上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

THESIS OF BACHELOR

论文题目：液氙中正比闪烁光的实验研究

学生姓名：谈安迪

学生学号：5080729020

专　　业：应用物理学

指导教师：季向东

学院(系)：致远学院

液氙中正比闪烁光的实验研究

摘要

本文报告的是以实验手段研究液氙探测器中的正比闪烁光。我们建造了一个由电子漂移室，阴极和阳极构成的液氙丝室。探测器中的直接闪烁光由一个放置在腔体外部的伽马射线源产生。加载在栅极、阴极和阳极上的高压在腔体内产生漂移和正比电场。当伽马射线与氙原子作用时，同时产生闪烁光和自由电子。闪烁光可用阴极下方的四个光电倍增管以及阳极上方的一个光电倍增管来探测。为了研究液氙探测器中的正比闪烁光的性质，我们再阴极和阳极之间以及在栅极上加上高压电场。信号产生的电荷将被一个电荷灵敏放大器所测量。本实验测量了正比闪烁光的大小与阳极丝上加载电压之间的关系，同时正比闪烁光的衰减也被用来判断液氙的相对纯度。

关键词： 暗物质探测，液氙探测器，正比闪烁光，

EXPERIMENTAL STUDY OF PROPORTIONAL SCINTILLATION IN LIQUID XENON

ABSTRACT

The main purpose of this thesis is to study proportional scintillation in the xenon liquid phase. A liquid xenon wire chamber with an electron drift region, a cathode and an anode has been constructed. In this experiment, the direct scintillation light is produced by a gamma ray source located outside the chamber. An electric field is produced by a high voltage applied by the meshes, cathode and anode inside the chamber. When a gamma ray interacts with a xenon atom, it produces both scintillation light and free electrons. The scintillation would be observed by four sensitive PhotoMultiplier Tubes (PMTs) below the cathode and one PMT at the top. High electric fields are required so as to observe proportional scintillation. The drifting charge is also measured by a Charge Sensitive Amplifier (CSA). Thus the onset of electron multiplication can be observed. The relationship between the amplitude of proportional scintillation and applied electrical potential on anode is investigated. The attenuation of the amplitude of proportional scintillation can also be used to measure the purity of xenon.

Keywords: dark matter detection, liquid xenon detector, proportional scintillation

CONTENT

1. Introduction

The main purpose of this thesis is to experimentally study proportional scintillation in the liquid phase of xenon. This same process was originally observed but never practiced by Miyajima et al [1] because of the limited technological capacity of his time. Thus, there are very few follow-up studies on this topic in the scientific literature. If proportional scintillation in liquid xenon becomes a useful detection signal, dark matter detectors could be constructed in an easier manner with higher sensitivity potential.

1.1 Dark Matter

A century ago, the two dark shadows of physics were cast to light with the encompassed discoveries of Einstein's Theory of General Relativity and Quantum mechanics. With this, there was a new perspective for deciphering and understanding nature. However, a new shadow darkens the academic research of modern physicists. In the last few decades, astrophysical observations, such as the velocity dispersions of galaxies, the gravitational lensing phenomena, and the measured distances of type Ia supernovae, strongly support the current standard model of the Universe in which about 96% of the energy density is formed by two mysterious dark components. The first of these dark components is believed to be a new type of matter, dark matter, which is non-luminous and non-baryonic, interacting with normal matter through the gravitational force. The second is called dark energy; much less is known about the latter, but it may be responsible for the accelerated expansion of the universe. However, with scientific advances, we now have experimental methods to unravel the mysteries of dark matter and dark energy. A leading candidate for this non-baryonic "dark" matter is the Weakly Interacting Massive Particle (WIMP). According to this assumption, these particles can be detected via their elastic collisions with nuclei of ordinary matter. Intense efforts are underway worldwide to realize more sensitive experiments, with increased target masses and improved background rejection capabilities.

1.2 Xenon as detector Medium

There are many types of detectors designed to detect the WIMPs. Among the suggested types of detectors, liquid noble gas detectors are one of the most promising types because of their size and sensitivity. Large homogeneous volumes can be instrumented with PMTs to measure the scintillation excitation of liquid xenon or liquid argon, and ionization can be measured through drifting after applying electric field to the volume. The ratio of the ionization to scintillation can be used to discriminate between electronic and nuclear recoils. Ionization scintillation dual-phase detectors are mostly used in this decade with the gas phase being used to generate a large proportional scintillation instead of the ionization. The sensitivity of the detector will be improved considerably by using this method. Many experiments are in currently in progress with additional competition expected, such as XENON100 collaboration in Gran Sasso in Italy and PANDAX collaboration in CJPL (China Jinping Underground Laboratory).

1.3 Motivation of this thesis

The 'bottleneck' of the much used dual-phase detective technology is the liquid leveling and the usage of large amount of low-radioactive cryostat VUV (Vacuum UltraViolet) PMTs. These problems obstruct the development of this technology for a larger detector. If the proportional scintillation signal can be produced in the liquid phase, the leveling will no longer be an issue. The first observation of proportional scintillation was made by M. Miyajima et al [1] [2] [3], who constructed a liquid xenon drift chamber with a proportional counter to measure the proportional ionization and proportional scintillation in liquid xenon as early as 1976. But at that time the liquid xenon technology was still at its infancy. The interests of their measurements were only concerned with the energy resolution and spatial resolution. Using with new technological developments, this thesis focuses on the properties of proportional scintillation and establishes proper operational parameters for large scale detector (e.g., 1 ton scale) in the future.

1.4 Experimental design

First, a liquid xenon chamber with several grids is constructed to carry out the experiment. Then a number of subsystems are constructed to support the measurement progress. These subsystems account for the following components of the experimental design: cryogenics, electronics, gas system and LN_2 emergency system,.

When gamma rays from an external source interact with the xenon atoms, they produce fast electrons. These electrons excite and ionize xenon atoms on their path. The excitations can be observed in the form of VUV light in wavelength 178 nm, whereas the ionization electrons are free to move under the influence of an electric field. However, some ionization electrons will recombine with Xe^+ ions and create light of the same wavelength as the direct scintillation (S1). The remaining free electrons drift toward the anode, where they can be detected with a CSA (Charge Sensitive Amplifier). The amount of recombination, and thus the free charge yield, depends on the ionization density of the original ionizing particle. It is different for nuclear recoils, such as in neutrons or electrons, normally resulting from gamma ray interactions. The difference in signal strength can be used to discriminate between the sources of radiations.

The charge signal can be detected on the anode with a CSA, but their noise contribution is of order 300 - 400 electrons equivalent. For WIMP searches the expected signal is much lower, of order few electrons, that is. it cannot be detected directly by an electronic amplifier. Under a strong electric field the electrons are accelerated resulting in excited Xe-gas atoms on their way, producing the so-called secondary scintillation, or proportional scintillation (S2). This light is then detected through PMTs. With this technique there is an apparent amplification of the signal, typical gain about few hundreds, and even single drifting electrons can be detected.

2. Experimental apparatus

To measure the proportional scintillation, the experimental setup is constructed with a small gridded ionization chamber, cryogenic system, gas system, LN_2 emergency system, data acquisition system, and slow control system.

2.1 GIC (Gridded Ionization Chamber)

The liquid xenon detector is a gridded ionization chamber. The sensitive volume is delineated by an anode and cathode. An electric field is generated by applying a high voltage power supply. To make the field in this volume homogeneous, several field shaping rings are installed. The scintillation will be observed by four vacuum VUV sensitive PMTs below the cathode, i.e. immersed into the liquid. The cathode has to be transparent for the light. As such it is formed by a grid of stretched wires. The anode is similarly formed by thinner stretched wires. A grid in front of the anode makes the observed electrical signal independent of interaction position. Finally, a metal plate above the anode (i.e. opposite from the grid) closes the field to the top, which makes the anode structure symmetric, like the field in a MWPC (Multi-Wire Proportional Chamber). Fig. 1 is the schematic structure of GIC. There is a square hole in the center of the plate to create a visible window, and it is soldered with stretched wire in order to not disturb the electric field. Another PMT is placed above the hole to observe the light.

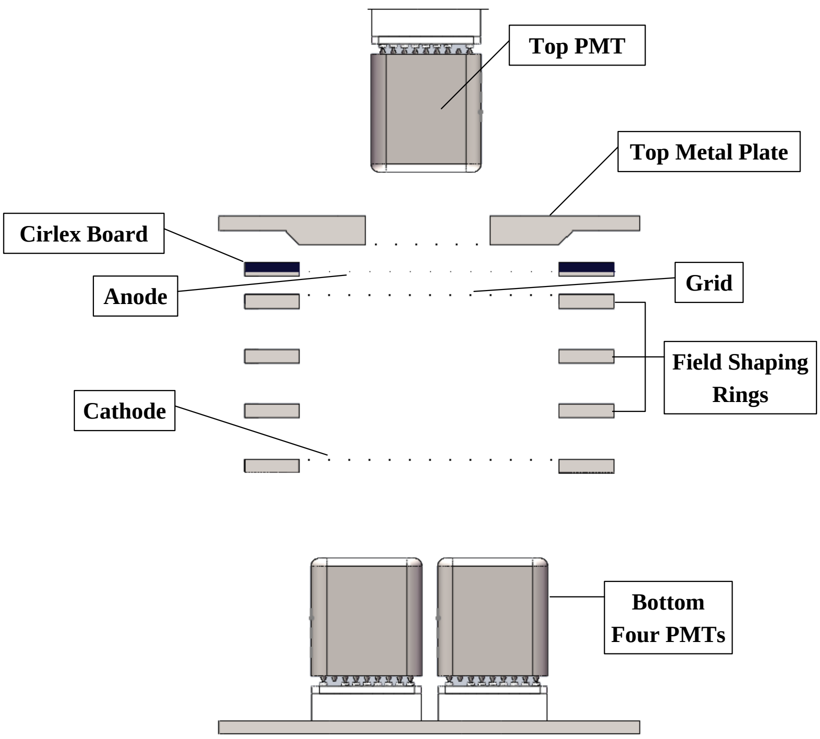


Fig. 1 Schematic structure of GIC

2.2 The Principle of MWPC

The basic structure of MWPC is a group of equidistant parallel anode wires in the middle of two parallel cathode planes or meshes. Fig. 2 is the schematic section of MWPC. The wires are paralleled to the Y axis (perpendicular to the paper); Z axis is perpendicular to the wire plane; l is the distance between the wire plane and the cathode plane; d is the space between the wires.

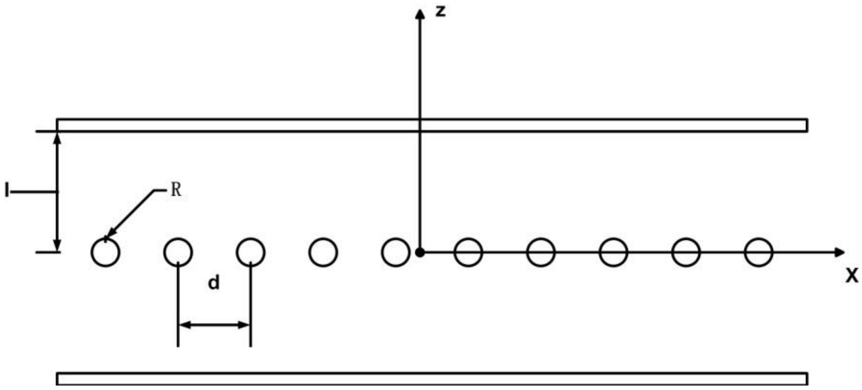


Figure 2 Schematic section of MWPC

Let the electrical potential at the surface of the wires be V_0, $V(R) = V_o$; and the potential at the cathode is zero, $V(l) = 0$. Assume that the wires are long enough, and then we can calculate the distribution of electrical potential [8]:

$$V(x,z) = \frac{CV_0}{4\pi\varepsilon_0}\left\{\frac{2\pi l}{d} - ln\left[4\left(sin^2\frac{\pi x}{d} + sinh^2\frac{\pi z}{d}\right)\right]\right\} \quad .$$

The electric field is the gradient of the potential:

$$E(x,z) = \frac{CV_0}{4\pi\varepsilon_0}\sqrt{\frac{1 + tan^2\frac{\pi x}{d}tanh^2\frac{\pi z}{d}}{tan^2\frac{\pi x}{d} + tanh^2\frac{\pi z}{d}}} \quad ,$$

Where C is the capacity of unit length; ε_0 is the dielectric constant. When $l > d \gg 2R$, one can approximate C by

$$C = \frac{2\pi\varepsilon_0}{\frac{\pi l}{d} - ln\frac{2\pi R}{d}} \quad .$$

Two approximations are relevant since the S2 is only produced in the areas very close to the anode wires. The electric field far from the wires is similar to the field in the parallel- plate capacitor.

For $l > z \gg d$,

$$E = \frac{CV_0}{2\varepsilon_0 d} \qquad ;$$

And For the electric field very close by the wires, it is dominated by the term

$$E(r) = \frac{CV_0}{2\pi\varepsilon_0} \cdot \frac{1}{r} \qquad .$$

In this experiment, the distances constructed are

$$\begin{aligned} R &= 12.7\ \mu m \\ l &= 3\ mm \\ d &= 3\ mm \qquad . \end{aligned}$$

Therefore, the capacity in the system is 8.22 pF. And then the electric field close to the anode wire is calculated to be

$$E(r) = \frac{783.7}{r}\ \left({}^{V}\!/_{m}\right) \qquad .$$

Given the characteristic electric field of the onset of proportional scintillation, the region of strong field can be calculated easily by the formulas above.

To observe proportional scintillation very high electric fields are necessary. We note that the field close to a wire is much stronger, decaying with r^{-1}. In our original plan, different wire diameters and high voltage settings will be evaluated. If the field strength is too large, an electron cannot only generate photons, but also other electrons (i.e. it can start an avalanche). An electron avalanche is not desirable since additional fluctuations will make the measured energy less precise.

A CSA is used to collect the electrons hitting the anode. It translates into a measurement of delay time between direct scintillation and proportional scintillation. CSA can also help us avoiding the charge avalanche.

2.3 Electric Field

In this experiment, a drift field and a MWPC-like field are needed to measure the S1 and S2 signals. Fig. 3 is the electric circuit diagram of the GIC. The cathode and the grid below the anode are used to generate the drift field by applying a negative high voltage on the cathode. In the experiment, the electrical potential on the cathode is maintained at a constant -3 kV. Two field-shaping rings are placed in equidistance. Three resistors are installed between these four plates to divide the electrical potential equally and it is 150 $M\Omega$ each to avoid a possible short circuit.

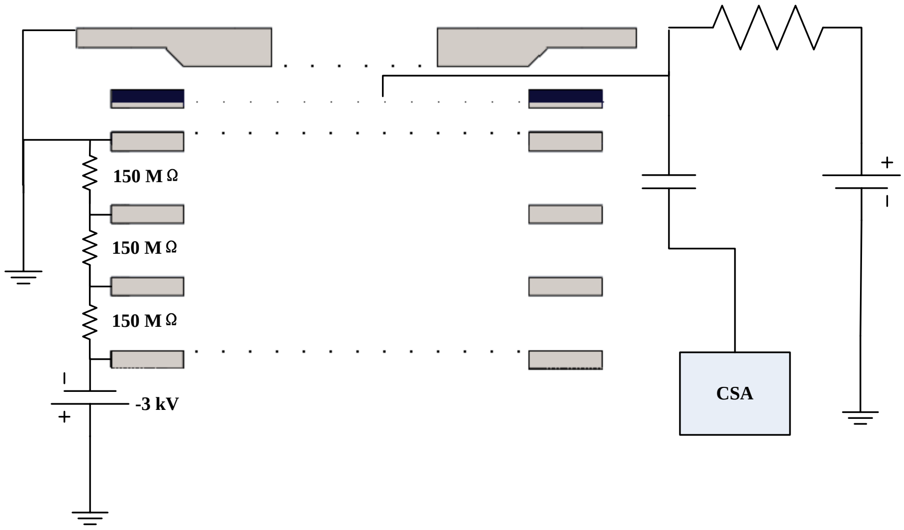


Fig. 3 Electric circuit diagram of the GIC

High optical transparency grids are used to create the drift field and the MWPC-like field while simultaneously allowing the measurement of the S1 and S2 signals with both top and bottom PMTs. The grid before the anode and the cathode are made of a stainless steel 316 square ring soldered with a cluster of parallel stretched stainless steel 316 wires of 100 μm in diameter and 3 mm spaces in between. Fig. 4 shows one of these grids.

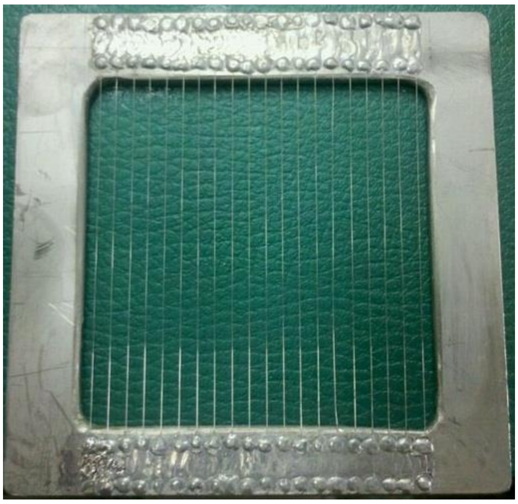


Fig. 4 Photo of the grid

The anode is made of a stainless steel 316 square ring (thinner in two opposite edges), a cluster of parallel stretched stainless steel 316 wires of 25 μm in diameter and two Cirlex boards. The latter are mounted on the thinner edges of the steel ring by Epoxy. Fig. 5 shows the anode ring before soldering the wires. The wires are too thin to be soldered directly on the stainless steel. Therefore the Cirlex boards are necessary to hold the wires. In this experiment, the horizontal spatial resolution of the signals is not evaluated. Thus all the wires are electrically conducted to each other and a CSA was connected to the wires through a 1 nF capacitor to amplify the charge collected on anode.

Two high voltage power supplies are used to apply the electrical potential on both anode and cathode. They are MODEL 3106D from CANBERRA. Each of them can alert the polarity, and the maximum output is 6 kilovolts.

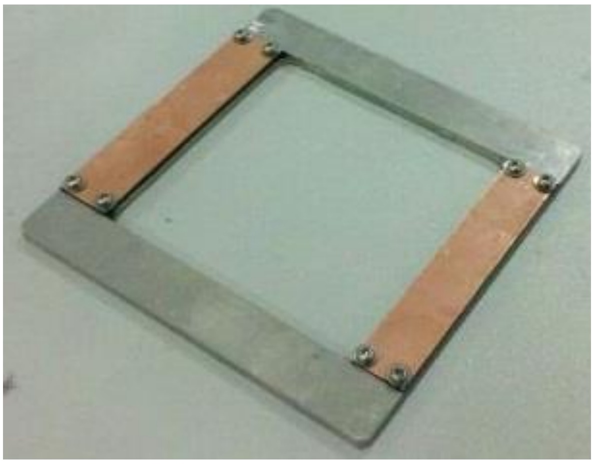


Fig. 5 Photo of the anode holder ring

The top plate was used to make the MWPC-like field, and the window with the grid in the center is made in order to make it transparent to the top PMT.

