Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ETH Institute for Particle Physics

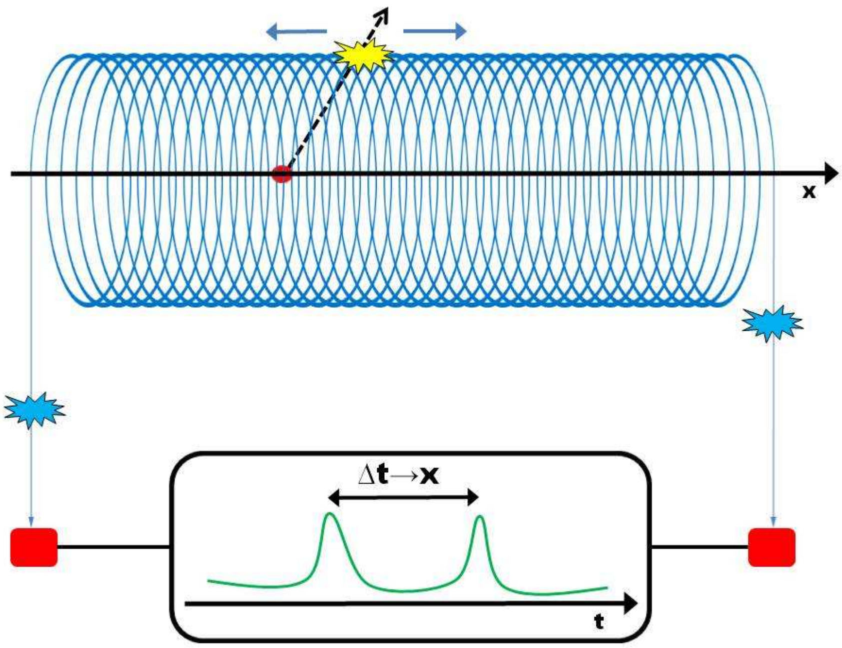
Physikpraktikum für Vorgerückte (VP)

vp.phys.ethz.ch

Development of a charged particle tracker with plastic scintillating fiber and Geiger-mode avalanche photodiode

Kim Siang Khaw

June 20, 2013



Abstract

Charged particle detector or tracker is one of the most important components in experimental particle physics. It provides both timing and spatial information about the charged particle like electron, and has many applications outside the field of particle physics as well. Here at the Institute for Particle Physics (IPP) at the Physics Department of ETH Zurich, we are developing a next generation charged particle tracker using plastic scintillating fiber (PSF) with its signal readout by multi pixel photon counter (MPPC). Peltier and water cooling systems are used to stabilized the system, a modern oscilloscope DRS4 is being used to record the waveforms and ROOT library is being used for analysis and histogramming.

In the due course of this experiment, you will have the chance to learn all the basics and techniques about particle detection and software for data analysis.

Contents

1 Introduction

Warning: This booklet provides only minimal information about the theoretical and experimental aspects of the experiment. Further reading is therefore recommended and will be given in the subsequent sections.

1.1 Particle Physics and Particle Detectors

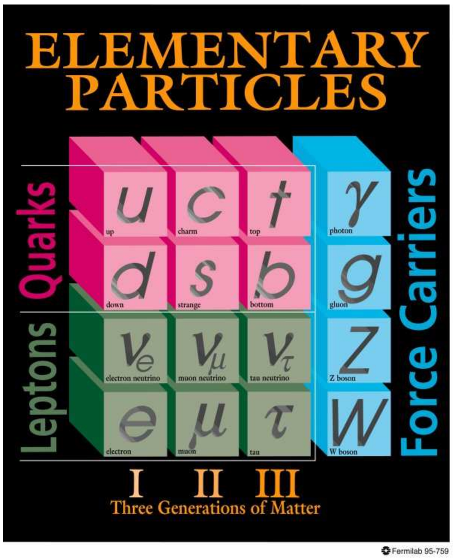


Figure 1: Elementary particles in Standard Model. It consists of leptons, quarks and force carriers.

Particle Physics is a field in Physics that studies the interactions in between a family of elementary particles such as electron and muon. All those particles and their interactions observed up to date can be described almost completely by Standard Model. (see Figure 1)

Standard model is a great collaboration effort of both theorist and experimentalist. In order to refine the model and to look for hidden new physics, experimentalist is pushing the technology to the limit to produce high efficiency, low dead-time, high DAQ rate and high precision particle detector.

1.2 Interaction of the charged particles with matter

When a charged particle with high velocity enters a material medium, it will interact with the electrons and nuclei in the medium and begins to lose its energy as it penetrates through the medium. Depending on the energy of the particle, energy loss can be categorized in 3 types; ionization, bremsstrahlung and pair production. In this experiment, only ionization process, where election-ion pair is produced, is relevant.

1.3 Physics of the Scintillation process

The physics of the scintillation process in polystyrene and in all organic materials is essentially governed by the electronic structure of the carbon atom which in the ground state is: $1s^2 2s^2 2p^2$. In the formation of hydrocarbons, the carbon atoms are considered to be in an excited state leading to a hybridized orbital structure, $1s2s2p^3$, with one π orbital and three ω orbitals.

In aromatic compounds like polystyrene, it is the excitation of ω-orbital electrons and their transition between discrete energy levels that leads to fluorescence. Fluorescence is defined as the radiative transition from the fist excited singlet state to the ground state, with the emission of a photon.

