(bins = 1000, range = (10, 1500))
plt.savefig('Co60_1D_gated.png')

new_data_0.to_csv('new_data_0.csv', sep = '\t')
new_data_1.to_csv('new_data_1.csv', sep = '\t')

data_0_counts = data_0[1].value_counts()
data_1_counts = data_1[1].value_counts()

plt.figure()
data_0[1].hist(bins = 2000, range = (0, 1500))
plt.savefig('Co60_ch0.png')

plt.figure()
data_1[1].hist(bins = 2000, range = (0, 1500))
plt.savefig('Co60_ch1.png')
data = pd.DataFrame()

data[0] = new_data_0[1]

data[1] = new_data_1[1]

print(data)
data.plot(x = 0, y = 1, kind = 'hexbin', xlim = (0,2500), ylim =
(0,2500), cmap = cm.gist_ncar, gridsize = 5000, figsize =
(10,10))
plt.savefig('Co60_2D.png')

```

END