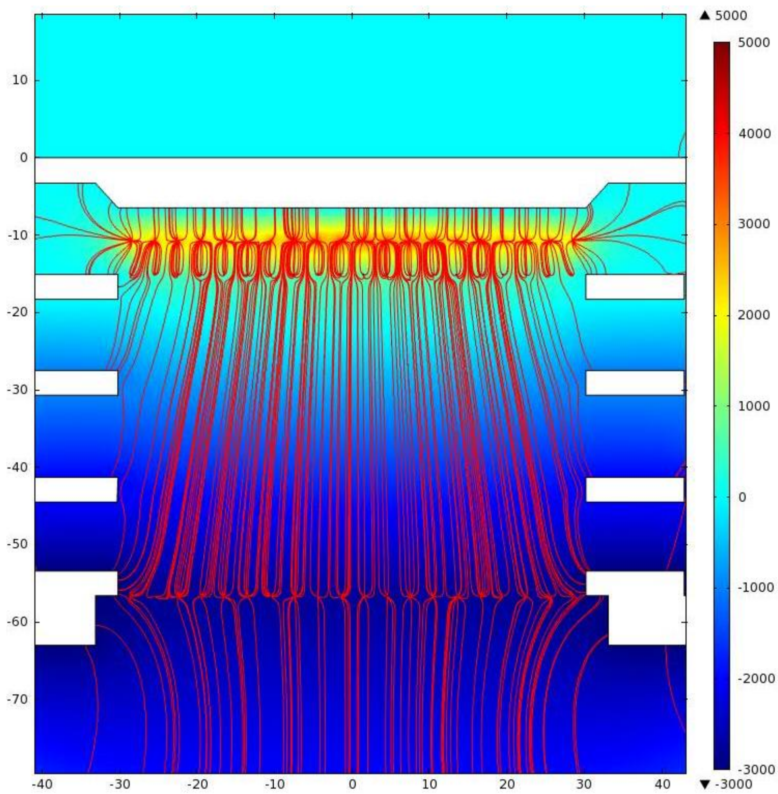


Fig. 6 Simulation of electric field in GIC

Fig. 6 shows the finite element analysis simulation of the drift field and the MWPC's field by using COMSOL Multi-physics. The color bar is the legend of potential in the space and the red lines are streamlines of the electric field. In this picture, the potentials on the anode and cathode are 5.0 kV and -3.0 kV respectively. The edge effect is prominent due to the finite calculation ability of the PC used to do this simulation. The mesh of this finite element is not fine enough, and this cannot produce the field very accurately. However, most of the center area of our sensitive area is homogeneous.

2.4 PMTs

Five PMTs used in the experiment are R8520-406 SEL Photomultiplier Tube from HAMAMATSU. They are 1 inch $\times$ 1 inch metal-channel PMTs with low intrinsic radioactivity. They have special Bialkali (antimony-rubidium-cesium, Sb-Rb-Cs; antimony-potassium-cesium, Sb-K-Cs) photocathodes for low temperature operation down to -110 ℃ and optimized for detection of the Xenon 178 nm scintillation light. The average QE (Quantum Efficiency) of these PMTs is estimated around 30% which is provided by the company and not calibrated in this experiment.

The bases of them were made with a design from Columbia University. It is a negative base with the maximum -900 V applied to the PMT. The power supply and signal cables are separated and four Bundy pins are used for one PMT through the chamber feedthrough.

A four channel high voltage programmable power supply, Mod. N1470 from CAEN is used to supply negative high voltage to the four bottom PMTs. Another two-channel high voltage power supply from ORTEC is used for applying negative HV to the top PMT.

The calibration of the gain of four bottom PMTs is made with the spectrum of SPE (Single PhotoElectron).

2.5 Cryogenic System

In this experiment, we use liquid xenon as detection medium. Xenon is a noble element and is gaseous at room temperature. Since the PMTs can only stand below 2 $BarG$, the suitable operation point of xenon is -95 ℃. Fig. 7 shows the phase diagram of xenon. The saturation vapor pressure is 0.8 $BarG$

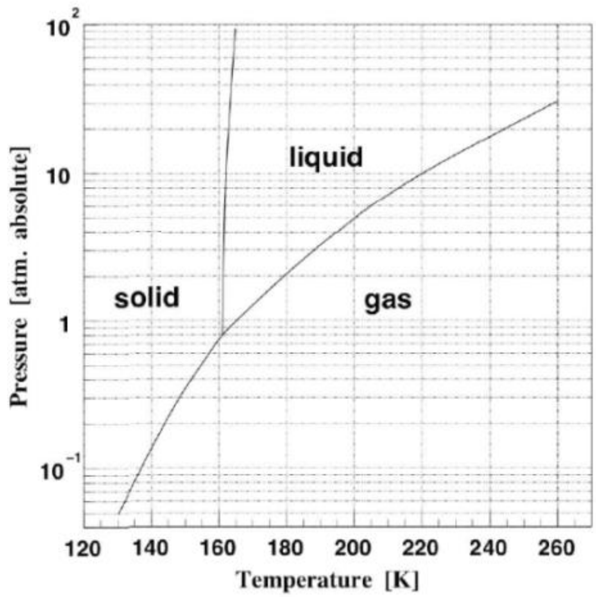


Fig. 7 Phase diagram of Xenon [7]

This means we need a cryogenic system to generate the right environment for our measurement. Fig. 8 is the schematic of the cryogenic system in this experiment.

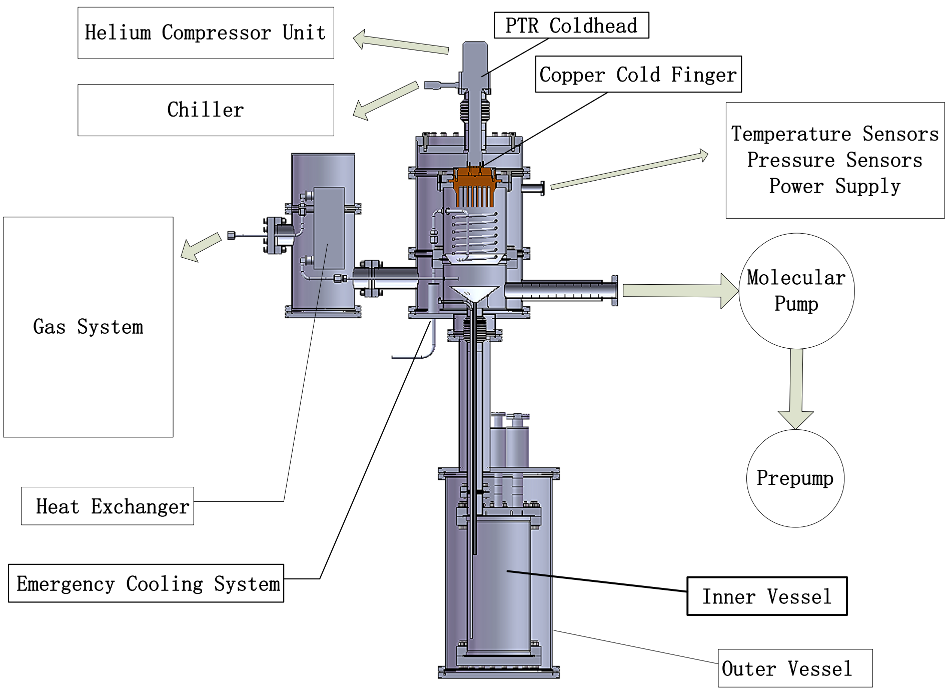


Fig. 8 Cryogenic system

To create a cryogenic environment, our system divides the space into three parts. The inner part is the space inside the inner vessel connected to the gas system; the outer part is the space between the inner vessel and the outer vessel; the third part is the space out of the system. The detector is located in the lower part of inner vessel. To save the cooling power, an outer vessel was used to generate an outer vacuum to obstruct the heat going into the inner vessel.

The molecular pump and its pre-pump are from Varian and connected with the inner vessel and outer vessel. During the run, only the outer space is pumped and produces a vacuum which the pressure is around $1.2 \times 10^{-3}\ Pa$.

The cooling power for the experiment is provided by a PTR (Pulse Tube Refrigerator) coldhead on the top of the system. The PTR is driven by a CAN-31D 3 kW air-cooled Helium Compressor Unit from Sumitomo Heavy Industries Ltd. The PTR takes out the heat from gaseous xenon inside through a piece of copper cold finger. Then, a chiller provides a water-cooling to the hot end of the PTR. There is a heater around the coldhead to balance the redundant cooling power in some process. Various sensors are installed to monitor the parameter of the system through the chamber feedthrough.

During every run, the xenon must be purified constantly, since any remaining impurity concentrations determine the attenuation for scintillation light and drifting charges. A high temperature getter is used to remove the electro-negative substances out-gassing from all surfaces of the entire system. The liquid needs to be evaporated before entering the hot getter and needs to recondense afterward. A large amount of thermal energy is stored in these phase transitions. A parallel plate heat exchanger is used to extract the heat from liquefying the gas and evaporate the liquid going into the gas system. A lot of power is saved in this process, and all the liquid xenon is purified as expected in a cycle. A double diaphragm pump keeps the xenon streaming through the getter.

An emergency cooling system has been designed to save the system when any unexpected situation happens (e.g. a power failure, loss of outer vacuum, loss of cooling power or any error on pump or chiller, etc.). In order to protect the entire detector inside the inner vessel, especially PMTs as mentioned before, a rupture disc and its holder are mounted to the inner volume. The rupture disc is designed by V-TEX Corporation to break when inner pressure arrives to 2.5 $BarG$. Once the rupture disc is broken, all xenon inside will be released The emergency system is consisted of a stainless steel tube wound into a coil, a solenoid valve with its power supply, a tank of liquid nitrogen, and a switch. In this experiment, Omega Process Panel Meter DP25B-E-230-AR was used to monitor the inner pressure, and it is a versatile meter. It has a switch and was set to be open when the inner pressure exceeds 1.3 $BarG$ and closes when the pressure is below 0.5 $BarG$ after opened. Once the switch is open, the solenoid valve will open and the LN_2 will cool the gaseous xenon around the coil in the inner vessel. Fig. 9 shows the circuit of the emergency system.

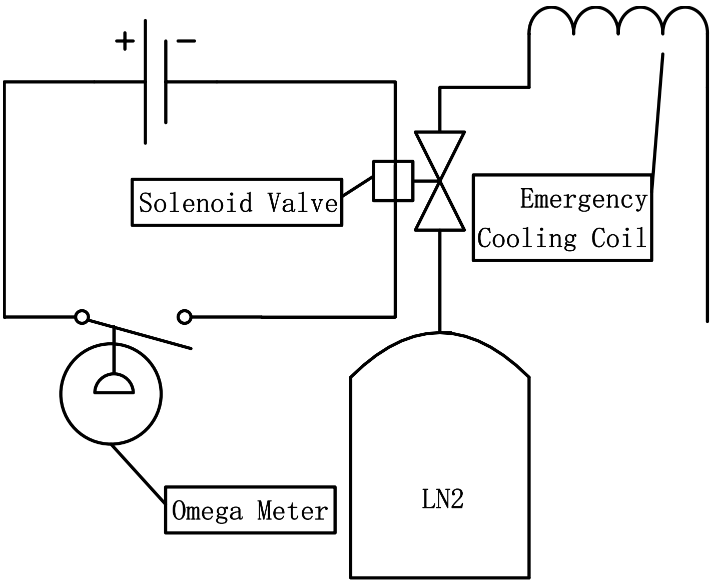


Fig. 9 Circuit of the emergency system

2.6 Slow Control System (SCS)

A slow control system is installed to monitor the crucial parameters of our system and send notification through emails and messages when any of these numbers exceeds the high or low limitation. The most crucial number is the inner pressure measured by Omega meter; it plays a key role in emergency system and is relative to the rupture disc. When the inner pressure exceeds 1.0 $BarG$ or it is lower than 0.5 $BarG$, the SCS will send warning emails and messages. The SCS in this experiment is a sub-system of the PANDAX slow control system. Besides the inner pressure and SCS also monitors the working state of pump, LakeShore temperature monitor unit with its controller and the air-pressure for pneumatic valves in gas systems.

2.7 Generator and Uninterruptible Power Supply (UPS)

There is a prerequisite to avoid malfunction of both LN_2 emergency system and SCS, the electric power supply for equipment, Omega meter, solenoid valve and computer. A diesel generator is spared for emergency power supply. When the commercial electricity is cut off, the entire system will automatically switch to the circuit with the generator. Furthermore, a UPS from SANTAK is in charge of electricity supply in the case of an unexpected power failure. It can offer electricity for these low power instruments overnight or before we fixed the shortage of power.

2.8 Gas System

The Xe used to fill the chamber is stored in two high-pressure aluminum cylinders. Each cylinder has an internal volume of 10 L and each bottle typically contains about 6 kg at a pressure of 54 atm. A regulator was used to abate the pressure. Two gravimeters, or stress sensors are used to show the xenon left in the bottles. Fig. 10 shows the schematic structure of the gas system.

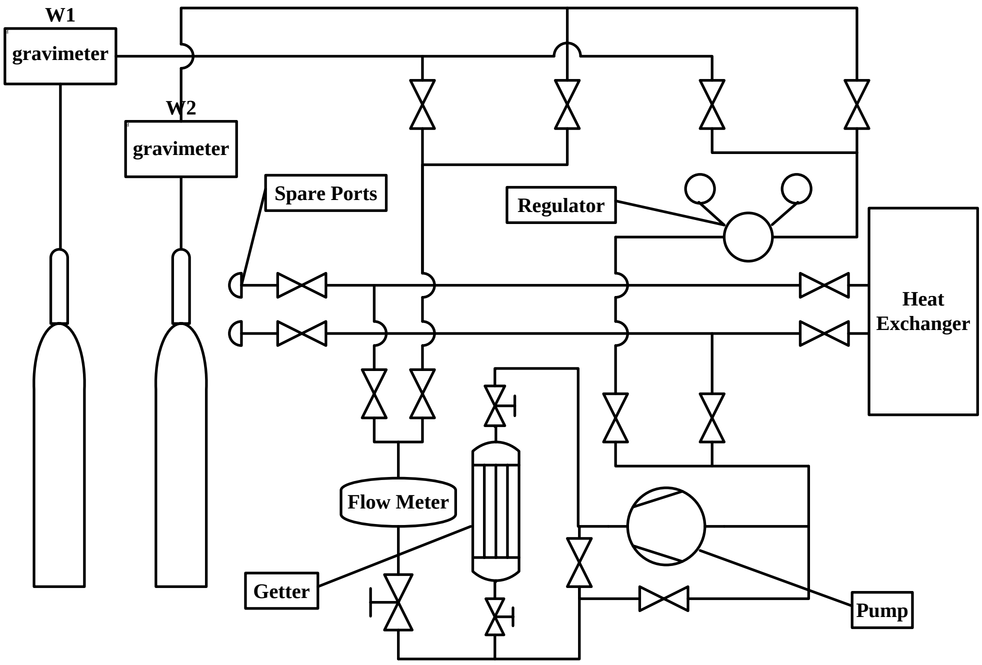


Fig. 10 Schematic structure of gas system

A pump, a high-temperature getter and a flow meter make up the xenon purification system. As mentioned before, the purity of xenon is the most important factor in the experiment. The maximum flow rate of the getter is 5 liters per minute and that means 10 kg of xenon will take eight hours to be purified once. Therefore, three cycles per day will improve the purity.

There are two spare ports left in the gas system to fill in or take out the xenon from the system. It can also be used for other possible applications in the future. For example, if we can use the cryo-well technology to measure the purity of our xenon instantly.

Except for the manual valve below the flow meter and the two manual valves on either hand of the getter, all valves used are electro-pneumatic valves which operating air pressure is 0.7 MPa. Therefore, an air-compressor is used to offer the high pressure air when the valve should be open.

2.9 Filling, Running and Recuperation

In this experiment, xenon is originally stored in the cylinders in a gas phase and filled into the inner vessel by expansion under the pressure gradient. Fig. 11 shows the access of filling procedure. The inner volume is pumped at first to the order of 10^{-4} Pa and baked at 50 ℃ (the highest temperature the photocathode stands) for 50 hours. Then, disconnect the pump to the inner vessel and start filling. The first step so-called pre-cooling is important for a small-volume detector. Gaseous xenon of 2.5 $BarG$ through the regulator goes through the getter and a flow meter with a flow rate controller is used to protect the getter since the maximum flow rate is 5 liters per minutes. The purified gaseous xenon is pushed into the inner vessel through the copper cold finger. The PTR cold head is set to -95 ℃, and the gaseous xenon condenses to a liquid, which becomes the liquid droplets. A funnel is designed to guide the liquid into the inner vessel. When the droplets fall down and touch the warm wall of the inner vessel, they will evaporate and generate a large local pressure which may damage the PMTs. Therefore, a buffer is connected to the inner vessel and the pre-cooling process is indispensable. The pre-cooling procedure is:

1. Fill gaseous xenon into the inner vessel at room temperature before cooling. And stop filling when the pressure arrives 2 atm.

2. Start cooling until the pressure decreases to the saturated vapor pressure around 0.8 $BarG$.

3. Fill the gaseous xenon again and get the equilibrium between the cooling power and the warm gas.

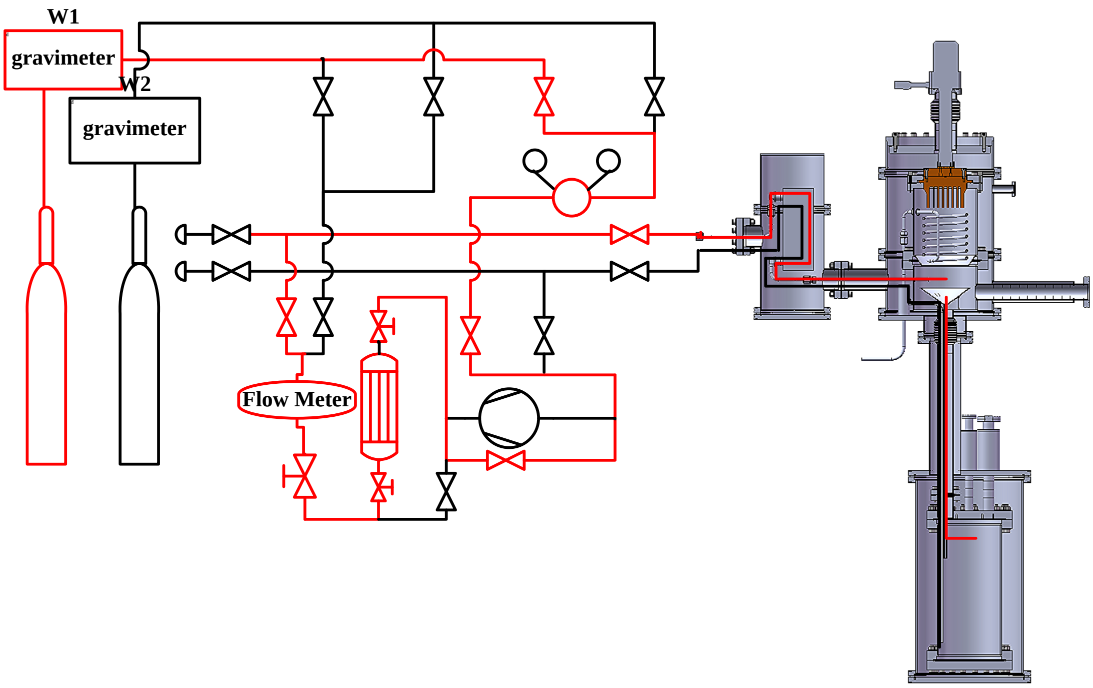


Fig. 11 Route of filling process

During the run, xenon must be purified continuously as mentioned before. Fig. 12 shows the access of purification. In the inner vessel, there are two pipes for the cryogenic system. The longer one is used to take the liquid out to the heat exchanger for purification; then, the liquefied pure xenon goes back into the upper portion of the chamber through the funnel. To make sure of high efficiency for this purification, the two pipes are separated as far as possible.