Excitation of π-electrons by ionizing radiation and the subsequent fluorescence of the excited states is the principal energy transfer process in organic scintillators. The primary scintillation efficiency, P, is just the fraction of the energy deposited in the scintillator that goes into π-electron excitation. For most aromatics, P $\approx$ 0.1. However, not all of this excitation energy leads to fluorescence.

1.4 Plastic Scintillating Fiber

Over 20 years ago, it was shown that long plastic scintillating fibers have better time resolutions than long bulk scintillators in the pioneering work of Kuhlen et al. [1]. By grouping the 1 mm diameter fibers into 2 cm $\times$ 3 cm cross section, they showed that the time resolution is about 2 times better than the geometrically identical bulk scintillation counter.

Companies such as BICRON [2] and KURARAY [3] are the main manufacturers of such fibers. The emission peak for blue(green) fiber is around 435(492) nm, with a decay time of about 3.0 ns and attenuation length of around 3.0 m. Number of photons produced per MeV energy from a minimum ionizing particle (MIP) is about 8000, for 1 mm diameter fiber. The photon trapping efficiency ranging from 4.4 % to 7.3 %, depending on the shape of the fiber (round or square).

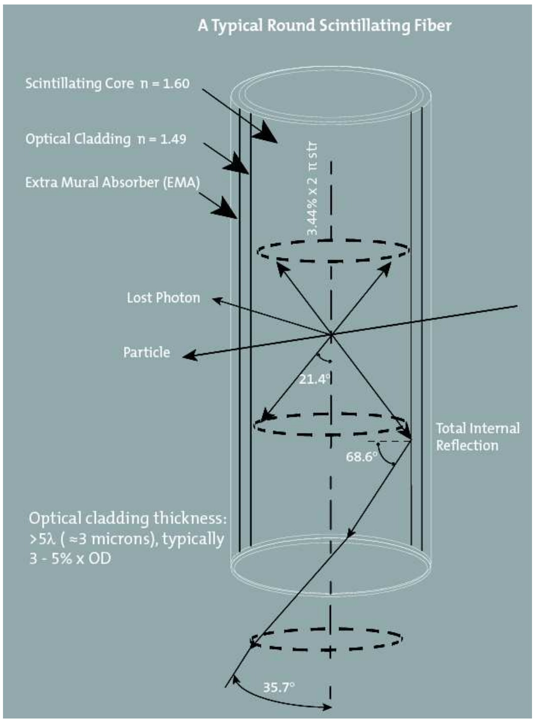


Figure 2: A typical round plastic scintillating fiber. [4]

Plastic scintillating fiber (PSF) which uses polystyrene (n = 1.61) as the core material and polymethylmathacrylate (n = 1.48) as the cladding material, is used in this experiment.

1.5 Multi pixel Photon Counter (MPPC/G-APD)

The MPPC (Multi-Pixel Photon Counter) is a new type of photon-counting device made up of multiple APD (avalanche photo diode) pixels operated in Geiger mode, and hence the name G-APD. The basic operating principle of G-APD is shown in Figure 3. The sum of the output from each APD pixel forms the MPPC output signal. This allows the counting of single photons or the detection of pulses of multiple photons. The MPPC has diverse applications such as fluorescence analysis, fluorescence lifetime measurement, flow cytometry, single molecule detection, neutrino detection, and PET (positron emission tomography).

The MPPC is essentially an opto-semiconductor device with excellent photon-counting capability Figure 4 and which also possesses great advantages. Features of the MPPC include the following:

- Excellent photon detection efficiency
- High gain: 10^5 to 10^6
- Room temperature operation
- Low bias (below 100 V) operation
- Insensitive to magnetic fields
- Excellent time resolution
- Compact and rugged

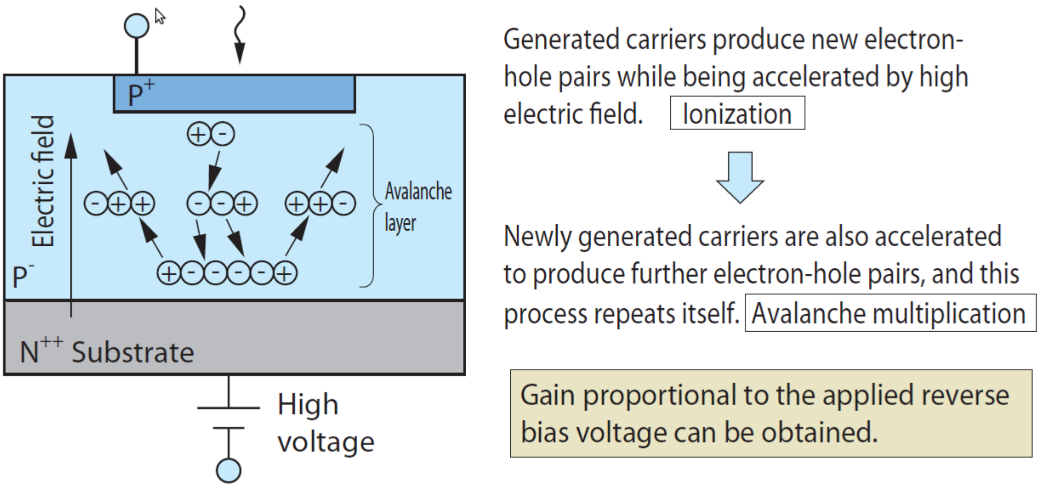


Figure 3: Operating principle example of APD. Taken from [4]

For more detailed information, please refer to [4] and a nice review written by D. Renker [5].

1.6 Peltier Cooling

1.6.1 History

The discovery began in the middle of 1821, where J. T. Seebeck discovered that two not similar metals, if they are connected in 2 different points and those points are held in different temperatures, there will be a micro-voltage developed. This effect is called the "Seebeck effect" as of it's discoverer.

Some years later, a scientist discovered the opposite of the Seebeck effect. He discovered that if someone applies voltage to a thermo-couple, one junction shall be heated and the other shall be cooled. The scientist was called Peltier and the effect called the "Peltier effect".

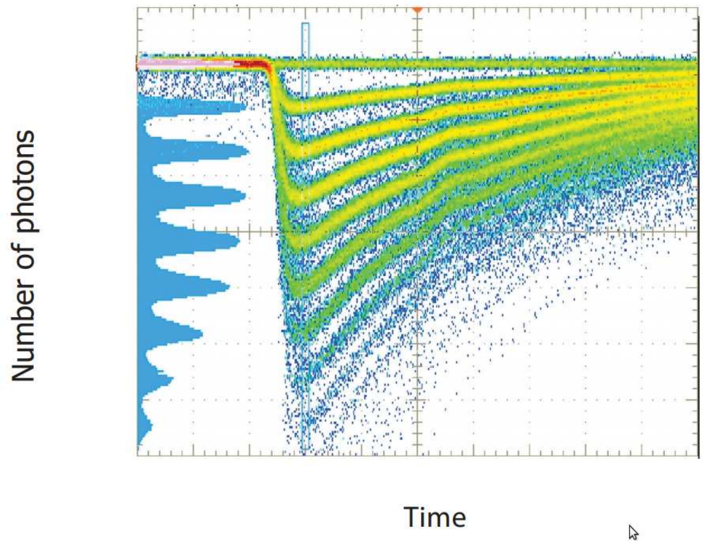


Figure 4: Photon counting ability of MPPC. Taken from [4]

1.6.2 Peltier Element

A Peltier thermo-element is a device that utilizes the peltier effect to implement a heat pump. A Peltier has two plates, the cold and the hot plate. Between those plates there are several thermo couples. All those thermo couples are connected together and two wires comes out. If voltage is applied to those wires, the cold plate will be cold and the hot plate will be hot.

The device is called a heat pump because it does not generate heat nor cold, it just transfers heat from one plate to another, and thus the other plate is cooled. It is also called a thermoelectric cooler or TEC for short.

1.6.3 Peltier Cooling System

Peltier cooling system in our lab consists of a peltier element, peltier controller and its power supply. The peltier controller uses derivative (PID) control that provides exceptionally tight control over a wide temperature range. Documentation about the model of peltier controller is available here [6].

1.7 Gas nitrogen

A tank of gas nitrogen is available in the laboratory. The pressure is about 200 bar and it has to be used with precautions. When experimenting with temperatures lower than 10∘C, a layer of ice will be formed on the surface of the apparatus. Hence it is adviced to flush in gas nitrogen to remove the water gas to prevent the ice formation. Make sure you open the windows of the room to prevent building up of gas nitrogen.

1.8 Domino Ring Sampler (DRS)

The DRS chip, which has been designed at the Paul Scherrer Institute, Switzerland by Stefan Ritt and Roberto Dinapoli is a Switched Capacitor Array (SCA) capable of digitizing eight channels at sampling speeds up to 5 GSPS. This chip is available through the PSI technology transfer program for other institutes and organizations. Currently DRS version 4 (DRS4) is widely being used and DRS version 5 will be released soon.

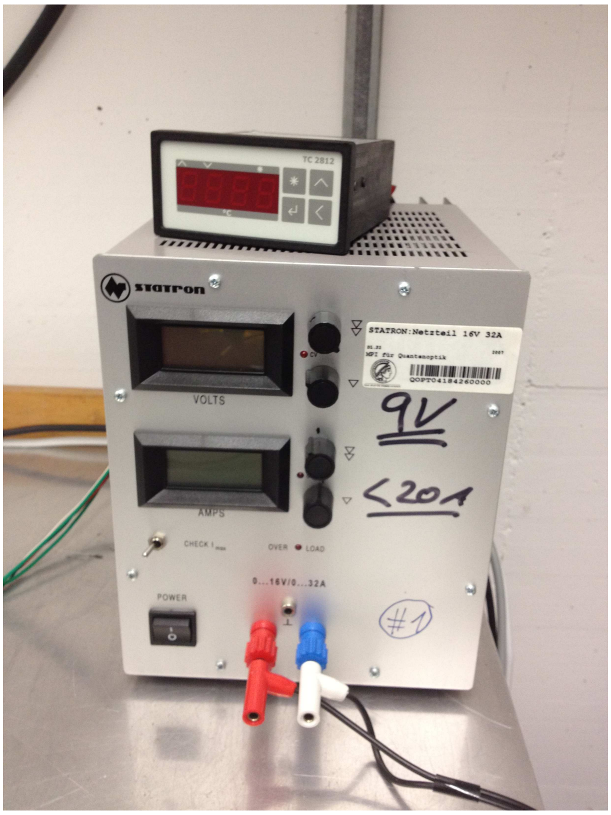


Figure 5: Peltier controller and its power supply

In order to simplify the design process to integrate the DRS4 chip into custom electronics, an evaluation board has been designed, which demonstrates the basic operation of the chip. It has SMA connectors for four input channels CH1 to CH4, an USB 2.0 connector and a LEMO trigger input (Figure 6). The board is powered through the USB port and contains an on-board trigger logic. It comes with MS Windows® and Linux drivers and two application programs. It is basically equivalent to a four channel 5 GSPS digital oscilloscope.