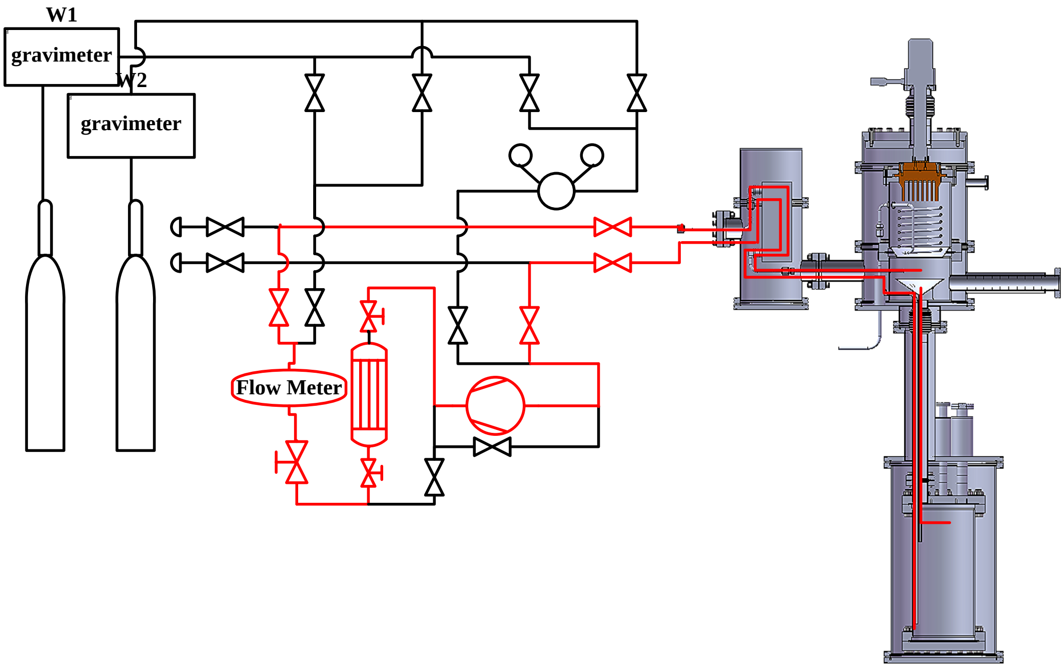


Fig. 12 Route of recycling process

After the run, the gas system is used to empty the chamber. A liquid nitrogen Dewar tank is used to cool down one storage cylinder. Using the fact that the temperature of liquid nitrogen is below the freezing point of xenon, the cylinder becomes a cryo-well to pump back the xenon back into the detector. Due to the efficiency issue, the recuperation does not use the getter, since its limitation on the flow rate. Fig. 13 shows the access of the recuperation.

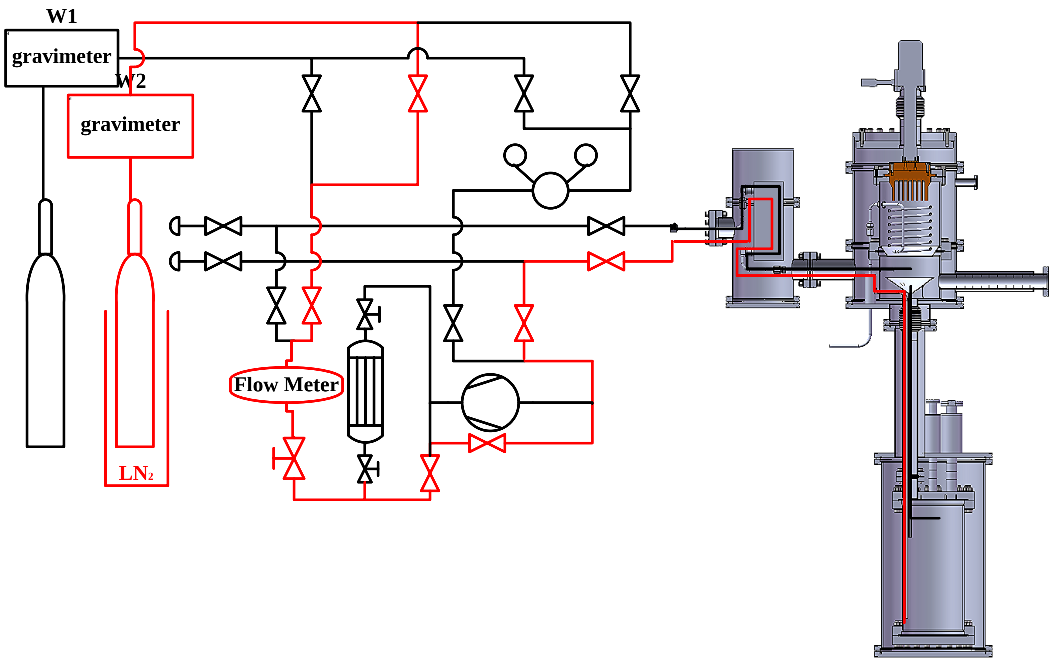


Fig. 13 Route of recuperation process

3. Data Acquisition

3.1 DAQ System

The DAQ (Data AcQuisiton) system is in proximity to the electronic system (a NIM cabinet) and based on an oscilloscope. The oscilloscope is a MSO9404A four channel mixed signal oscilloscope with 4 GHz and 20 GSa/s from infiniium series scopes of Agilent Technologies. Electronics units, such as pulse shaping unit, dual sum and invert unit, and some amplifier are not used in this experiment. So does the flash ADC (Analog Digital Convertor) module. The raw data from the PMTs and CSA are saved by the oscilloscope and store the waveforms to a 1 TB portable hard drive from WD Elements.

3.2 Trigger

Fig. 14 shows the screen on the Agilent oscilloscope. An advanced trigger function is used to improve the efficiency of required data acquisition.

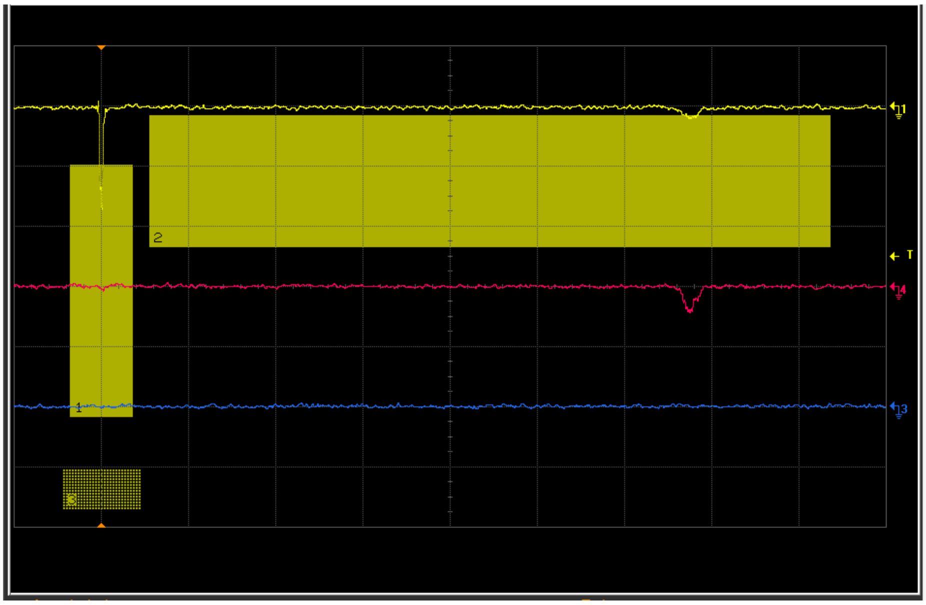


Fig. 14 Photo of the screen on the oscilloscope

The yellow signal, Ch1 (Channel 1), is the sum scintillation signal of four bottom PMTs after SPE calibration; the red signal, Ch2 (Channel 2), is the scintillation signal of the top PMT. The blue signal, Ch3 (Channel 3), is the charge signal from anode after the pulse shaping of CSA. The scale in the data acquisition is set to be 2.00 $\mu s/div$ in horizontal time axis and different scales in perpendicular voltage axis in three channels. 200, 100 and 10 mV/div is set in Ch1, Ch2 and Ch3 respectively. In the Fig. 14, the first pulse is a direct scintillation and the second pulse is the proportional scintillation. The drift time is about 13 μs and evaluated by the time interval between the rising edges of S1and S2. The structure of S1 and S2 will be explained below.

Some parameters are fixed to be the same for all three channels. Tab. 1 shows the actual values of these parameters.

Tab.1 Parameters setting

Bandwidth Limit	45.8 kHz
Analog Memory Depth	1.024 kpts
Analog Sampling Rate	50.0 MSa/s
Digital Memory Depth	40.921 kpts
Digital Sampling Rate	2.0 GSa/s

There are two types of triggers that can be used to capture the event. One is normal HW Trig (HardWare Trigger) with various modes like usually used edge mode or pulse width mode. On the other hand, a trick used here is the InfiniiScan type of trigger with Zone Qualify mode. Three rectangular regions on the display are created to qualify the signal. In this experiment, only Ch1 is used as trigger source for these zones. Therefore the zones are all in yellow. The normal zones, Z1 (Zone 1) and Z2, are used to qualify the signals of the source channel that the signal must intersect with the zone. Oppositely the shadow zone, Z3, is used to select the source signal which must not intersect with the zone. During the acquisition of data in this experiment, Z1 is aimed to set the lower limitation for the height of pulse of Ch1. Z2 is placed for the sake of requirement that there must have another pulse after the first pulse at least. Z3 is used with the aim of setting the higher limitation of the height of the first pulse of Ch1.

With the setting in Tab.1, the waveforms of Ch1 saved as txt files to the portable hard drive have at least two pulses within 16.0 μs. Most of the first pulses have the height from the Cs gamma source. The details will be explained in the data analysis section below.

3.3 Structure of signal

Fig. 15 shows the formation process of S1 and S2. The gamma rays from Cs source will interact with the xenon and produce fast electrons. These electrons excite and ionize xenon atoms on their path and form a package of free electrons and positive Xe^{+} ions. Both recombination of ionized electrons with ions and the return to ground state yield VUV light in wavelength 178 nm. Because of the formation of this scintillation, the direct scintillation has a short rising time and a relatively long drop time. The Full width at Half Maximum (FWHM) of direct scintillation is about 100 ns.

Some of these free electrons drift upward to the anode because of the applied electric field. If xenon is pure enough that the mean free path of these electrons is larger than the scale of the detector, which means the electron will not be absorbed by electronegative impurities like oxygen; these electrons will crash toward the anode wires because of the extremely strong field around these thin wires. The acceleration of these free electrons will lead to a secondary excitation around the anode and produced proportional scintillation. The width of the proportional scintillation is determined by the duration of electrons collection on the anode, because the original free electrons may be produced at different depth. Usually, the FWHM of proportional scintillation is around hundreds of nanoseconds, and it is more symmetrical in the sense of the rising time and drop times are proximal.

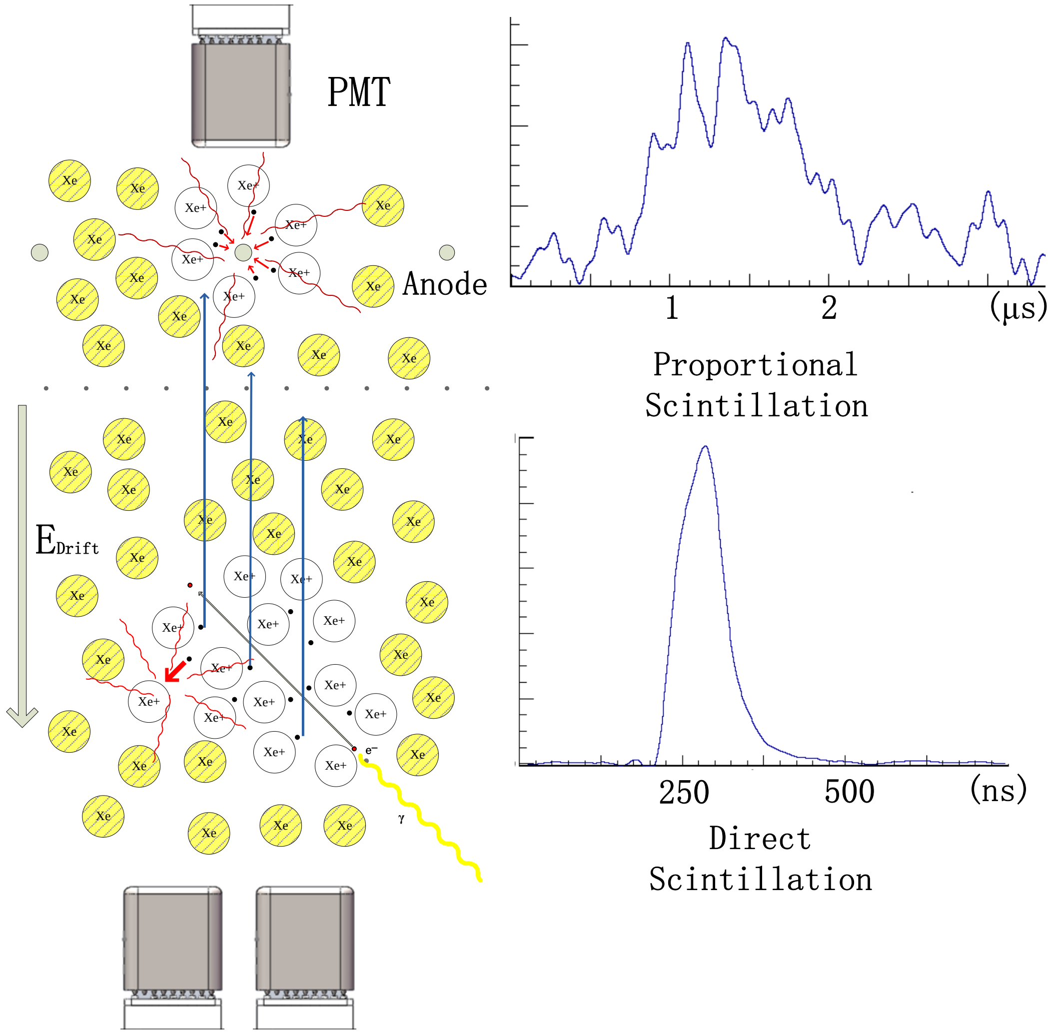


Fig. 15 Formation of S1 and S2 signal

3.4 Physical mechanism of other types of events

Some typical event patterns are shown below. Alpha particles and electrons from the environment have almost no possibility to go into the sensitive area due to the fact of their weak penetration ability to vessels.

Muons from cosmic rays are charged particles with vast amount of energy compared to gamma rays. Therefore, the PMTs' signal is saturated when a muon is observed. Fig. 16 is an event of muon trace detected on the screen of the scope. No electric field was applied in the event. The top PMT and four bottom PMTs observed the flaming light. The muon will interact with xenon through electromagnetic interaction and produce free electrons on its track. The CSA integrates the charge signal from anode. Charge that is collected at the same time as the scintillation signal and 13 μs duration of the oblique line show that two ends of the muon track may locate at the top and side surface of the sensitive area. If they are in the side and bottom surface, the distance between the side end point and the top surface can be displayed in the form of delay between the light and charge signals.

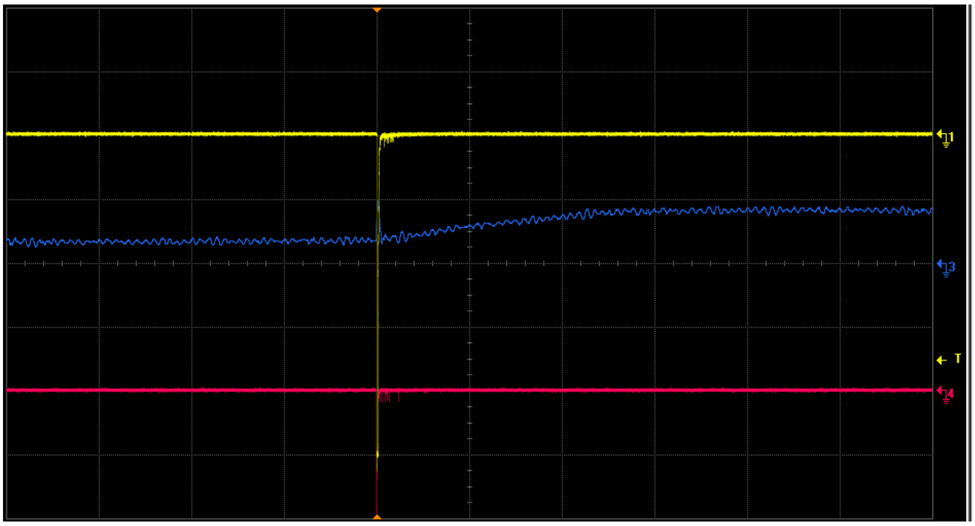


Fig. 16 Muon event with weak electric field

The scales in this photo are 500 mV/div in Ch1 and Ch2, 50 mV/div in Ch3 and 5.00 $\mu s/div$ in horizontal.

Fig. 17 shows a muon event from the saturation of PMTs. The electric field was applied in this event and the electrons that reached the anode also produce the proportional scintillation.