However, to increase the data writing speed, the data is saved in binary format. Hence, some special program is needed to analyze the recorded data.

2 General view of the laboratory

This student lab usually takes about 6 weeks, 4 weeks for the experiment and 2 weeks for the data analysis. You must submit the first version of the report within 2 weeks after you have finished the analysis and has to be finished before the next semester.

2.1 Goals of the laboratory

At the end of the laboratory, students should be able to

- list out all the elementary particles in the Standard Model of Particle Physics (C1)
- describe the interaction of charged particles with plastic scintillators (C2)
- describe the working principle of Geiger-mode Avalanche Photo Diode (G-APD) (C2)
- describe data acquisition techniques (DAQ) used in the laboratory (C2)

Figure 6: DRS4 Evaluation board version 4.

- drive a LED using Function Generator (C3)
- take data using DRS4 Digitizer (C3)
- describe various Waveform Analysis Techniques (C2)
- understand and program in C++ by using ROOT framework (C3)
- analyze the data using ROOT framework (C4)
- design a charged particle tracker based on small tasks (C5)
- evaluate the performance of the tracker built (C6)

where C1-C6 indicates the level of difficulty.

2.2 Proposed Schedule

Proposed schedule for the experiment is described in this section but in the end it is up to you to decide.

Week 1

In this week, introduction to this experiment will be done. Detailed instructions on how to drive a G-APD will be given using "old-fashion" setup. Study the signal coming from the G-APD using oscilloscope, and learn about photon counting. Differentiate between a signal and a noise using LED as example.

Week 2

Once you understood the basic ideas from last week, we will proceed with a more modern way of driving a G-APD. On this week, you will need to characterize the G-APD given, by varying temperature and applied voltage. From the data, you need to find out the gain and breakdown voltage of a G-APD.

Word	Byte 0	Byte 1	Byte 2	Byte 3	Contents
0	'E'	'H'	'D'	'R'	Event Header
1	Serial number				Serial number starting with 1
2	Year		Month		Event date/time 16-bit values
3	Day		Hour		
4	Minute		Second		
5	Millisecond		*reserved*		
6	Time Bin #0				Time of sample bins in ns encoded in 4-Byte floating point format
7	Time Bin #1				
...	...				
1029	Time Bin 1023				
1030	'C'	'0'	'0'	'1'	Channel 1 header
1031	Voltage Bin #0		Voltage Bin #1		Channel 1 waveform data encoded in 2-Byte integers. 0=-0.5V and 65535=+0.5V
1032	Voltage Bin #2		Voltage Bin #3		
...	...		...		
1542	Voltage Bin #1022		Voltage Bin #1023		
1543	'C'	'0'	'0'	'2'	Channel 2 header
1544	Voltage Bin #0		Voltage Bin #1		Channel 2 waveform data encoded in 2-Byte integers. 0=-0.5V and 65535=+0.5V
1545	Voltage Bin #2		Voltage Bin #3		
...	...		...		
2055	Voltage Bin #1022		Voltage Bin #1023		
2056	'E'	'H'	'D'	'R'	Next Event Header
...					

Figure 7: DRS4 data format. Taken from [7]

Week 3

In this week we will do some tests with LEDs which could give almost constant number of photons. Coincidence of signals between 2 G-APDs will be utilized to reduce the backgrounds to find out the spectra of signal coming from LED.

Week 4

In this week, plastic scintillating fiber (PSF) will be used as the simple particle tracker. Radioactive source or LED will be used to test our "dummy tracker" and from there you will need to find out the efficiency and time resolution of this method. Since PSF is available in various sizes (0.5 mm, 0.75 mm, 1.0 mm, 1.5 mm and 2.0 mm), you can find out the time resolutions of different sizes.

Week	Time	Topics	Goals
1	9 am - 12 pm	Self introduction, overview of the lab	to have an overview of the lab
2	9 am - 10 am	G-APD	to understand the working principle
	10 am - 12 pm	DAQ in this lab	to understand DAQ and master it
	1:30 pm - 6 pm	Characterization of G-APD	able to set up the experiment for measuring the signal from G-APD
3	9 am - 10 am	LED and function generator	able to control the intensity and frequency of the LED
	10 am - 12 pm	LED + G-APD	able to set up the experiment to detect the LED pulses
	1:30 pm - 3 pm	LED + 2 G-APD	able to select events where both G-APDs have picked up the same pulse
	3 pm - 6 pm	Comparison	Analyze and compare both scenarios
4	9 am - 10 am	Plastic scintillating fiber (PSF)	to understand how PSF detect charged particles
	10 am - 12 pm	Attenuation length of PSF	to set up the experiment to measure the attenuation length of the PSF
	1:30 pm - 6 pm		to take data and calculate the attenuation length from it
5	9 am - 10 am	Radioactive source	to understand different types of radioactive source
	10 am - 12 pm	Charged particle tracker	to design a tracker based on experience gained previously
	1:30 pm - 6 pm		to set up the tracker and analyze the data taken
6	9 am - 6 pm	Reservation, feedback and discussion	to give feedback about the experiment or repeat unsatisfied part of the experiment

3 Experimental Topics

There are mainly four topics at the moment.