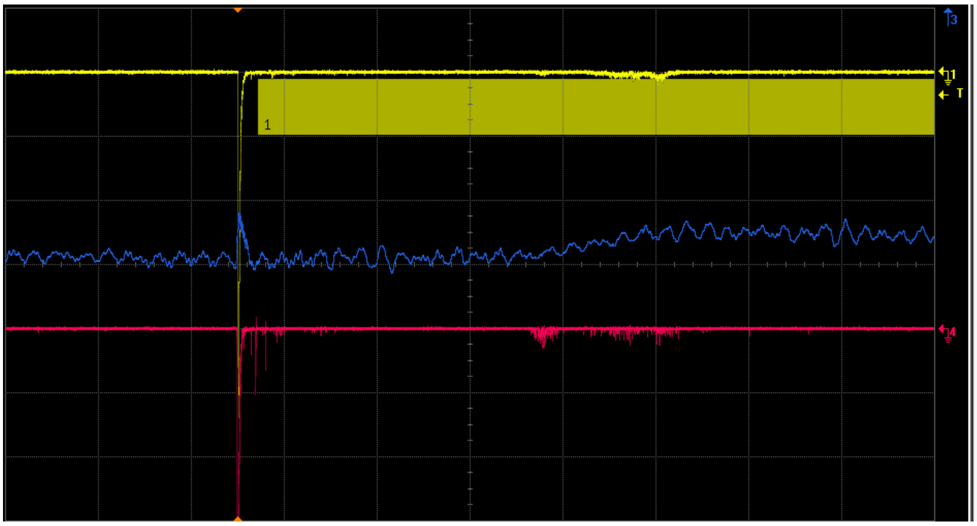


Fig. 17 Muon event with electric field

The scales in this photo are 500 mV/div in Ch1, 100 mV/div in Ch2, 20 mV/div in Ch3 and 2.00 $\mu s/div$ in horizontal.

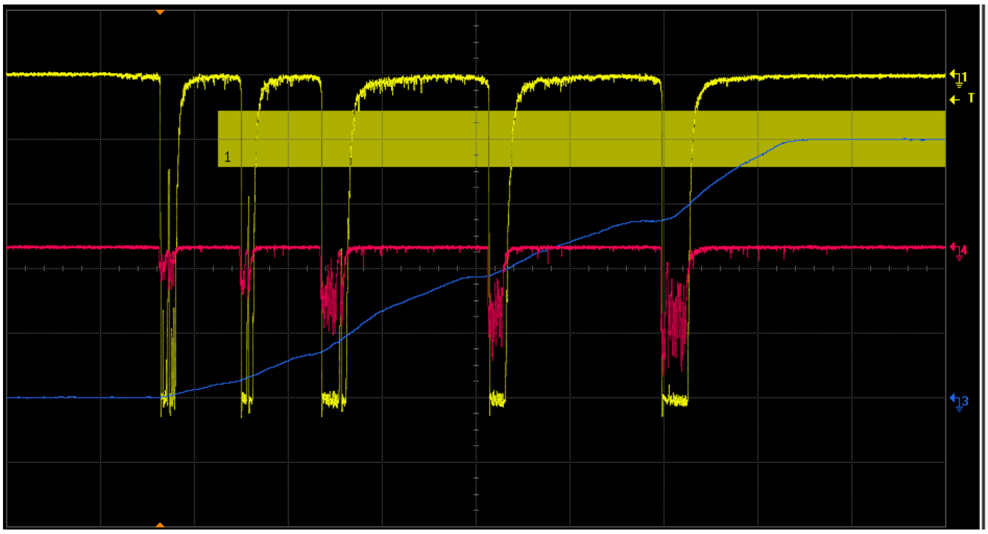


Fig. 18 An event with electric field

The scales in this photo are 500 mV/div in Ch1, 500 mV/div in Ch2, 1.00 V/div in Ch3 and 1.00 $\mu s/div$ in horizontal.

Fig. 18 shows one of events that we do not understand the physical mechanism behind yet. The top PMTs are saturated but have several pulses in a row, which are wider than the muon case each. At the same time, the top PMT observed the light with the same positions and width but increasing height. The CSA has several large amplitude charge signals step by step right after every light pulse. One possible hypothesis is that tiny sparks happened inside between the anode and the grids before it.

4. Data Analysis and Results

4.1 Calibration of signal

We planned that the gains of PMTs should be calibrated by a ^{137}Cs source outside the chamber. However, after many optimizations the spectrum is still structureless and not compatible with the spectrum of other experiments using the same source. Fig. 19 shows the spectrum of the ^{137}Cs source.

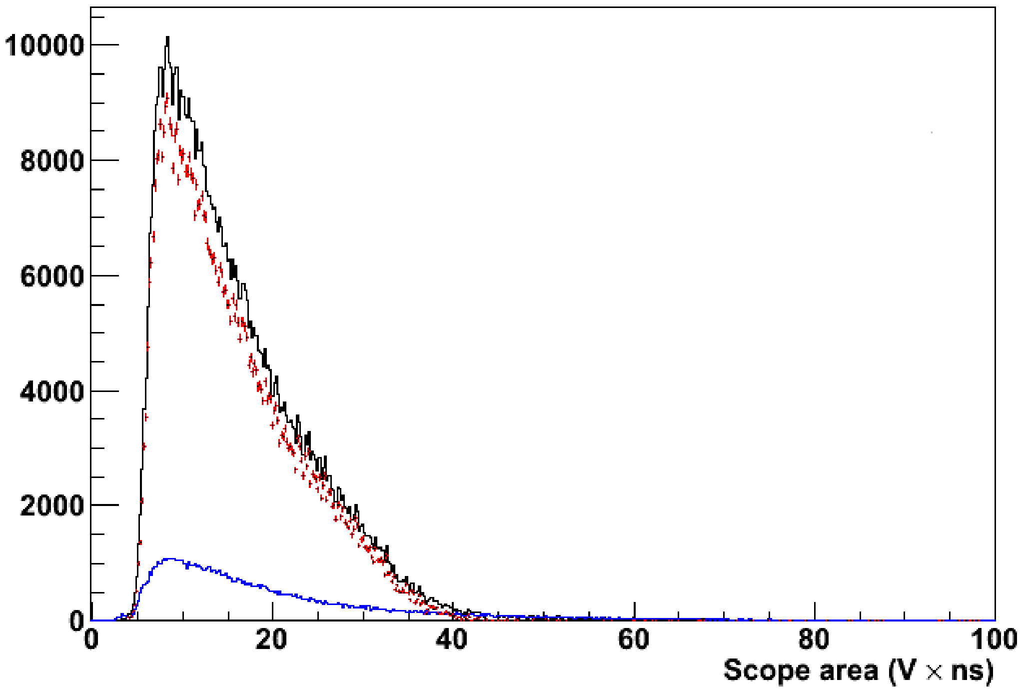


Fig. 19 Spectrum of 137Cs

The black line in the 1-D histogram above is the energy spectrum of Cs and the background together. The blue line is background only and the red one is the Cs spectrum derived from the black one subtracted by the background. The signal input to the scope through 50 Ω impedance LEMO cables and BNC cables which means the scope area divided by 50 is the energy. From the observation of the figure, Cs events have maximum energy around 40 $V \times ns$.

A ^{60}Co source is also used to calibrate the PMTs. Fig. 20 shows the spectrum of ^{60}Co. The representation of three lines in different colors is the same with the Cs case. A sub-graph is drawn to show the cutoff of the Co energy around 80 $V \times ns$.

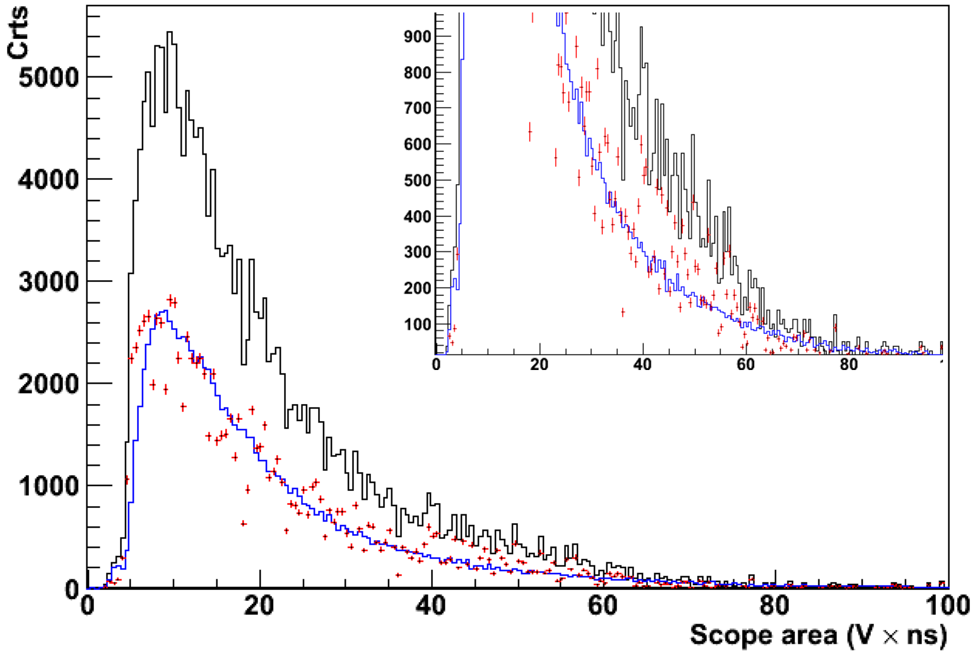


Fig. 20 Spectrum of ^{60}Co

From Figs. 19 and 20, the same phenomenon is true about the cutoff of the spectrum. The highest energy from Co source is almost twice of the one in Cs case.

One conceivable interpretation is that the detector has some light leakage with the sensitive area and that makes the light collection efficiency of PMTs strongly dependent on the position of the event. This geometric effect will change the solid angle strongly and make the spectrum structureless. That means the spectrum of every fixed point of event will have a normal Compton plateau ended with a Compton edge and a high resolution energy peak placed on 662 keV like the one in Fig. 21.

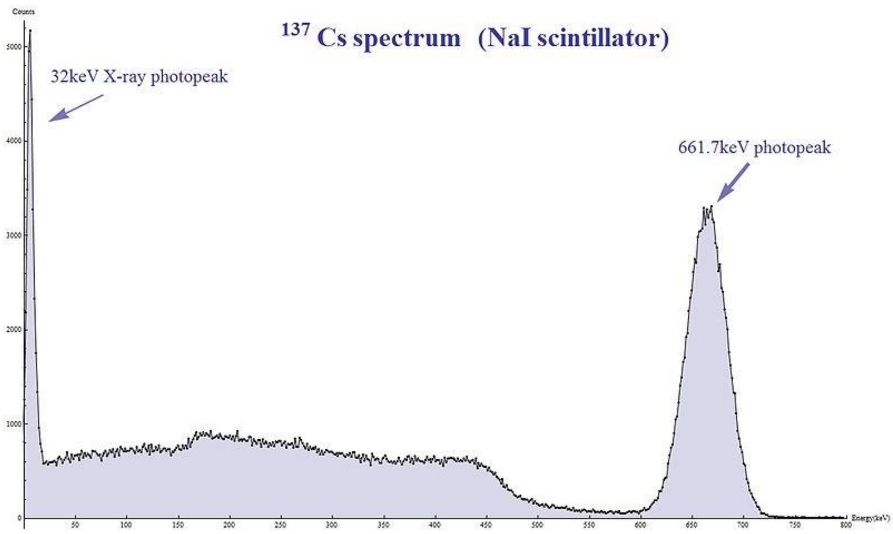


Fig. 21 Typical spectrum of ^{137}Cs

However, due to the great spatial dependence, the spectrum was shifted compared with different positions and overlapped together. The narrow gamma peaks will stand side by side but the stumpy Compton plateau will accumulate its altitude. That may lead the spectrum measured in this experiment.

Calibration with single photon electron is used eventually in this experiment. Photon cathode has probability to occur a single-photon emission spontaneously even without source. Single photoelectron and double photoelectrons will produce structure in the background spectrum when triggered at low level. Fig. 22 shows the background spectrum of photoelectrons.

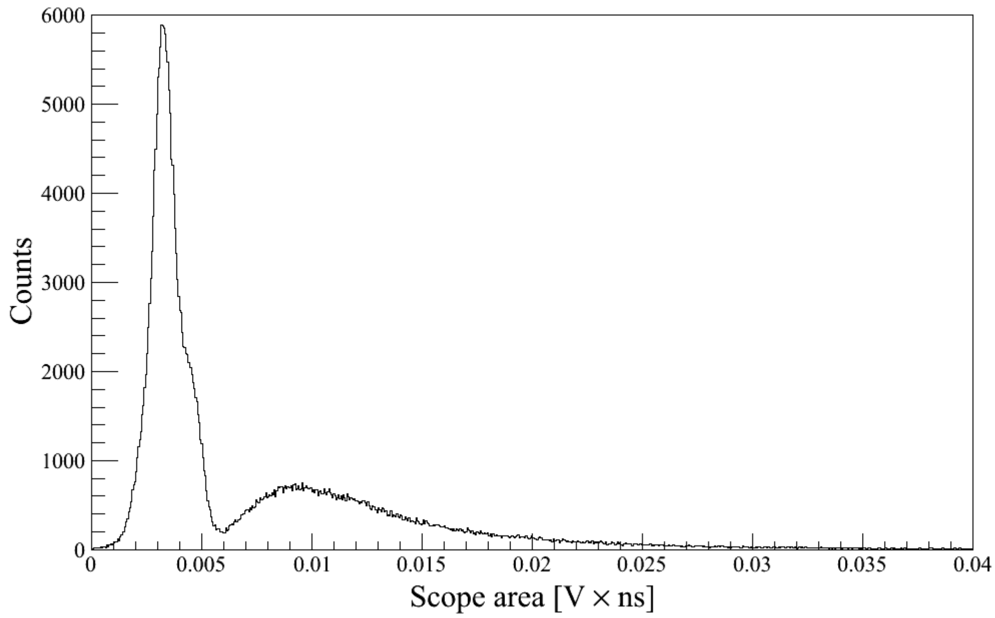


Fig. 22 Spectrum of SPE

The energy peak of single photoelectrons is located at 0.0096 $V \times ns$. The energy peak of background is located at 0.0032 $V \times ns$. Therefore, the modified area of SPE is 0.0064 $V \times ns$. The charge corresponding to this area is $1.28 \times 10^{-4}\ nC$ derived from being divided by 50 Ω. Therefore, the total gain of four bottom PMTs is 8.00×10^5, and the average gain of each PMT is then 2.00×10^5.

21 eV are needed to produce a photon in liquid xenon. Therefore, the photon yield of a gamma ray with 662 keV is 31523.8 from ^{137}Cs. The cutoff in Cs spectrum can be used to estimate the light collection efficiency. 40 $V \times ns$ divided by the energy of SPE gives the number of photoelectrons produced by photo cathode. Quantum efficiency of PMTs is 30%. Therefore 13889 photons are observed by the PMT's, which mean the light collection efficiency of geometry factor is around 44.058%.

4.2 Data Analysis

Data stored in the portable hard drive is transferred to a computer and analyzed with ROOT and a program specifically written for data analysis of this experiment. The raw data for an event consists of 3 waveforms of 16000 samples each. The first waveform is from the four bottom PMTs. The scintillation signals of these four PMTs are calibrated by the amplitude of single photon electron waveforms and are added together through a linear fan-in/fan-out unit of model 428F from LeCory. The second waveform is the scintillation signal from the top PMT and the left one is the charge signal from CSA.

Generally speaking, the data of this experiment is processed in four stages.

I) Preprocessing the waveforms.

II) Searching for peak candidates.

III) Sorting the S1 and S2 peaks.

IV) Computing the selected events into histogram for analysis in ROOT.

The program is called *DAPSLXe* (Data analysis for Proportional Scintillation in Liquid Xenon) and exhibited in Appendix. The concise algorithm and ideas of computing are narrated straightforwardly here.

4.2.1 Create trees in ROOT

ROOT is an object-oriented data analysis framework developed by CERN (European Organization for Nuclear Research). It was originally designed for particle physics data analysis and contains several features specific to this field. A key feature of ROOT is a data container called tree, with its substructures branches and leaves. A tree can be seen as a sliding window to the raw data, as stored in a file. Data from the next entry is the file can be retrieved by advancing the index in the tree. With this framework, the enormous information of large quantity events from the raw data can be treated with extremely high efficiency. Any special variable of concerned events can be used to select data from "ROOT" through the tree and its branch to the leaves (i.e. events).

Two trees are built in ROOT for this experiment. T1 (the first tree) is used to the first-level analysis and T2 is defined for a further, second-level, analysis. Step I) and II) mainly deal with the T1 and T2, which is in charge of the issue after discrimination of S1 and S2 peaks. Branches on T1 and T2 are exhibited in Tab. 2. Ch3, the charge signal is not processing in the data analysis. There are two sets of branches for Ch1 and Ch2 on T1 and Ch1 set is listed. Similarly four sets of branches for S1-Ch1, S2-Ch1, S1-Ch2 and S2-Ch2 on T2 and S1-Ch1 set are listed.

Tab.2

Tree	Branch		
T1	*EventID*	*BaseLineCh#*	*AreaTotCh#*
	NbPeaksCh#	*PeakTotCh#*	*PeakTimeCh#*
	PeakLeftMostCh#	*PeakRightMostCh#*	*PeakHeightCh#*
	PeakHighWidthCh#	*PeakHalfWidthCh#*	*PeakLowWidthCh#*

	PeakLeftLowCh#	*PeakLeftHalfCh#*	*PeakLeftHighCh#*
	PeakRightLowCh#	*PeakRightHalfCh#*	*PeakRightHighCh#*
	PeakRiseTimeCh#	*PeakDropTimeCh#*	
T2	*NbS#PeaksCh#*	*S#TotCh#*	*S#TimeCh#*
	S#LeftMostCh#	*S#RightMostCh#*	*S#HeightCh#*
	S#HeightWidthCh#	*S#HalfWidthCh#*	*S#LowWidthCh#*
	S#LeftLowCh#	*S#LeftHalfCh#*	*S#LeftHighCh#*
	S#RightLowCh#	*S#RightHalfCh#*	*S#RightHighCh#*
	S#RiseTimeCh#	*S#DropTimeCh#*	

4.2.2 Preprocess the waveforms.

After loading txt files to ROOT, several variables are defined for the first step of the preprocessing, the modification of baselines of the waveforms since the 50 Hz noise from commercial electricity is indispensable. The explanation of the baseline shift effect is shown in Fig. 23.