3.1 Multi pixel Photon Counter (MPPC/G-APD)

- Understanding of the operational principle of a MPPC.

- Studies of the gain, dark count (0.5 p.e., 1.2 p.e.) as a function of applied voltage and temperature.
- Studies of the ratio between signal and noise as a function of voltage and temperature.

3.2 Plastic Scintillating Fiber (PSF)

- Understanding of the principle of scintillating fiber.
- Studies of the effect of the polishing on the edges of the fiber.
- Studies of the reproducibility of signal when connecting MPPC and PSF using optical grease.
- Studies of the attenuation lengths of different fibers (diameter, shape, etc).

3.3 Peltier and Water Cooling

- Understanding of the principle of peltier cooling.
- Check the stability of the temperature at different temperature (0 C, 5 C, 10 C, 15 C, 20 C, 25 C).

3.4 DRS4 Data Analysis

- Understanding of the basic principle of DRS4.
- Understanding of the binary data structure of DRS4.
- Basic knowledge in C++ and ROOT.
- Optimization of the time resolution of time spectra.
- Optimization of the energy resolution of energy spectra.

4 Basics for conducting this experiment

4.1 Experimental setup

This experiment is conducted on the tabletop scale (Figure 9). A chamber (Figure 8) is used to contain G-APDs and Peltier cooling system, etc. A small table with shielding box is used to contain fibers. A PC operated in Ubuntu (Linux) is used for the data acquisition (Figure 10).

4.2 Multi pixel Photon Counter (MPPC/G-APD)

Applied voltage for the G-APD is about 71.0 V. Please refer to the package where it is stored for its manufacturing value. The current flowing through it is in the order of μA.

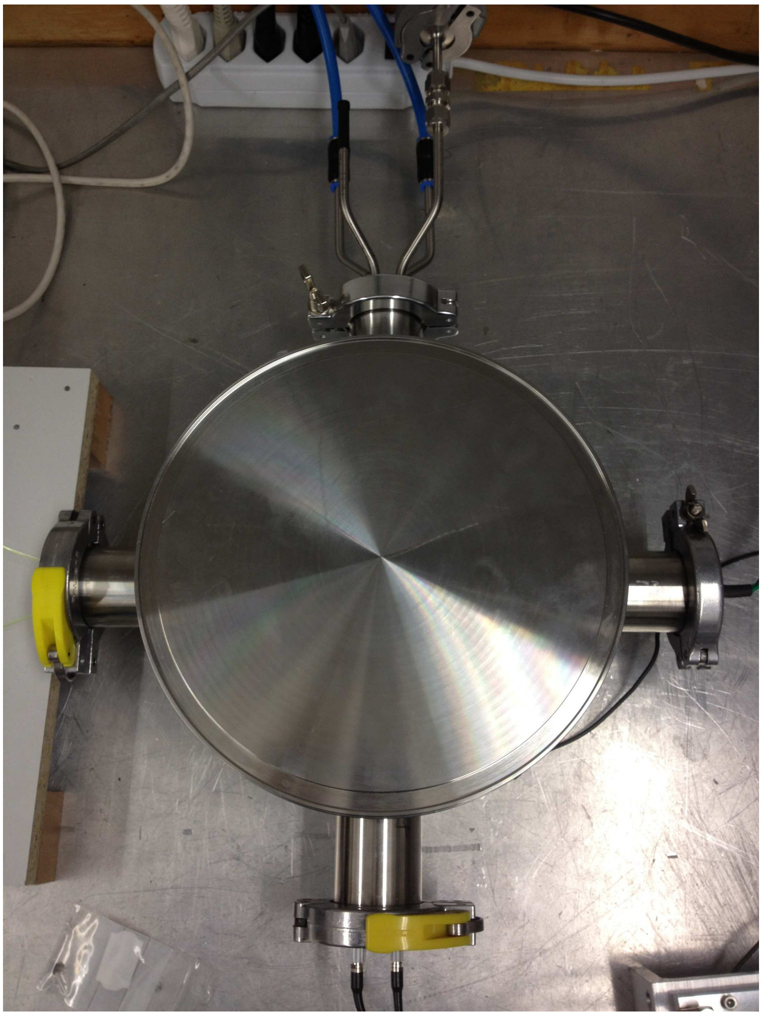


Figure 8: Chamber used in the experiment.

Figure 9: Table used in the experiment.

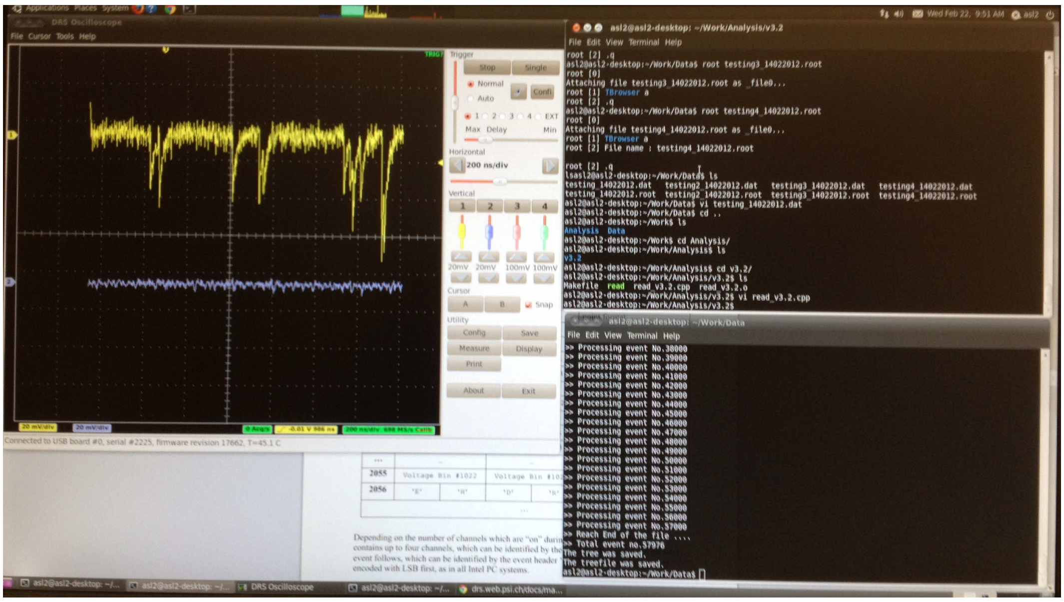
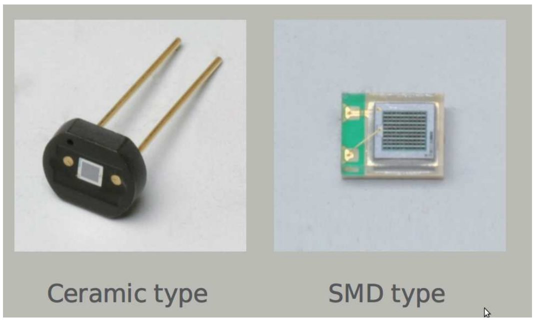


Figure 10: Screen-shot of the terminal used for DAQ and data analysis



(a) 1 × 1 mm type MPPC

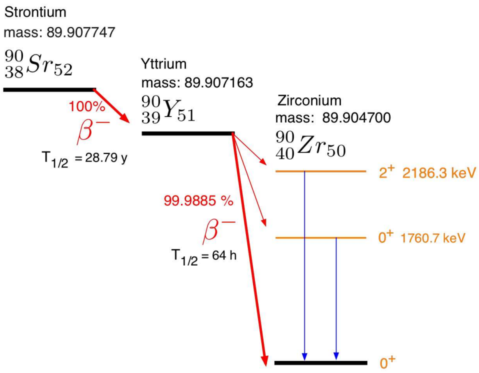
(b) 3 × 3 mm type MPPC

Figure 11: Active area 1 × 1 mm and 3 × 3 mm types of MPPC manufactured by Hamamatsu Photonics. [4]

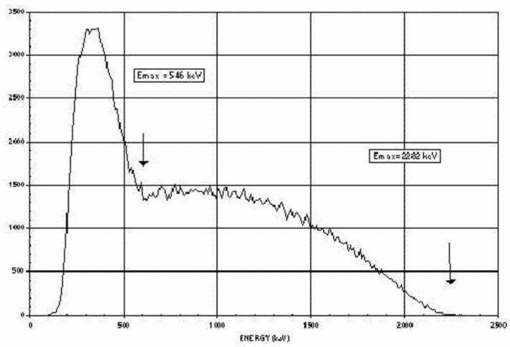
4.3 Radioactive source

Strontium-90 (Sr-90) is used as the beta emitter in this experiment. It is in equilibrium with the Yttrium-90 (Y-90) daughter. Thus, a 37 kBq (as of 01.05.2012) Sr-90 source also contains 37 kBq of Y-90.

Sr-90 undergoes beta decay to Yttrium-90 with a half-life of 28.8 years and a decay energy of 0.546 MeV, whereas Y-90 undergoes beta decay to Zirconium-90 with a half-life of 64 hours and a decay energy of 2.28 MeV.



(a) Decay chain



(b) Energy spectrum

Figure 12: Decay chains of Sr-90 to Y-90 and Zr-90.

4.4 MSCB Power Supply

MSCB interface is used to drive the power supply for G-APD. Open a terminal using ASL computer (Ubuntu OS). You will see the following at the terminal;

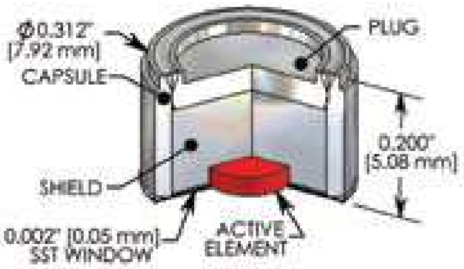
```
asl@asl-desktop:~\$
```

Then go to the folder of MSCB by type in the following command;

```
asl@asl-desktop:~\$cd Programs/mscb/
```

Then after type in ”ls”, which is to list out all the folders and files, you will see the following;

```
asl@asl-desktop:~/Programs/mscb\$ls
calib_1000.c calib_hvr.exe embedded libusb0.dll midas mscb.c
```



(a) Source capsule

(b) Actual source

Figure 13: Single-encapsulated stainless steel capsule used in this experiment.

```
mscb.hvi mscbrpc.c mscbrpc.o msc.exe musbstd.o mxml.o strlcpy.c
strlcpy.o calib_hvr.c drivers labview Makefile msc mscb.h mscb.o
mscbrpc.h msc.c msc.o mxml sendmail.c strlcpy.h
```

To control the MSCB, use the following command;

```
asl@asl-desktop:~\$sudo ./msc
```

If there is no error, that means you can start to set the voltage to the channel that you want. Due to the convention, address number 104 corresponding to the Channel 0 (CH0) of the MSCB module, 105 to CH1 and so on.