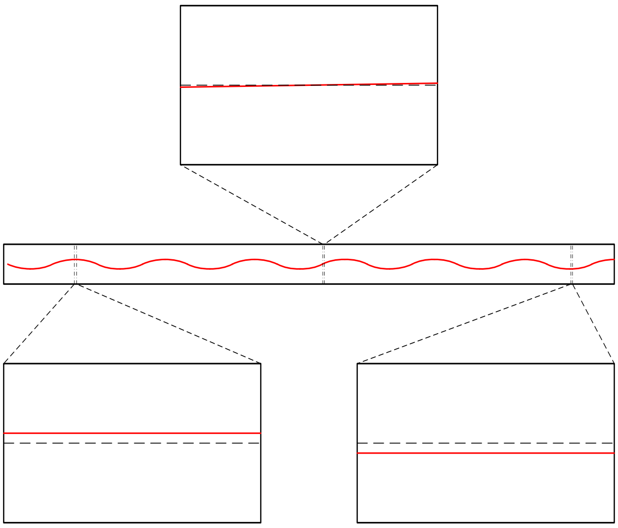


Fig. 23 Baseline shift effect

A threshold (*pre_signal_threshold*) is defined to set the range to calculate for baseline rectification. This number is chosen to be 0.1 V from the observation on the oscilloscope. Then the voltage will be integrated until it exceeds the *pre_signal_threshold*. Then the baseline shift is defined as the average voltage of these bins. And the standard deviation *sigma_baseline* was used to set a signal threshold by multiply 6 for peak search. The original waveform subtracts the baseline shift afterward. A *max_pulse_width* is 200 bins (250 ns) also defined to be to search the peak.

4.2.3 Search for peak candidates

After we get the signal threshold, the criterion of peak candidates can be defined. First of all, the bin of maximum voltage (larger than signal threshold) is picked out and assigned to variable *PeakTimeCh#*. And then an interval belongs to this peak by finding out the 90% height bin 50% height bin, 10% height bin and end bin. These bins are defined to be the first bin which the average of its 10 neighbor bins is enter the section of 92% ~ 88%, 52% ~ 48% and 12% ~ 8% respectively during the bidirectional search from *PeakTimeCh#*. They are saved in variables such as *PeakRightHighCh#*, *PeakRightHalfCh#* and *PeakRightLowCh#*. The end bins assigned to *PeakLeftMostCh#* and *PeakRightMostCh#* are defined as the end of *max_pulse_width* unless there is a bin which the average of its 10 neighbor bins is enter the interval of plus-minus one-tenth of the signal threshold. After one peak is selected, the voltage of the interval between *PeakLeftMostCh#* and *PeakRightMostCh#* will be effaced and repeat the peak search process to find a second peak until the maximum voltage of the rest bins are less than the signal threshold. Peak rising time, peak decay time and different widths of a peak are defined relatively in the branches of T1.

4.2.4 Sorting the S1 and S2 peaks

As mentioned in last section, the structures of S1 peaks and S2 peaks are dissimilar to each other. The half rising time (HRT) of a peak is defined by the interval between the bins of PeakLeftHighCh# - PeakLeftHalfCh#. We divide the range of the half rising time into three pieces, i.e. peaks HRT shorter than 50 bins (72.5 ns) are treated as S1 peaks and those longer but shorter than 400 bins (500 ns) are classified to be S2 peaks.

After the discrimination, S1 and S2 peaks are permutated by the size of their areas separately in each event. This operation makes the cut of events easier in the future.

4.2.5 Computing and Result

In order to pick the concerned events, several cuts are applied. To eliminate the ambiguous events, one has to make sure that there is only one S1 and one S2 with in one waveform. The singleS1 and singleS2 are aliases of the cuts which limit both second biggest peaks of S1 and S2 under 5 and 3 $V \times ns$. This saves the events as much as possible since some noises may be recognized as a peak. The rising time of the peaks are restricted within 300 ns, and the full width of them from *PeakRightMostCh# to PeakLefttMostCh#* are confined between 100 ns and 1 μs. According to the Cs spectrum, a cut is employed to the area of S1 peaks within 10 to 70 $V \times ns$. To avoid the malfunction of *DAPSLXe* program, a drift time cut is used to ensure the disjunction of S1 peak and S2 peak. An event is ignored with a drift time less than 3 μs or more than 17 μs (i.e., the high limit due to the geometry and the constant drift field).

Events in restraint of these criterions will be plotted in two 2-dimensional histograms. Color bars in three histograms represent the counts with the same X-Y value. Fig. 24 is the first histogram and shows the correlation between the areas of S1 and S2. The energy resolution of S2 with the fixed S1 area is observed.

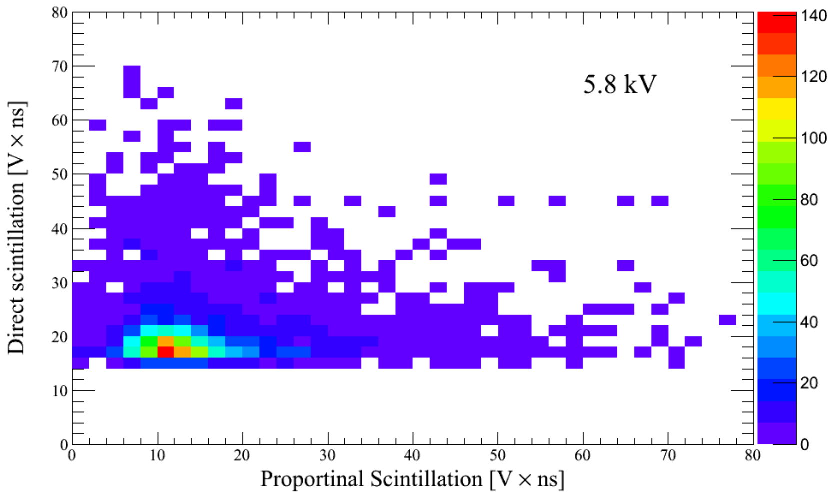
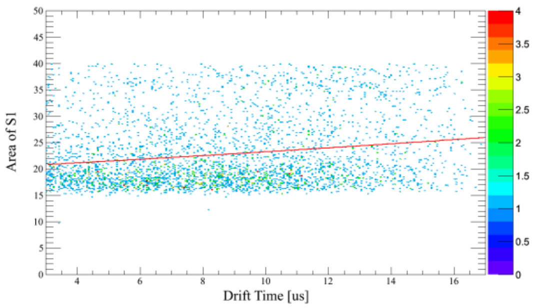
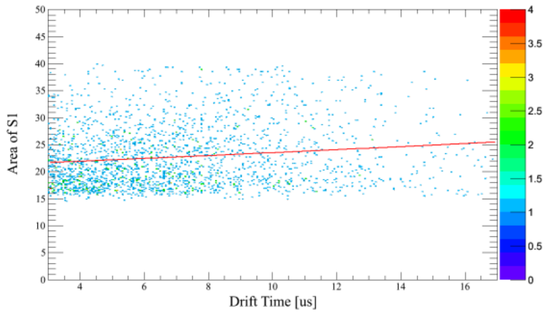


Fig. 24 Correlation between areas of S1 and S2 in case of 5.8 kV on anode

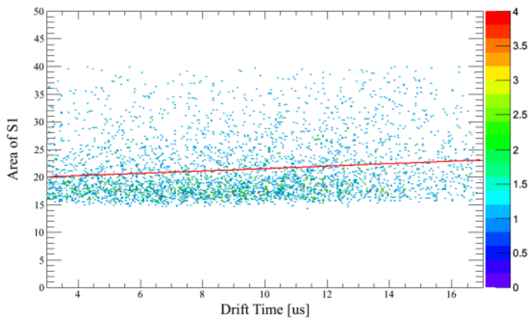
In principle, S1 signals are almost independent of perpendicular position, since the solid angles of the bottom PMTs are only slightly varying at different heights. However, the impurities in xenon, such as water, will lead to an observable attenuation of scintillation signals. Fig. 25 shows the light attenuation on different days.



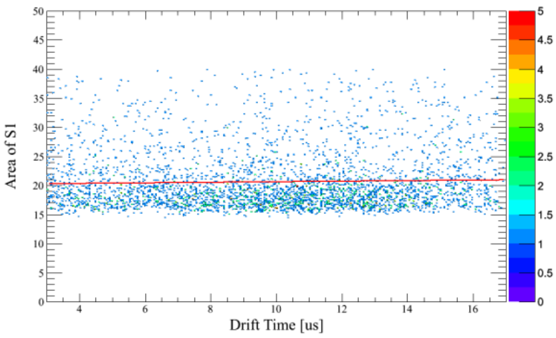
(a)



(b)



(c)



(d)

Fig. 25 Attenuation of S1 in different days.

Shorter drift time means father distance between S1 and the bottom PMTs. Fig.25 shows that the opaque impurities are removed. Also, in Fig.25, (a) is under the situation of 5.0 kV applied on the anode; (b), (c) and (d) are cases of 4.6, 5.2 and 5.6 kV respectively. The horizontal axis represents the drift time, and the perpendicular axis is the area of the S1 signals. The color bar shows the counts of different pixels; the red lines are fit curves by using the exponential function defined in ROOT.

$$f(x) = e^{p_0 + p_1 \cdot x}$$

Then we use the fitting function to modify the S1 signal by multiplying ${}^{1}/_{f(x)}$ and its normalization factor. In order to avoid confusion, the modified S1 signal is denoted as S1*.

The ratio between the areas of S2 and S1* should be only dependent on the electric potential of the anode in this experiment. Therefore, a drift time dependence on the ratio of S2/S1* is used to characterize the purity of xenon. The impurity of xenon leads the attenuation on the size of S2; thus, the ratio decreases with drift time. In fact, the characteristic drift time, defined as the electron life time, is a form of calibration of purity. For liquid xenon of certain impurities, the ratio is an exponential decay with a drift time. We fit the ratio with the drift time by an exponential function defined in ROOT.

$$f(x) = e^{p_0 - p_1 \cdot x}$$

Therefore, the slope, p_1, is the inverse of the electron life time. Fig. 26 exhibits the correlation between the ratio and drift time for four days in a row.

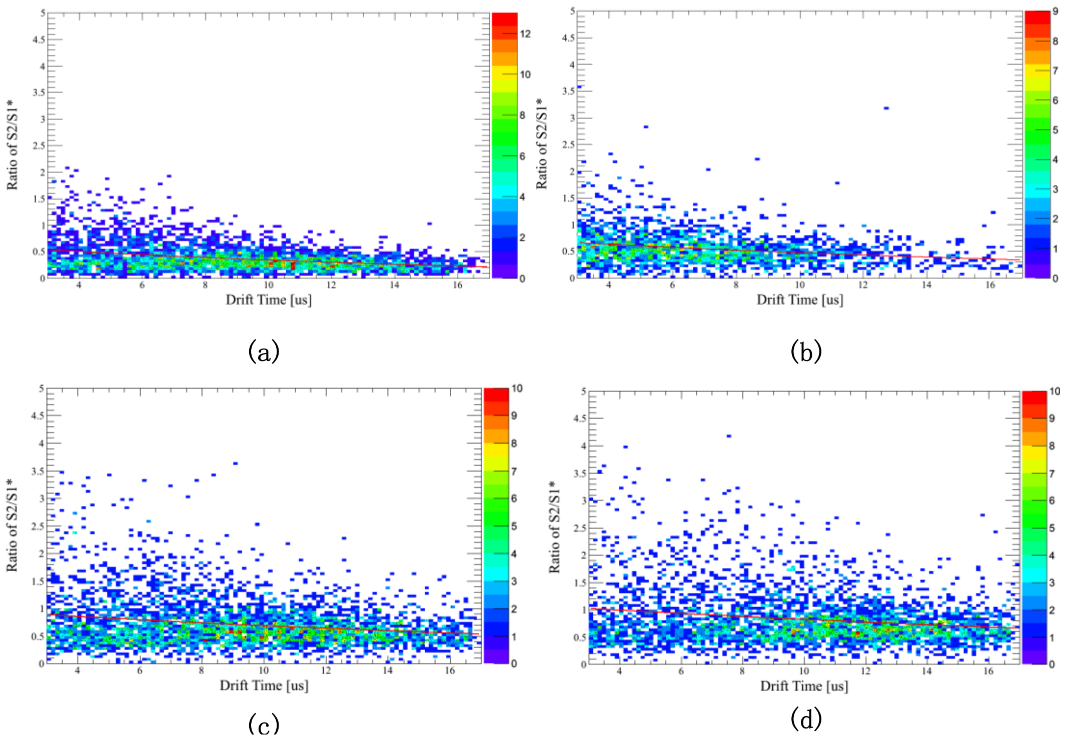


Fig. 26 Correlation between S2/S1 ratio and the drift time

In Fig. 26, (a), (b), (c), and (d) are under the same conditions as Fig. 25. Some intuitions emerge from the distribution of points. More events are attenuated at the long drift time regions of

(a) and (b). By contrast, the distribution in (d) seems to be much more uniform. As shown in the distribution, the average ratio of S2/S1* becomes bigger with the increasing high voltage on the anode.

By fitting the histogram exponentially, Fig. 27 interprets the variations of the electron life time, which is the purity by each day. The purple line in the graph is just a smooth line connecting points to show the tendency of the increasing purity.

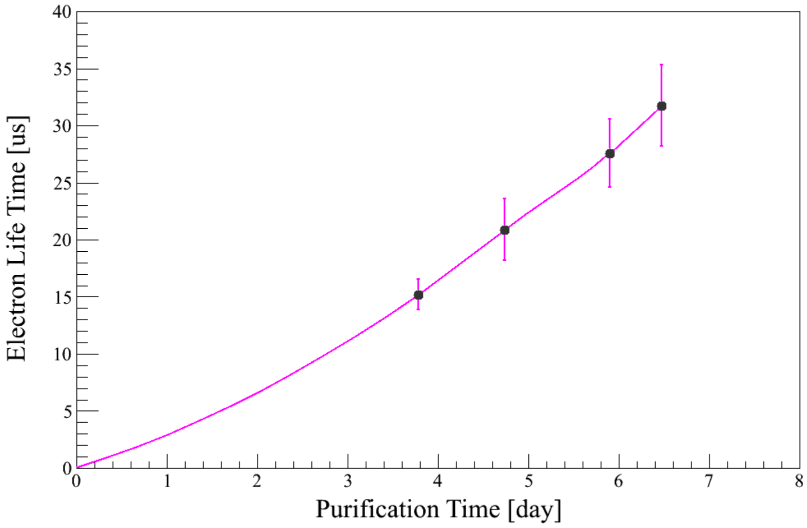


Fig. 27 Evolution of electron life time with purification time

The rate constant for electron attachment to O_2 in liquid xenon has been studied by G. Bakale et al. in 1976 [10]. The reaction of an electron with an impurity, O_2 equivalent, leads to the formation of a negative ion, and the decrease of the electron concentration $[e]$ is given as

$$\frac{d\,[e]}{dt} = -k \cdot [e] \cdot [o_2] \qquad ,$$

where $[o_2]$ is the concentration of impurity given in ${}^{mol}/_{l}$ and k is the attachment rate constant in $l/(mol \cdot s)$. The temporal variation of the electron concentration $[e]$ is then given as

$$[e(t)] = [e(0)] \cdot e^{-k \cdot [o_2] \cdot t} \qquad .$$

And the time

$$\tau = \frac{1}{k \cdot [o_2]}$$

is called the electron life time.

In their measurement, the attachment rate constant of O_2 is plotted versus the electric field strength.

In our experiment, the drift electric field is 1 kV/cm. Thus

$$k = 6.8 \times 10^{10}\ l/(mol \cdot s) \quad .$$

In the fourth day, the electron life time τ is 32 μs, apply the formula above, the electro-negative impurity concentration, the concentration of O_2 equivalent substances is derived:

$$[o_2] = \frac{1}{k\tau} = 4.596 \times 10^{-7} mol/l$$

$$[o_2]^* = \frac{[o_2] \cdot A_{Xe}}{\rho_{Xe}} = \frac{(4.596 \times 10^{-7}) \times 131.3}{3.057 \times 10^3} = 19.74\ ppb \quad .$$

According to this calculation, the purity of xenon is good enough for detector in our scale.

Using a linear fitting function, the dependence of the average ratio on anode potential is plotted in Fig. 28.

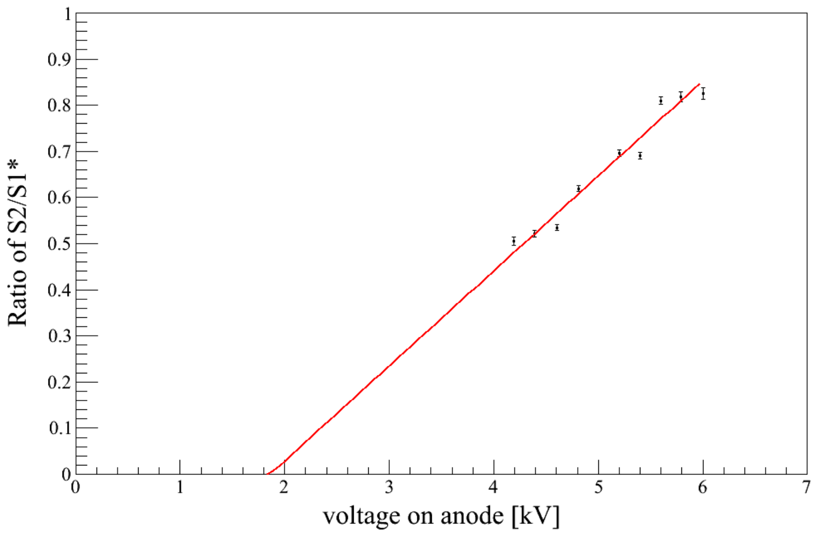


Fig. 28 Dependence of S2/S1* ratio on anode potential

The standard error of the slope 0.205968 $/kV$ is 0.00483261 $/kV$, and the χ^2 test is the default measurement for the accuracy of the fit. The NDF (Number of Degrees of Freedom) is 9, since 9 points are measured with a $\chi^2 = 97.3$. Therefore, the ratio of S2/S1 is proportional to the potential applied on the anode.

5. Summary

In this experiment, an integral experimental system was set up, which is constructed with a small gridded ionization chamber, cryogenic system, gas system, LN_2 emergency system, data acquisition system, and slow control system. We measured the dependence for the average ratio of S2/S1* on the anode potential after the modification of a direct scintillation signal. Meanwhile, the relative purity of xenon is measured by both the attenuation of light and charge signals. In this thesis, proportional scintillation is investigated for the correlation between high voltage on the anode and the amplitude of the secondary scintillation.

However, the result is not perfect due to the several reasons below. The system is still far from its equilibrium point under the purification. The impurity will lead to the attenuation of proportional scintillation. Therefore, the mean ratio on different voltages will distort the real relationship, since various data may be taken at different times. This deviation cannot be rectified because of the lack of an absolute relationship for the ratio on the electron life time. On the other hand, the disappearances of the structure of ^{137}Cs and ^{60}Co sources still need to be explained. In the next run, the trigger setup could be switched to a signal from the top PMT. The narrow window will cut down the event rate but will fix the solid angle to reduce the dependence on a geometric effect. A PTFE shield will also be used to cover the detector in the future and improve the light collection efficiency. The charge signal is not used in an event cut this time but will be added as a criterion to cut the events for the next run.

Bibliography

[1] Miyajima, M. A self-triggered liquid xenon drift chamber by the use of proportional ionization or proportional scintillation. [J] Nuclear instruments and methods (1979).160(2): 239-246.