To set the parameters of CH0, first you need to go to address 104 as follows;

```
>>addr 104
```

To view the current parameters in CH0, type

```
>>read
```

To initialize CH0, type

```
>>wr 0 3
```

To set a value to certain parameter, i.e. voltage, type

```
>>wr 1 71
```

The first argument is write (wr), the second argument is the parameter number (1) and the third argument is the value that you want to set (71). To check the current status, type "read" again and you will have;

```
>>read
```

4.5 Pre-amplifier

Applied voltage for the pre-amp is 12 V. The gain is about 10-100 times. Please verify it to make sure.

4.6 Peltier Cooling

Please find out the operational principle of peltier element. Basically it has 2 sides, 1 for cooling and the other for heating, depending on the direction of signal is being sent. The supply voltage is 12 V and current is about $6A$. pt100 is used as the thermal sensor and also as a feedback to the cooling system. For long term operation, the heat produced by it is no longer negligible and water cooling must be used to remove the excess heat.

4.7 Water Cooling

A closed-loop water cooling system is used to remove the heat produced by peltier element. The temperature can be controlled by turning the knobs.

4.8 Chamber, Flanges and Feed-through

To avoid noise pick up for the G-APD signal, the wiring must be separated for different systems. For example, since there are 4 exits for the chamber, we devote 1 for the fiber, 1 for the peltier cooling system, 1 for the G-APD and the other for water cooling and gas system.

4.9 DRS4 data acquisition

Signal of G-APD is to be connected to the input (CH1 to CH4) of DRS4. DRS4 is then connected to a computer through USB interface. Open a terminal using ASL computer (Ubuntu OS). You will see the following at the terminal;

```
asl@asl-desktop:~\$
```

Then go to the folder of DRS4 by type in the following command;

```
asl@asl-desktop:~\$cd Programs/drs-4.0.0/
```

Then after type in "ls", which is to list out all the folders and files, you will see the following;

```
asl@asl-desktop:~/Programs/drs-4.0.0\$ls
AboutDialog.o Data doc DOScreen.o drscl.o drs_exam.o drsosc
DRSOsc.o firmware InfoDialog.o Makefile Measurement.o mxml.o
rb.o src TriggerDialog.o ConfigDialog.o DisplayDialog.o
DOFrame.o drscl drs_exam DRS.o drsosc.cfg EPThread.o include
main.o MeasureDialog.o musbstd.o Osci.o run.sh strlcpy.o
```

To run DRS4 oscilloscope mode, use the following command;

```
asl@asl-desktop:~/Programs/drs-4.0.0\$sudo ./drsosc
```

You can select the trigger level using the scroll on the right top corner, and the trigger source can be from CH1 to CH4 or external trigger. Under the column "Utility", you can save the file by inserting the file name. (Figure 14)

4.10 DRS4 Data Viewer (drs4view)

drs4view is a program written in C++ using ROOT libraries. It serves as a simple event viewer of the data taken using DRS4.

To run drs4view, use the following command;

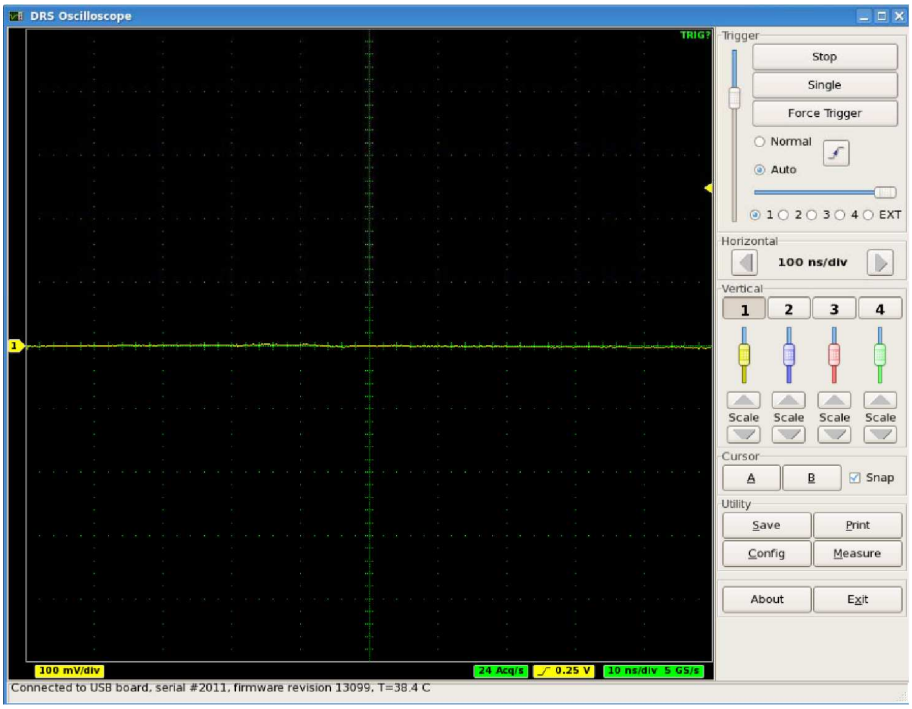


Figure 14: Graphical User Interface of DRS4 Oscilloscope.