[2] Masuda, K. A liquid xenon proportional scintillation counter [J]. Nuclear instruments and methods (1979).160(2): 247-253.

[3] Derenzo, S. E., T. S. Mast, et al. Electron avalanche in liquid xenon [J]. Physical Review A (1974). 9(6): 2582.

[4] Muller, R. A., S. E. Derenzo, et al. Liquid-filled proportional counter [J]. Physical Review Letters (1971). 27(8): 532-535.

[5] Ni, K. Development of a liquid xenon time projection chamber for the XENON dark matter search [D]. (2006).

[6] Aprile, E., J. Angle, et al. Design and performance of the XENON10 dark matter experiment [J/OL]. (2010). Arxiv preprint arXiv: 1001.2834.

[7] Aprile, E. Liquid xenon detectors for particle physics and astrophysics [J/OL]. Reviews of modern physics (2010). 82(3): 2053-2097.

[8] Xie Yigang, Chen Chang, et al.. Li Zi Tan Ce Qi Yu Shu Ju Huo Qu[M], (2003) Science Press.

[9] Collaboration, X., E. Aprile, et al. The XENON100 Dark Matter Experiment [J/OL]. (2011).Arxiv preprint arXiv: 1107.2155.

[10] Bakale, G., U. Sowada, et al. Effect of an electric field on electron attachment to sulfur hexafluoride, nitrous oxide, and molecular oxygen in liquid argon and xenon. [J]. The Journal of Physical Chemistry (1976) 80(23): 2556-2559.

Acknowledgements

This thesis would not have been possible without the guidance and the help of several individuals who contributed to this study.

First and foremost, my utmost gratitude is to Professor Ji Xiangdong，whose encouragement, guidance, and enlightenment I will never forget. Professor Ji has given so much inspiration to me during this research. As a master in science, he is longing for finding time to investigate the beauty of the natural world to admire and debate elegant theories with his students. His passion for pursuing the truth in physics provides profound insights into the universe. Also, his perseverance in the face of obstacles impressed me deeply. I have benefited profoundly form working by his side.

I consider it to be a great honor to work with Professor Karl-Ludwig Giboni. He is an expert and pioneer in xenon detectors. I owe my deepest gratitude to his patient instruction on every aspect of the experimental process. With his guidance of implementing scientific philosophic principles, I will hold them dear in my career.

It gives me great pleasure in acknowledging the support and help of Dr. Liu Jianglai，who offers so much suggestions and clues on the process of data analysis in this experiment. I am deeply touched by his great enthusiasm and joy for scientific research when contemplating unknown phenomena.

It is with immense gratitude that I acknowledge the support and help of Dr. Ni Kaixuan and Dr. Zhao Li. They are both energetic and professional in their academic inquiries. I would like to thank them for teaching me the knowledge of xenon detectors and technologies in our experiments.

I cannot find words to express my gratitude to B.Sc. Gao Fei. This thesis would have remained a dream if he did not provide his unrelenting support. Without his help and instruction on ROOT, Linux, and C++, the data analysis of this thesis would have been never accomplished.

I want to also thank B.Sc. Wei Yuehuan, Xiao Mengjiao, and Xiao Yanyang. I sincerely appreciate their help and suggestions regarding detector technology, PMT properties, or the art of programming.

I would like to thank Dr. Michael Schubnell, Associate Research Scientist in the Physics Department at the University of Michigan, who developed the slow control system for PANDAX. As a subsystem to that, the slow control system for this experiment makes my work easier.

Last but not the least, my parents, my girlfriend: Ma Ningxin, and my friends: especially Song Anyi and Wang Peijiang, for their continuous encouragement, which supported my dream to be a physicist wholeheartedly.

Appendix

Algorithm 1 *DAPSLXe.cpp*

```
#include <TApplication.h>
#include <TCanvas.h>
#include <TTree.h>
#include <TSQLServer.h>
#include <TSQLRow.h>
#include <TSQLResult.h>
#include <TF1.h>
#include <TH1F.h>
#include <TF2.h>
#include <TFile.h>
#include <TGraph.h>
#include <vector>
#include <string>
#include <TROOT.h>
#include <iostream>
#include <fstream>
#include "TStopwatch.h"
#include "TMath.h"
#include <sstream>
#include <unistd.h>
#include "Rtypes.h"
using namespace std;
using std::string;
using std::cout;
using std::endl;
using std::ostringstream;

string to_string(int t)
{
ostringstream res;
res<<t;
return res.str();
}

void swapint(int &val1,int &val2)
{
        int temp = val1;
        val1 = val2;
        val2 = temp;
}

void swapfloat(float &val1,float &val2)
{
        float temp = val1;
        val1 = val2;
        val2 = temp;
}

int main(int argc, char **argv)
{
       long iNbEventsToProcess    =    -1;
```

```
gROOT->ProcessLine("#include <vector>");

TString raw_data_path   =   argv[ 3 ];
Char_t *root_data_path =   "/home/andy/root_file";


    int file_number_low = atoi(argv[ 4 ]);
    int file_number_high = atoi(argv[ 5 ]);

    Char_t *in_file_name   =   argv[ 1 ];
    Char_t *out_file_name   =   argv[ 2 ];

    const static int number_of_files = file_number_high;
Char_t out_path[ 1000 ];
sprintf(out_path,"%s",root_data_path);

TString data_file[ number_of_files ];
    Char_t outfile[ 1000 ];
sprintf(outfile, "%s/%s.root", out_path, out_file_name);
    int zeros_tmp = 0;


   for (int i = 0;i<file_number_high+1;i++)
     {
       if(i)
       {
       zeros_tmp = 4-int(log10(1.0*i));
       }
       else
       zeros_tmp = 4;
       data_file[ i ] = raw_data_path;

       data_file[ i ]+=in_file_name;
       for(int j=0;j<zeros_tmp;j++)
          {
            data_file[ i ]+=to_string(0);
          }
         data_file[ i ]+=to_string(i);
         data_file[ i ]+=".txt";
      }

cout << "Save to file" << outfile << endl;

stringstream hTmp("");
    const static int nb_samples=16000;
    const static float window_length=20.0;
    const static int Max_Peak=100;
    const static float pre_signal_threshold=0.1;

    const static int max_pulse_width=int(500./(window_length*1000/nb_samples));

    #include "Generate_Variables.h"

    float voltage1[ nb_samples ];
    float voltage2[ nb_samples ];
```

```
	TFile *ofile = new TFile(outfile,"RECREATE");
	TTree *T1= new TTree("T1","Root Data in Andy's Lab, First-Level Analysis, T1");
	TTree *T2= new TTree("T2","Root Data in Andy's Lab, Second-Level Analysis, T2");

		#include "Generate_Branches.h"

	string temp_string;
	TStopwatch timer;
	Double_t t_passed = 0, t_left;
		iNbEventsToProcess = file_number_high-file_number_low;
	cout << "will loop over " << iNbEventsToProcess << " events " << endl;
	for(int jentry=file_number_low; jentry< file_number_high+1; jentry++)
	{
		if(jentry%100 == 0 && jentry!=0) {
			t_passed += timer.CpuTime();
			timer.Start();
			t_left = t_passed/(float)jentry*(float)(iNbEventsToProcess-jentry) ;
			cerr << "Job progress : " << (float)jentry/(float)iNbEventsToProcess*100 << "% Time
passed/left: " << t_passed << "/" << t_left << " sec    \r";
		}
		EventID=jentry;

		#include "Initialization.h"

		ifstream f(data_file[ jentry ],ios::in);
		for(int i=0;i<nb_samples;i++)
		{
		   f>>voltage1[ i ];
		   f>>voltage2[ i ];
		 }
		f.close();

	    TH1F *vol1=new TH1F("vol1","",nb_samples,0,nb_samples);
	    TH1F *vol2=new TH1F("vol2","",nb_samples,0,nb_samples);

	  for (int i=0;i<nb_samples;i++)
	    {
	     vol1->SetBinContent(i+1,-1.0*voltage1[ i ]);
	     vol2->SetBinContent(i+1,-1.0*voltage2[ i ]);
	     }

	float sum_baseline=0;
	float square_baseline=0;
	float sigma_baseline=0;
	int sample_baseline=1;

	#include "Ch1Analize.h"

	#include "Ch2Analize.h"

	  vol1->Delete();
	  vol2->Delete();

	  T1->Fill();
	  T2->Fill();
	}
	    T1->Write();
```

```
        T2->Write();
        ofile->Close();

    return(0);
}
```

Generate_Variables.h

```
float BaseLineCh1,BaseLineCh2,BaseLineCh3;
float AreaTotCh1,AreaTotCh2,AreaTotCh3;
int NbPeaksCh1,NbPeaksCh2,NbPeaksCh3;
int EventID;
float PeakTotCh1[Max_Peak],PeakTotCh2[Max_Peak],PeakTotCh3[Max_Peak];
int PeakTimeCh1[Max_Peak],PeakTimeCh2[Max_Peak],PeakTimeCh3[Max_Peak];
int PeakLeftMostCh1[Max_Peak],PeakLeftMostCh2[Max_Peak],PeakLeftMostCh3[Max_Peak];
int PeakRightMostCh1[Max_Peak],PeakRightMostCh2[Max_Peak],PeakRightMostCh3[Max_Peak];
float PeakHeightCh1[Max_Peak],PeakHeightCh2[Max_Peak],PeakHeightCh3[Max_Peak];
floatPeakHighWidthCh1[Max_Peak],PeakHighWidthCh2[Max_Peak],PeakHighWidthCh3[Max_Peak]
float PeakHalfWidthCh1[Max_Peak],PeakHalfWidthCh2[Max_Peak],PeakHalfWidthCh3[Max_Peak];
float PeakLowWidthCh1[Max_Peak],PeakLowWidthCh2[Max_Peak],PeakLowWidthCh3[Max_Peak];
float PeakLeftLowCh1[Max_Peak],PeakLeftLowCh2[Max_Peak],PeakLeftLowCh3[Max_Peak];
float PeakLeftHalfCh1[Max_Peak],PeakLeftHalfCh2[Max_Peak],PeakLeftHalfCh3[Max_Peak];
float PeakLeftHighCh1[Max_Peak],PeakLeftHighCh2[Max_Peak],PeakLeftHighCh3[Max_Peak];
float PeakRightLowCh1[Max_Peak],PeakRightLowCh2[Max_Peak],PeakRightLowCh3[Max_Peak];
float PeakRightHalfCh1[Max_Peak],PeakRightHalfCh2[Max_Peak],PeakRightHalfCh3[Max_Peak];
float PeakRightHighCh1[Max_Peak],PeakRightHighCh2[Max_Peak],PeakRightHighCh3[Max_Peak];
float PeakRiseTimeCh1[Max_Peak],PeakRiseTimeCh2[Max_Peak],PeakRiseTimeCh3[Max_Peak];
float PeakDropTimeCh1[Max_Peak],PeakDropTimeCh2[Max_Peak],PeakDropTimeCh3[Max_Peak];
int NbS1PeaksCh1,NbS1PeaksCh2,NbS1PeaksCh3;
float S1TotCh1[Max_Peak],S1TotCh2[Max_Peak],S1TotCh3[Max_Peak];
int S1TimeCh1[Max_Peak],S1TimeCh2[Max_Peak],S1TimeCh3[Max_Peak];
int S1LeftMostCh1[Max_Peak],S1LeftMostCh2[Max_Peak],S1LeftMostCh3[Max_Peak];
int S1RightMostCh1[Max_Peak],S1RightMostCh2[Max_Peak],S1RightMostCh3[Max_Peak];
float S1HeightCh1[Max_Peak],S1HeightCh2[Max_Peak],S1HeightCh3[Max_Peak];
float S1HighWidthCh1[Max_Peak],S1HighWidthCh2[Max_Peak],S1HighWidthCh3[Max_Peak];
float S1HalfWidthCh1[Max_Peak],S1HalfWidthCh2[Max_Peak],S1HalfWidthCh3[Max_Peak];
float S1LowWidthCh1[Max_Peak],S1LowWidthCh2[Max_Peak],S1LowWidthCh3[Max_Peak];
float S1LeftLowCh1[Max_Peak],S1LeftLowCh2[Max_Peak],S1LeftLowCh3[Max_Peak];
float S1LeftHalfCh1[Max_Peak],S1LeftHalfCh2[Max_Peak],S1LeftHalfCh3[Max_Peak];
float S1LeftHighCh1[Max_Peak],S1LeftHighCh2[Max_Peak],S1LeftHighCh3[Max_Peak];
float S1RightLowCh1[Max_Peak],S1RightLowCh2[Max_Peak],S1RightLowCh3[Max_Peak];
float S1RightHalfCh1[Max_Peak],S1RightHalfCh2[Max_Peak],S1RightHalfCh3[Max_Peak];
float S1RightHighCh1[Max_Peak],S1RightHighCh2[Max_Peak],S1RightHighCh3[Max_Peak];
float S1RiseTimeCh1[Max_Peak],S1RiseTimeCh2[Max_Peak],S1RiseTimeCh3[Max_Peak];
float S1DropTimeCh1[Max_Peak],S1DropTimeCh2[Max_Peak],S1DropTimeCh3[Max_Peak];

int NbS2PeaksCh1,NbS2PeaksCh2,NbS2PeaksCh3;
float S2TotCh1[Max_Peak],S2TotCh2[Max_Peak],S2TotCh3[Max_Peak];
int S2TimeCh1[Max_Peak],S2TimeCh2[Max_Peak],S2TimeCh3[Max_Peak];
int S2LeftMostCh1[Max_Peak],S2LeftMostCh2[Max_Peak],S2LeftMostCh3[Max_Peak];
int S2RightMostCh1[Max_Peak],S2RightMostCh2[Max_Peak],S2RightMostCh3[Max_Peak];
float S2HeightCh1[Max_Peak],S2HeightCh2[Max_Peak],S2HeightCh3[Max_Peak];
float S2HighWidthCh1[Max_Peak],S2HighWidthCh2[Max_Peak],S2HighWidthCh3[Max_Peak];
float S2HalfWidthCh1[Max_Peak],S2HalfWidthCh2[Max_Peak],S2HalfWidthCh3[Max_Peak];
```

```
float S2LowWidthCh1[Max_Peak],S2LowWidthCh2[Max_Peak],S2LowWidthCh3[Max_Peak];
float S2LeftLowCh1[Max_Peak],S2LeftLowCh2[Max_Peak],S2LeftLowCh3[Max_Peak];
float S2LeftHalfCh1[Max_Peak],S2LeftHalfCh2[Max_Peak],S2LeftHalfCh3[Max_Peak];
float S2LeftHighCh1[Max_Peak],S2LeftHighCh2[Max_Peak],S2LeftHighCh3[Max_Peak];
float S2RightLowCh1[Max_Peak],S2RightLowCh2[Max_Peak],S2RightLowCh3[Max_Peak];
float S2RightHalfCh1[Max_Peak],S2RightHalfCh2[Max_Peak],S2RightHalfCh3[Max_Peak];
float S2RightHighCh1[Max_Peak],S2RightHighCh2[Max_Peak],S2RightHighCh3[Max_Peak];
float S2RiseTimeCh1[Max_Peak],S2RiseTimeCh2[Max_Peak],S2RiseTimeCh3[Max_Peak];
float S2DropTimeCh1[Max_Peak],S2DropTimeCh2[Max_Peak],S2DropTimeCh3[Max_Peak];
```

Generate_Branches.h

```
T1->SetMaxTreeSize(100e9);
T1->Branch("BaseLineCh1", &BaseLineCh1, "BaseLineCh1/F");
T1->Branch("AreaTotCh1", &AreaTotCh1, "AreaTotCh1/F");

T1->Branch("NbPeaksCh1", &NbPeaksCh1, "NbPeaksCh1/I");
T1->Branch("EventID", &EventID, "EventID/I");
T1->Branch("PeakTotCh1", PeakTotCh1, "PeakTotCh1[NbPeaksCh1]/F");
T1->Branch("PeakTimeCh1", PeakTimeCh1, "PeakTimeCh1[NbPeaksCh1]/I");
T1->Branch("PeakLeftMostCh1", PeakLeftMostCh1, "PeakLeftMostCh1[NbPeaksCh1]/I");
T1->Branch("PeakRightMostCh1", PeakRightMostCh1, "PeakRightMostCh1[NbPeaksCh1]/I");
T1->Branch("PeakHeightCh1", PeakHeightCh1, "PeakHeightCh1[NbPeaksCh1]/F");
T1->Branch("PeakHighWidthCh1", PeakHighWidthCh1, "PeakHighWidthCh1[NbPeaksCh1]/F");
T1->Branch("PeakHalfWidthCh1", PeakHalfWidthCh1, "PeakHalfWidthCh1[NbPeaksCh1]/F");
T1->Branch("PeakLowWidthCh1", PeakLowWidthCh1, "PeakLowWidthCh1[NbPeaksCh1]/F");
T1->Branch("PeakLeftLowCh1", PeakLeftLowCh1, "PeakLeftLowCh1[NbPeaksCh1]/F");
T1->Branch("PeakLeftHalfCh1", PeakLeftHalfCh1, "PeakLeftHalfCh1[NbPeaksCh1]/F");
T1->Branch("PeakLeftHighCh1", PeakLeftHighCh1, "PeakLeftHighCh1[NbPeaksCh1]/F");
T1->Branch("PeakRightLowCh1", PeakRightLowCh1, "PeakRightLowCh1[NbPeaksCh1]/F");
T1->Branch("PeakRightHalfCh1", PeakRightHalfCh1, "PeakRightHalfCh1[NbPeaksCh1]/F");
T1->Branch("PeakRightHighCh1", PeakRightHighCh1, "PeakRightHighCh1[NbPeaksCh1]/F");
T1->Branch("PeakRiseTimeCh1", PeakRiseTimeCh1, "PeakRiseTimeCh1[NbPeaksCh1]/F");
T1->Branch("PeakDropTimeCh1", PeakDropTimeCh1, "PeakDropTimeCh1[NbPeaksCh1]/F");

T1->Branch("NbPeaksCh2", &NbPeaksCh2, "NbPeaksCh2/I");
T1->Branch("PeakTotCh2", PeakTotCh2, "PeakTotCh2[NbPeaksCh2]/F");
T1->Branch("PeakTimeCh2", PeakTimeCh2, "PeakTimeCh2[NbPeaksCh2]/I");
T1->Branch("PeakLeftMostCh2", PeakLeftMostCh2, "PeakLeftMostCh2[NbPeaksCh2]/I");
T1->Branch("PeakRightMostCh2", PeakRightMostCh2, "PeakRightMostCh2[NbPeaksCh2]/I");
T1->Branch("PeakHeightCh2", PeakHeightCh2, "PeakHeightCh2[NbPeaksCh2]/F");
T1->Branch("PeakHighWidthCh2", PeakHighWidthCh2, "PeakHighWidthCh2[NbPeaksCh2]/F");
T1->Branch("PeakHalfWidthCh2", PeakHalfWidthCh2, "PeakHalfWidthCh2[NbPeaksCh2]/F");
T1->Branch("PeakLowWidthCh2", PeakLowWidthCh2, "PeakLowWidthCh2[NbPeaksCh2]/F");
T1->Branch("PeakLeftLowCh2", PeakLeftLowCh2, "PeakLeftLowCh2[NbPeaksCh2]/F");
T1->Branch("PeakLeftHalfCh2", PeakLeftHalfCh2, "PeakLeftHalfCh2[NbPeaksCh2]/F");
T1->Branch("PeakLeftHighCh2", PeakLeftHighCh2, "PeakLeftHighCh2[NbPeaksCh2]/F");
T1->Branch("PeakRightLowCh2", PeakRightLowCh2, "PeakRightLowCh2[NbPeaksCh2]/F");
T1->Branch("PeakRightHalfCh2", PeakRightHalfCh2, "PeakRightHalfCh2[NbPeaksCh2]/F");
T1->Branch("PeakRightHighCh2", PeakRightHighCh2, "PeakRightHighCh2[NbPeaksCh2]/F");
T1->Branch("PeakRiseTimeCh2", PeakRiseTimeCh2, "PeakRiseTimeCh2[NbPeaksCh2]/F");
T1->Branch("PeakDropTimeCh2", PeakDropTimeCh2, "PeakDropTimeCh2[NbPeaksCh2]/F");

T2->SetMaxTreeSize(100e9);

T2->Branch("NbS1PeaksCh1", &NbS1PeaksCh1, "NbS1PeaksCh1/I");
```

```
T2->Branch("S1TotCh1", S1TotCh1, "S1TotCh1[NbS1PeaksCh1]/F");
T2->Branch("S1TimeCh1", S1TimeCh1, "S1TimeCh1[NbS1PeaksCh1]/I");
T2->Branch("S1LeftMostCh1", S1LeftMostCh1, "S1LeftMostCh1[NbS1PeaksCh1]/I");
T2->Branch("S1RightMostCh1", S1RightMostCh1, "S1RightMostCh1[NbS1PeaksCh1]/I");
T2->Branch("S1HeightCh1", S1HeightCh1, "S1HeightCh1[NbS1PeaksCh1]/F");
T2->Branch("S1HighWidthCh1", S1HighWidthCh1, "S1HighWidthCh1[NbS1PeaksCh1]/F");
T2->Branch("S1HalfWidthCh1", S1HalfWidthCh1, "S1HalfWidthCh1[NbS1PeaksCh1]/F");
T2->Branch("S1LowWidthCh1", S1LowWidthCh1, "S1LowWidthCh1[NbS1PeaksCh1]/F");
T2->Branch("S1LeftLowCh1", S1LeftLowCh1, "S1LeftLowCh1[NbS1PeaksCh1]/F");
T2->Branch("S1LeftHalfCh1", S1LeftHalfCh1, "S1LeftHalfCh1[NbS1PeaksCh1]/F");
T2->Branch("S1LeftHighCh1", S1LeftHighCh1, "S1LeftHighCh1[NbS1PeaksCh1]/F");
T2->Branch("S1RightLowCh1", S1RightLowCh1, "S1RightLowCh1[NbS1PeaksCh1]/F");
T2->Branch("S1RightHalfCh1", S1RightHalfCh1, "S1RightHalfCh1[NbS1PeaksCh1]/F");
T2->Branch("S1RightHighCh1", S1RightHighCh1, "S1RightHighCh1[NbS1PeaksCh1]/F");
T2->Branch("S1RiseTimeCh1", S1RiseTimeCh1, "S1RiseTimeCh1[NbS1PeaksCh1]/F");
T2->Branch("S1DropTimeCh1", S1DropTimeCh1, "S1DropTimeCh1[NbS1PeaksCh1]/F");

T2->Branch("NbS1PeaksCh2", &NbS1PeaksCh2, "NbS1PeaksCh2/I");
T2->Branch("S1TotCh2", S1TotCh2, "S1TotCh2[NbS1PeaksCh2]/F");
T2->Branch("S1TimeCh2", S1TimeCh2, "S1TimeCh2[NbS1PeaksCh2]/I");
T2->Branch("S1LeftMostCh2", S1LeftMostCh2, "S1LeftMostCh2[NbS1PeaksCh2]/I");
T2->Branch("S1RightMostCh2", S1RightMostCh2, "S1RightMostCh2[NbS1PeaksCh2]/I");
T2->Branch("S1HeightCh2", S1HeightCh2, "S1HeightCh2[NbS1PeaksCh2]/F");
T2->Branch("S1HighWidthCh2", S1HighWidthCh2, "S1HighWidthCh2[NbS1PeaksCh2]/F");
T2->Branch("S1HalfWidthCh2", S1HalfWidthCh2, "S1HalfWidthCh2[NbS1PeaksCh2]/F");
T2->Branch("S1LowWidthCh2", S1LowWidthCh2, "S1LowWidthCh2[NbS1PeaksCh2]/F");
T2->Branch("S1LeftLowCh2", S1LeftLowCh2, "S1LeftLowCh2[NbS1PeaksCh2]/F");
T2->Branch("S1LeftHalfCh2", S1LeftHalfCh2, "S1LeftHalfCh2[NbS1PeaksCh2]/F");
T2->Branch("S1LeftHighCh2", S1LeftHighCh2, "S1LeftHighCh2[NbS1PeaksCh2]/F");
T2->Branch("S1RightLowCh2", S1RightLowCh2, "S1RightLowCh2[NbS1PeaksCh2]/F");
T2->Branch("S1RightHalfCh2", S1RightHalfCh2, "S1RightHalfCh2[NbS1PeaksCh2]/F");
T2->Branch("S1RightHighCh2", S1RightHighCh2, "S1RightHighCh2[NbS1PeaksCh2]/F");
T2->Branch("S1RiseTimeCh2", S1RiseTimeCh2, "S1RiseTimeCh2[NbS1PeaksCh2]/F");
T2->Branch("S1DropTimeCh2", S1DropTimeCh2, "S1DropTimeCh2[NbS1PeaksCh2]/F");

T2->Branch("NbS2PeaksCh1", &NbS2PeaksCh1, "NbS2PeaksCh1/I");
T2->Branch("S2TotCh1", S2TotCh1, "S2TotCh1[NbS2PeaksCh1]/F");
T2->Branch("S2TimeCh1", S2TimeCh1, "S2TimeCh1[NbS2PeaksCh1]/I");
T2->Branch("S2LeftMostCh1", S2LeftMostCh1, "S2LeftMostCh1[NbS2PeaksCh1]/I");
T2->Branch("S2RightMostCh1", S2RightMostCh1, "S2RightMostCh1[NbS2PeaksCh1]/I");
T2->Branch("S2HeightCh1", S2HeightCh1, "S2HeightCh1[NbS2PeaksCh1]/F");
T2->Branch("S2HighWidthCh1", S2HighWidthCh1, "S2HighWidthCh1[NbS2PeaksCh1]/F");
T2->Branch("S2HalfWidthCh1", S2HalfWidthCh1, "S2HalfWidthCh1[NbS2PeaksCh1]/F");
T2->Branch("S2LowWidthCh1", S2LowWidthCh1, "S2LowWidthCh1[NbS2PeaksCh1]/F");
T2->Branch("S2LeftLowCh1", S2LeftLowCh1, "S2LeftLowCh1[NbS2PeaksCh1]/F");
T2->Branch("S2LeftHalfCh1", S2LeftHalfCh1, "S2LeftHalfCh1[NbS2PeaksCh1]/F");
T2->Branch("S2LeftHighCh1", S2LeftHighCh1, "S2LeftHighCh1[NbS2PeaksCh1]/F");
T2->Branch("S2RightLowCh1", S2RightLowCh1, "S2RightLowCh1[NbS2PeaksCh1]/F");
T2->Branch("S2RightHalfCh1", S2RightHalfCh1, "S2RightHalfCh1[NbS2PeaksCh1]/F");
T2->Branch("S2RightHighCh1", S2RightHighCh1, "S2RightHighCh1[NbS2PeaksCh1]/F");
T2->Branch("S2RiseTimeCh1", S2RiseTimeCh1, "S2RiseTimeCh1[NbS2PeaksCh1]/F");
T2->Branch("S2DropTimeCh1", S2DropTimeCh1, "S2DropTimeCh1[NbS2PeaksCh1]/F");

T2->Branch("NbS2PeaksCh2", &NbS2PeaksCh2, "NbS2PeaksCh2/I");
T2->Branch("S2TotCh2", S2TotCh2, "S2TotCh2[NbS2PeaksCh2]/F");
T2->Branch("S2TimeCh2", S2TimeCh2, "S2TimeCh2[NbS2PeaksCh2]/I");
T2->Branch("S2LeftMostCh2", S2LeftMostCh2, "S2LeftMostCh2[NbS2PeaksCh2]/I");
```

```
T2->Branch("S2RightMostCh2", S2RightMostCh2, "S2RightMostCh2[NbS2PeaksCh2]/I"); /
T2->Branch("S2HeightCh2", S2HeightCh2, "S2HeightCh2[NbS2PeaksCh2]/F");
T2->Branch("S2HighWidthCh2", S2HighWidthCh2, "S2HighWidthCh2[NbS2PeaksCh2]/F");
T2->Branch("S2HalfWidthCh2", S2HalfWidthCh2, "S2HalfWidthCh2[NbS2PeaksCh2]/F");
T2->Branch("S2LowWidthCh2", S2LowWidthCh2, "S2LowWidthCh2[NbS2PeaksCh2]/F");
T2->Branch("S2LeftLowCh2", S2LeftLowCh2, "S2LeftLowCh2[NbS2PeaksCh2]/F");
T2->Branch("S2LeftHalfCh2", S2LeftHalfCh2, "S2LeftHalfCh2[NbS2PeaksCh2]/F");
T2->Branch("S2LeftHighCh2", S2LeftHighCh2, "S2LeftHighCh2[NbS2PeaksCh2]/F");
T2->Branch("S2RightLowCh2", S2RightLowCh2, "S2RightLowCh2[NbS2PeaksCh2]/F");
T2->Branch("S2RightHalfCh2", S2RightHalfCh2, "S2RightHalfCh2[NbS2PeaksCh2]/F");
T2->Branch("S2RightHighCh2", S2RightHighCh2, "S2RightHighCh2[NbS2PeaksCh2]/F");
T2->Branch("S2RiseTimeCh2", S2RiseTimeCh2, "S2RiseTimeCh2[NbS2PeaksCh2]/F");
T2->Branch("S2DropTimeCh2", S2DropTimeCh2, "S2DropTimeCh2[NbS2PeaksCh2]/F");
```

Initialization.h

```
NbPeaksCh1 = 0; NbPeaksCh2 = 0;NbPeaksCh3 = 0;
NbS1PeaksCh1 = 0; NbS1PeaksCh2 = 0; NbS1PeaksCh3 = 0;
NbS2PeaksCh1 = 0; NbS2PeaksCh2 = 0; NbS2PeaksCh3 = 0;
BaseLineCh1 = 0; BaseLineCh2 = 0;BaseLineCh3 = 0;
AreaTotCh1 = 0; AreaTotCh2 = 0;AreaTotCh3 = 0;
for(int i = 0;i<Max_Peak;i++)
{
PeakTotCh1[ i ] = 0; PeakTotCh2[ i ] = 0;PeakTotCh3[ i ] = 0;
PeakTimeCh1[ i ] = 0; PeakTimeCh2[ i ] = 0;PeakTimeCh3[ i ] = 0;
PeakLeftMostCh1[ i ] = 0; PeakLeftMostCh2[ i ] = 0;PeakLeftMostCh3[ i ] = 0;
PeakLeftLowCh1[ i ] = 0; PeakLeftLowCh2[ i ] = 0;PeakLeftLowCh3[ i ] = 0;
PeakLeftHalfCh1[ i ] = 0; PeakLeftHalfCh2[ i ] = 0;PeakLeftHalfCh3[ i ] = 0;
PeakLeftHighCh1[ i ] = 0; PeakLeftHighCh2[ i ] = 0;PeakLeftHighCh3[ i ] = 0;
PeakHeightCh1[ i ] = 0; PeakHeightCh2[ i ] = 0;PeakHeightCh3[ i ] = 0;
PeakRightMostCh1[ i ] = 0; PeakRightMostCh2[ i ] = 0;PeakRightMostCh3[ i ] = 0;
PeakRightLowCh1[ i ] = 0; PeakRightLowCh2[ i ] = 0;PeakRightLowCh3[ i ] = 0;
PeakRightHalfCh1[ i ] = 0; PeakRightHalfCh2[ i ] = 0;PeakRightHalfCh3[ i ] = 0;
PeakRightHighCh1[ i ] = 0; PeakRightHighCh2[ i ] = 0;PeakRightHighCh3[ i ] = 0;
PeakLowWidthCh1[ i ] = 0; PeakLowWidthCh2[ i ] = 0;PeakLowWidthCh3[ i ] = 0;
PeakHighWidthCh1[ i ] = 0; PeakHighWidthCh2[ i ] = 0;PeakHighWidthCh3[ i ] = 0;
PeakHalfWidthCh1[ i ] = 0; PeakHalfWidthCh2[ i ] = 0;PeakHalfWidthCh3[ i ] = 0;
PeakRiseTimeCh1[ i ] = 0; PeakRiseTimeCh2[ i ] = 0;PeakRiseTimeCh3[ i ] = 0;
PeakDropTimeCh1[ i ] = 0; PeakDropTimeCh2[ i ] = 0;PeakDropTimeCh3[ i ] = 0;

S1TotCh1[ i ] = 0; S1TotCh2[ i ] = 0;S1TotCh3[ i ] = 0;
S1TimeCh1[ i ] = 0; S1TimeCh2[ i ] = 0;S1TimeCh3[ i ] = 0;
S1LeftMostCh1[ i ] = 0; S1LeftMostCh2[ i ] = 0;S1LeftMostCh3[ i ] = 0;
S1LeftLowCh1[ i ] = 0; S1LeftLowCh2[ i ] = 0;S1LeftLowCh3[ i ] = 0;
S1LeftHalfCh1[ i ] = 0; S1LeftHalfCh2[ i ] = 0;S1LeftHalfCh3[ i ] = 0;
S1LeftHighCh1[ i ] = 0; S1LeftHighCh2[ i ] = 0;S1LeftHighCh3[ i ] = 0;
S1HeightCh1[ i ] = 0; S1HeightCh2[ i ] = 0;S1HeightCh3[ i ] = 0;
S1RightMostCh1[ i ] = 0; S1RightMostCh2[ i ] = 0;S1RightMostCh3[ i ] = 0;
S1RightLowCh1[ i ] = 0; S1RightLowCh2[ i ] = 0;S1RightLowCh3[ i ] = 0;
S1RightHalfCh1[ i ] = 0; S1RightHalfCh2[ i ] = 0;S1RightHalfCh3[ i ] = 0;
S1RightHighCh1[ i ] = 0; S1RightHighCh2[ i ] = 0;S1RightHighCh3[ i ] = 0;
S1LowWidthCh1[ i ] = 0; S1LowWidthCh2[ i ] = 0;S1LowWidthCh3[ i ] = 0;
S1HighWidthCh1[ i ] = 0; S1HighWidthCh2[ i ] = 0;S1HighWidthCh3[ i ] = 0;
S1HalfWidthCh1[ i ] = 0; S1HalfWidthCh2[ i ] = 0;S1HalfWidthCh3[ i ] = 0;
S1RiseTimeCh1[ i ] = 0; S1RiseTimeCh2[ i ] = 0;S1RiseTimeCh3[ i ] = 0;
S1DropTimeCh1[ i ] = 0; S1DropTimeCh2[ i ] = 0;S1DropTimeCh3[ i ] = 0;
```

```
S2TotCh1[ i ] = 0; S2TotCh2[ i ] = 0;S2TotCh3[ i ] = 0;
S2TimeCh1[ i ] = 0; S2TimeCh2[ i ] = 0;S2TimeCh3[ i ] = 0;
S2LeftMostCh1[ i ] = 0; S2LeftMostCh2[ i ] = 0;S2LeftMostCh3[ i ] = 0;
S2LeftLowCh1[ i ] = 0; S2LeftLowCh2[ i ] = 0;S2LeftLowCh3[ i ] = 0;
S2LeftHalfCh1[ i ] = 0; S2LeftHalfCh2[ i ] = 0;S2LeftHalfCh3[ i ] = 0;
S2LeftHighCh1[ i ] = 0; S2LeftHighCh2[ i ] = 0;S2LeftHighCh3[ i ] = 0;
S2HeightCh1[ i ] = 0; S2HeightCh2[ i ] = 0;S2HeightCh3[ i ] = 0;
S2RightMostCh1[ i ] = 0; S2RightMostCh2[ i ] = 0;S2RightMostCh3[ i ] = 0;
S2RightLowCh1[ i ] = 0; S2RightLowCh2[ i ] = 0;S2RightLowCh3[ i ] = 0;
S2RightHalfCh1[ i ] = 0; S2RightHalfCh2[ i ] = 0;S2RightHalfCh3[ i ] = 0;
S2RightHighCh1[ i ] = 0; S2RightHighCh2[ i ] = 0;S2RightHighCh3[ i ] = 0;
S2LowWidthCh1[ i ] = 0; S2LowWidthCh2[ i ] = 0;S2LowWidthCh3[ i ] = 0;
S2HighWidthCh1[ i ] = 0; S2HighWidthCh2[ i ] = 0;S2HighWidthCh3[ i ] = 0;
S2HalfWidthCh1[ i ] = 0; S2HalfWidthCh2[ i ] = 0;S2HalfWidthCh3[ i ] = 0;
S2RiseTimeCh1[ i ] = 0; S2RiseTimeCh2[ i ] = 0;S2RiseTimeCh3[ i ] = 0;
S2DropTimeCh1[ i ] = 0; S2DropTimeCh2[ i ] = 0;S2DropTimeCh3[ i ] = 0;


}
```

Ch#Analize.h (#=1, 2)

```
    for(int isample=1;isample<nb_samples;isample++)
     {
        if(vol1->GetBinContent(isample)<pre_signal_threshold)
            {
            sum_baseline+=vol1->GetBinContent(isample);
            square_baseline+=pow(vol1->GetBinContent(isample),2);
            sample_baseline++;
              }
    }

     BaseLineCh#=sum_baseline/sample_baseline;
     sigma_baseline=sqrt(square_baseline/sample_baseline-pow( BaseLineCh#,2));
     float signal_threshold=4.0*sigma_baseline;
    for(int isample=1;isample<nb_samples;isample++)
     {
          vol1->SetBinContent(isample,vol1->GetBinContent(isample)-BaseLineCh#);
     }

     AreaTotCh#= vol1->Integral(1,nb_samples)*window_length*1000/nb_samples;

    float Volt_tmp=0;

while(vol1->GetMaximum()>signal_threshold&&vol1->GetBinContent(vol1->GetMaximumBin()-1)>
signal_threshold&&vol1->GetBinContent(vol1->GetMaximumBin()+1)>signal_threshold)          {
       NbPeaksCh#++;
       PeakHeightCh#[NbPeaksCh#-1]=vol1->GetMaximum();
       PeakTimeCh#[NbPeaksCh#-1]=vol1->GetMaximumBin();

         for(int
isample=PeakTimeCh#[NbPeaksCh#-1];isample<PeakTimeCh#[NbPeaksCh#-1]+max_pulse_width;isa
mple++)
       {
              Volt_tmp=0;
         for(int jsample=isample-5;jsample<=isample+5;jsample++)
```

```
           {
             Volt_tmp+=vol1->GetBinContent(jsample);
            }
          Volt_tmp/=11.0;

if(Volt_tmp>0.88*PeakHeightCh#[NbPeaksCh#-1]&&Volt_tmp<0.92*PeakHeightCh#[NbPeaksCh#-1
])
           PeakRightHighCh#[NbPeaksCh#-1]=1.0*isample;

          else if
     (Volt_tmp>0.48*PeakHeightCh#[NbPeaksCh#-1]&&Volt_tmp<0.52*PeakHeightCh#[NbPeaksCh
     #-1])
           PeakRightHalfCh#[NbPeaksCh#-1]=isample;

          else if
(Volt_tmp>0.08*PeakHeightCh#[NbPeaksCh#-1]&&Volt_tmp<0.12*PeakHeightCh#[NbPeaksCh#-1])
           PeakRightLowCh#[NbPeaksCh#-1]=isample;

          else if(Volt_tmp<0.1*signal_threshold&&Volt_tmp>-0.1*signal_threshold)
          { PeakRightMostCh#[NbPeaksCh#-1]=isample; isample=
PeakTimeCh#[NbPeaksCh#-1]+max_pulse_width;}

}

       for(int
isample=PeakTimeCh#[NbPeaksCh#-1];isample>PeakTimeCh#[NbPeaksCh#-1]-max_pulse_width;isa
mple--)
       {
             Volt_tmp=0;
          for(int jsample=isample-5;jsample<=isample+5;jsample++)
           {
             Volt_tmp+=vol1->GetBinContent(jsample);
            }
          Volt_tmp/=11.0;


if(Volt_tmp<0.92*PeakHeightCh#[NbPeaksCh#-1]&&Volt_tmp>0.88*PeakHeightCh#[NbPeaksCh#-1
])
           PeakLeftHighCh#[NbPeaksCh#-1]=1.0*isample;

          else if
(Volt_tmp<0.52*PeakHeightCh#[NbPeaksCh#-1]&&Volt_tmp>0.48*PeakHeightCh#[NbPeaksCh#-1])
           PeakLeftHalfCh#[NbPeaksCh#-1]=isample;

          else if
(Volt_tmp<0.12*PeakHeightCh#[NbPeaksCh#-1]&&Volt_tmp>0.08*PeakHeightCh#[NbPeaksCh#-1])
           PeakLeftLowCh#[NbPeaksCh#-1]=isample;

          else if(Volt_tmp<0.1*signal_threshold&&Volt_tmp>-0.1*signal_threshold)
           {PeakLeftMostCh#[NbPeaksCh#-1]=isample;
isample=PeakTimeCh#[NbPeaksCh#-1]-max_pulse_width;
           }

}
          PeakLowWidthCh#[NbPeaksCh#-1]=
PeakRightLowCh#[NbPeaksCh#-1]-PeakLeftLowCh#[NbPeaksCh#-1];
          PeakHalfWidthCh#[NbPeaksCh#-1]=
```

```
PeakRightHalfCh#[NbPeaksCh#-1]-PeakLeftHalfCh#[NbPeaksCh#-1];
            PeakHighWidthCh#[NbPeaksCh#-1]=
PeakRightHighCh#[NbPeaksCh#-1]-PeakLeftHighCh#[NbPeaksCh#-1];
            PeakRiseTimeCh#[NbPeaksCh#-1]=
PeakLeftHalfCh#[NbPeaksCh#-1]-PeakLeftLowCh#[NbPeaksCh#-1];
            PeakDropTimeCh#[NbPeaksCh#-1]=
PeakRightLowCh#[NbPeaksCh#-1]-PeakRightHalfCh#[NbPeaksCh#-1];

PeakTotCh#[NbPeaksCh#-1]=vol1->Integral(PeakLeftMostCh#[NbPeaksCh#-1],PeakRightMostCh#[
NbPeaksCh#-1])*window_length*1000/nb_samples;


         for(int
isample=PeakLeftMostCh#[NbPeaksCh#-1];isample<PeakRightMostCh#[NbPeaksCh#-1];isample++)
           {
            vol1->SetBinContent(isample,0);
           }
            if(NbPeaksCh#>=Max_Peak)
            break;
     }



        for(int ipeak=0;ipeak<NbPeaksCh#;ipeak++)
        {

if(PeakRiseTimeCh#[ipeak]<35*nb_samples/(window_length*1000)&&PeakRiseTimeCh#[ipeak]>0)
           {
            NbS1PeaksCh#++;
             S1HeightCh#[NbS1PeaksCh#-1]=PeakHeightCh#[ipeak];
             S1TotCh#[NbS1PeaksCh#-1]=PeakTotCh#[ipeak];
             S1TimeCh#[NbS1PeaksCh#-1]=PeakTimeCh#[ipeak];
             S1LeftMostCh#[NbS1PeaksCh#-1]=PeakLeftMostCh#[ipeak];
             S1LeftLowCh#[NbS1PeaksCh#-1]=PeakLeftLowCh#[ipeak];
             S1LeftHalfCh#[NbS1PeaksCh#-1]=PeakLeftHalfCh#[ipeak];
             S1LeftHighCh#[NbS1PeaksCh#-1]=PeakLeftHighCh#[ipeak];

             S1RightMostCh#[NbS1PeaksCh#-1]=PeakRightMostCh#[ipeak];
             S1RightLowCh#[NbS1PeaksCh#-1]=PeakRightLowCh#[ipeak];
             S1RightHalfCh#[NbS1PeaksCh#-1]=PeakRightHalfCh#[ipeak];
             S1RightHighCh#[NbS1PeaksCh#-1]=PeakRightHighCh#[ipeak];

             S1HalfWidthCh#[NbS1PeaksCh#-1]=PeakHalfWidthCh#[ipeak];
             S1LowWidthCh#[NbS1PeaksCh#-1]=PeakLowWidthCh#[ipeak];
             S1HighWidthCh#[NbS1PeaksCh#-1]=PeakHighWidthCh#[ipeak];

             S1RiseTimeCh#[NbS1PeaksCh#-1]=PeakRiseTimeCh#[ipeak];
             S1DropTimeCh#[NbS1PeaksCh#-1]=PeakDropTimeCh#[ipeak];


           }
         else if(PeakRiseTimeCh#[ipeak]<400*nb_samples/(window_length*1000))
           {
            NbS2PeaksCh#++;
             S2HeightCh#[NbS2PeaksCh#-1]=PeakHeightCh#[ipeak];
             S2TotCh#[NbS2PeaksCh#-1]=PeakTotCh#[ipeak];
             S2TimeCh#[NbS2PeaksCh#-1]=PeakTimeCh#[ipeak];
```

```
      S2LeftMostCh#[NbS2PeaksCh#-1]=PeakLeftMostCh#[ipeak];
       S2LeftLowCh#[NbS2PeaksCh#-1]=PeakLeftLowCh#[ipeak];
       S2LeftHalfCh#[NbS2PeaksCh#-1]=PeakLeftHalfCh#[ipeak];
       S2LeftHighCh#[NbS2PeaksCh#-1]=PeakLeftHighCh#[ipeak];
       S2RightMostCh#[NbS2PeaksCh#-1]=PeakRightMostCh#[ipeak];
       S2RightLowCh#[NbS2PeaksCh#-1]=PeakRightLowCh#[ipeak];
       S2RightHalfCh#[NbS2PeaksCh#-1]=PeakRightHalfCh#[ipeak];
       S2RightHighCh#[NbS2PeaksCh#-1]=PeakRightHighCh#[ipeak];
       S2HalfWidthCh#[NbS2PeaksCh#-1]=PeakHalfWidthCh#[ipeak];
       S2LowWidthCh#[NbS2PeaksCh#-1]=PeakLowWidthCh#[ipeak];
       S2HighWidthCh#[NbS2PeaksCh#-1]=PeakHighWidthCh#[ipeak];
       S2RiseTimeCh#[NbS2PeaksCh#-1]=PeakRiseTimeCh#[ipeak];
       S2DropTimeCh#[NbS2PeaksCh#-1]=PeakDropTimeCh#[ipeak];
       }
  }


for(int is1=0;is1<NbS1PeaksCh#-1;is1++)
{
    for(int js1=is1+1;js1<NbS1PeaksCh#;js1++)
    {
        if(S1TotCh#[js1]>S1TotCh#[is1])
        {
        swapfloat(S1TotCh#[is1],S1TotCh#[js1]);
          swapint(S1TimeCh#[is1],S1TimeCh#[js1]);
          swapint(S1LeftMostCh#[is1],S1LeftMostCh#[js1]);
          swapfloat(S1LeftLowCh#[is1],S1LeftLowCh#[js1]);
          swapfloat(S1LeftHalfCh#[is1],S1LeftHalfCh#[js1]);
          swapfloat(S1LeftHighCh#[is1],S1LeftHighCh#[js1]);
          swapfloat(S1HeightCh#[is1],S1HeightCh#[js1]);
          swapint(S1RightMostCh#[is1],S1RightMostCh#[js1]);
          swapfloat(S1RightLowCh#[is1],S1RightLowCh#[js1]);
          swapfloat(S1RightHalfCh#[is1],S1RightHalfCh#[js1]);
          swapfloat(S1RightHighCh#[is1],S1RightHighCh#[js1]);
          swapfloat(S1LowWidthCh#[is1],S1LowWidthCh#[js1]);
          swapfloat(S1HighWidthCh#[is1],S1HighWidthCh#[js1]);
          swapfloat(S1HalfWidthCh#[is1],S1HalfWidthCh#[js1]);
          swapfloat(S1RiseTimeCh#[is1],S1RiseTimeCh#[js1]);
          swapfloat(S1DropTimeCh#[is1],S1DropTimeCh#[js1]);
       }
     }
  }


for(int is2=0;is2<NbS2PeaksCh#-1;is2++)
{
    for(int js2=is2+1;js2<NbS2PeaksCh#;js2++)
    {
        if(S2TotCh#[js2]>S2TotCh#[is2])
        {
        swapfloat(S2TotCh#[is2],S2TotCh#[js2]);
          swapint(S2TimeCh#[is2],S2TimeCh#[js2]);
          swapint(S2LeftMostCh#[is2],S2LeftMostCh#[js2]);
          swapfloat(S2LeftLowCh#[is2],S2LeftLowCh#[js2]);
          swapfloat(S2LeftHalfCh#[is2],S2LeftHalfCh#[js2]);
          swapfloat(S2LeftHighCh#[is2],S2LeftHighCh#[js2]);
          swapfloat(S2HeightCh#[is2],S2HeightCh#[js2]);
```

```
          swapint(S2RightMostCh#[is2],S2RightMostCh#[js2]);
            swapfloat(S2RightLowCh#[is2],S2RightLowCh#[js2]);
            swapfloat(S2RightHalfCh#[is2],S2RightHalfCh#[js2]);
            swapfloat(S2RightHighCh#[is2],S2RightHighCh#[js2]);
            swapfloat(S2LowWidthCh#[is2],S2LowWidthCh#[js2]);
            swapfloat(S2HighWidthCh#[is2],S2HighWidthCh#[js2]);
            swapfloat(S2HalfWidthCh#[is2],S2HalfWidthCh#[js2]);
            swapfloat(S2RiseTimeCh#[is2],S2RiseTimeCh#[js2]);
            swapfloat(S2DropTimeCh#[is2],S2DropTimeCh#[js2]);
         }
       }
     }


  for(int ipeak=0;ipeak<NbPeaksCh#-1;ipeak++)
  {
      for(int jpeak=ipeak+1;jpeak<NbPeaksCh#;jpeak++)
      {
          if(PeakTotCh#[jpeak]>PeakTotCh#[ipeak])
          {
          swapfloat(PeakTotCh#[ipeak],PeakTotCh#[jpeak]);
            swapint(PeakTimeCh#[ipeak],PeakTimeCh#[jpeak]);
            swapint(PeakLeftMostCh#[ipeak],PeakLeftMostCh#[jpeak]);
            swapfloat(PeakLeftLowCh#[ipeak],PeakLeftLowCh#[jpeak]);
            swapfloat(PeakLeftHalfCh#[ipeak],PeakLeftHalfCh#[jpeak]);
            swapfloat(PeakLeftHighCh#[ipeak],PeakLeftHighCh#[jpeak]);
            swapfloat(PeakHeightCh#[ipeak],PeakHeightCh#[jpeak]);
            swapint(PeakRightMostCh#[ipeak],PeakRightMostCh#[jpeak]);
            swapfloat(PeakRightLowCh#[ipeak],PeakRightLowCh#[jpeak]);
            swapfloat(PeakRightHalfCh#[ipeak],PeakRightHalfCh#[jpeak]);
            swapfloat(PeakRightHighCh#[ipeak],PeakRightHighCh#[jpeak]);
            swapfloat(PeakLowWidthCh#[ipeak],PeakLowWidthCh#[jpeak]);
            swapfloat(PeakHighWidthCh#[ipeak],PeakHighWidthCh#[jpeak]);
            swapfloat(PeakHalfWidthCh#[ipeak],PeakHalfWidthCh#[jpeak]);
            swapfloat(PeakRiseTimeCh#[ipeak],PeakRiseTimeCh#[jpeak]);
            swapfloat(PeakDropTimeCh#[ipeak],PeakDropTimeCh#[jpeak]);
         }
       }
     }
```

Algorithm 2 *plot_scope_trace.C (provided by Dr. Liu Jianglai)*

```
#include <vector>
#include <stdlib.h>
#include <stdio.h>
#include <TApplication.h>
#include <TROOT.h>
#include <TSQLServer.h>
#include <TSQLRow.h>
#include <TSQLResult.h>
#include <TCanvas.h>
#include <TH1.h>
#include <string>
#include <vector>
#include <fstream>
#include <iostream>
#include "Rtypes.h"
```

```
#include <TGraphErrors.h>
#include <TStyle.h>
#include <TMultiGraph.h>
#include <TStopwatch.h>

#include <TMath.h>
#include <TPaveText.h>
#include <TObjString.h>
#include <TLegend.h>
#include <TLegendEntry.h>
#include <TSystem.h>
#include <TFile.h>

using namespace std;

int draw_canvas(0);

vector<TString> SplitLine(const TString line)
{
   vector<TString> words;
   words.clear();
   char delimiters[] = " \t\n;,";
   TObjArray* Strings = line.Tokenize(delimiters);
   if(Strings->GetEntriesFast()) {
      TIter iString(Strings);
      TObjString* os=0;
      while ((os=(TObjString*)iString()))
         words.push_back(os->GetString());

      }
   }
   delete Strings;
   return words;
}

TH1F* convert_points_to_hist(vector<Double_t> x)
{
   vector<Float_t> xlow;
   double binsize(0);
   for(size_t ii=0;ii<x.size()-1;ii++){
      binsize = x[ii+1]-x[ii];
      xlow.push_back(x[ii]-0.5*binsize);
   }
   xlow.push_back(x[x.size()-1]-0.5*binsize);
   xlow.push_back(x[x.size()-1]+0.5*binsize);

   TH1F* hh = new TH1F("hh","",x.size(),&xlow[0]);


   return hh;
}

TH1F* convert_scope_trace_to_hist(const TString filename)
{
   int id = 0;
   TString line;
   ifstream fin(filename.Data());
```

```
    vector<double> time;
    vector<double> amp;
    while(fin){

        line.ReadLine(fin);
        if(line.IsNull()) break;
        vector<TString> words = SplitLine(line);
        if(words[0].IsFloat()) {
            time.push_back(atof(words[0].Data())*1e9);
            time.push_back((id++/16000.0-0.5)*1e3);
            amp.push_back(atof(words[0].Data()));
        }

}

    TH1F *hh = convert_points_to_hist(time);
    for(size_t ii=0;ii<time.size();ii++){
        hh->Fill(time[ii],amp[ii]);
    }

    hh->Draw();
    return hh;
}

double get_area(const TString filename,
                const double xlow, const double xhigh)
{
    TH1F* hh = convert_scope_trace_to_hist(filename);
    if(draw_canvas){
        hh->Draw();
        gPad->Update();
    }
    int binlow = hh->FindBin(xlow);
    int binhigh = hh->FindBin(xhigh);

    double area = -hh->Integral(binlow,binhigh)*(hh->GetBinLowEdge(2)-hh->GetBinLowEdge(1));

    delete hh; hh = 0;
    return area;
}

void draw_charge_histogram(const int runno1, const int runno2)
{
    TH1F* hCharge = new TH1F("hCharge","",100,-0.5,1);
    hCharge->SetXTitle("Scope area (V#dot ns)");
    TString filename;
    for(size_t ii=runno1;ii<=runno2;ii++){
        filename = Form("Waveforms/event%05d.txt",ii);
        double charge = get_area(filename, 75, 105);
        hCharge->Fill(charge);
    }
    hCharge->Draw();
}

int main(int argc, char** argv)
{
```

```
  if(argc!=8) {
    cout<<"Syntax: "<<argv[0]<<" <first run> <last run> <time low> <time high> <waveform folder>
<outfile name> <canvas? 0:1>"<<endl;
    exit(1);
  }

  gROOT->SetStyle("Plain");
  TApplication theApp("App",0,0);
  int firstrun = atoi(argv[1]);
  int lastrun = atoi(argv[2]);
  int timelow = atof(argv[3]);
  int timehigh = atof(argv[4]);
  TString waveformfolder = TString(argv[5]);
  TString outfilename = TString(argv[6]);
  draw_canvas = atoi(argv[7]);
  TH1F* hCharge = new TH1F("hCharge","",2000,-0.2,100);
  hCharge->SetXTitle("Scope area (V #times ns)");
  TString filename;
  for(size_t ii=firstrun;ii<=lastrun;ii++){
    filename = Form("%s/event%05d.txt",waveformfolder.Data(),ii);
    cout<<"Opening "<<filename.Data()<<endl;
    ifstream thisf(filename.Data());
    if(!thisf) continue;
    thisf.close();
    double charge = get_area(filename, timelow, timehigh);

   hCharge->Fill(charge);
 }
    hCharge->Draw();

  TFile *file = new TFile(outfilename, "recreate");
  hCharge->Write();
  file->Close();
  theApp.Run();
}
```