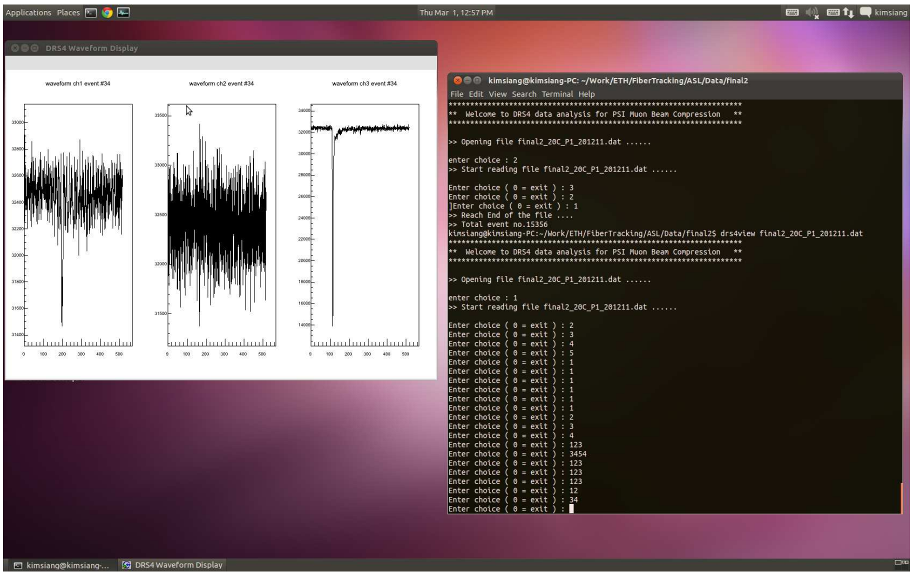


Figure 15: A screen-shot of drs4view.

```
asl@asl-desktop:~/Work/Data\$drs4view XXX.dat
```

At the prompt line, simply enter the event number that you want to inspect (e.g. 1) and the corresponding waveforms in all recorded channels will be plotted. Then you will be asked to input another event number (e.g. 2). If you wish to end the viewer, simply enter 0.

```

>> Opening file XXX.dat ......

enter choice : 1
>> Start reading file XXX.dat ......

Enter choice ( 0 = exit ) : 2
```

4.11 Some useful commands for Linux

In this section, you will find some useful commands for LINUX which will aid your research work at advanced student lab.

cd
Change the working directory.

```
\$cd Programs
```

cp
Copy a file.

```
\$cp file.txt copy_of_file.txt
```

rm
Remove a file.

```
\$rm file.txt
```

locate
Find the location of a certain file.

```
\$locate file.txt
```

4.12 C++ language

Mastering the C++ language is not a must but will help to understand more about the data analysis of DRS4. To analyze the DRS4 data, you will need to convert its binary format to something readable. To do so, download the c++ program written by K. S. Khaw from http://www.phys.ethz.ch/~khaw/research/ Download the latest version (currently is version 3 with files read_v3.cpp and Makefile). In case the Makefile is in the txt format (Makefile.txt), execute the following command in your terminal;

```
mv Makefile.txt Makefile
```

Put both files into a folder called "Analysis" and compile them;

```
asl@asl-desktop:~/Programs/drs-4.0.0/Data/Analysis\$ls
Makefile   read_v3.cpp
asl@asl-desktop:~/Programs/drs-4.0.0/Data/Analysis\$make
g++ -O2 -pipe -Wall -W -Woverloaded-virtual -fPIC -Iinclude
```

```
-pthread -I /usr/lib -o read_v3.o -c read_v3.cpp
```

To analyze a data file located at Data folder (asl@asl-desktop: /Work/Data/example.dat), run the following command;

```
asl@asl-desktop:~/Work/Data/\$readDRS4 example.dat
**************************************
**  Welcome to DRS4 data analysis  **
**************************************
>> Opening file example.dat ......
>> Creating rootfile example.root ......
>> Start reading example.dat ......
>> Processing event No.0
>> Reach End of the file ....
>> Total event no.1
The tree was saved.
```

You will see the above results if the file is successfully analyzed. Now the "example.root" file is being created and you will need to use ROOT to do further analysis (implementing energy and timing cuts, etc.).

4.13 ROOT

ROOT [8], a free and yet amazing software developed by CERN is very important for data analysis. In this section, a minimum of "what you should be able to do" is demonstrated. For detailed information, please refer to the documentations available in [8].

ROOT is very convenient in the sense that it can be used in various ways; using graphical user interface (GUI) or interactively, using it as a library in your compiled analysis code (for example C++) and also executing it from a macro (written codes in C++ without the needs of compilation, hence easy to use).

To start using ROOT interactively, type

```
asl@asl-desktop:~\$ root
  *******************************************
  *                                         *
  *        W E L C O M E  to  R O O T       *
  *                                         *
  *   Version   5.30/00      27 June 2011   *
  *                                         *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch            *
  *                                         *
  *******************************************
ROOT 5.30/00 (tags/v5-30-00@40062, Jun 28 2011, 11:49:57 on linux)
CINT/ROOT C/C++ Interpreter version 5.18.00, July 2, 2010
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
root [0]
```

Since it is very troublesome to re-type or trace back all the commands that you have typed, using ROOT in macro form is highly recommended. From now on, every example will be given in the form of macros.

4.13.1 Histogram

Histogram is often used in experimental particle physics to illustrate energy, momenta and time spectra (Figure 16). A histogram consists of tabular frequencies, shown as adjacent rectangles, erected over discrete intervals (bins), with an area equal to the frequency of the observations in the interval. A good explanation about histogram is available here [9].

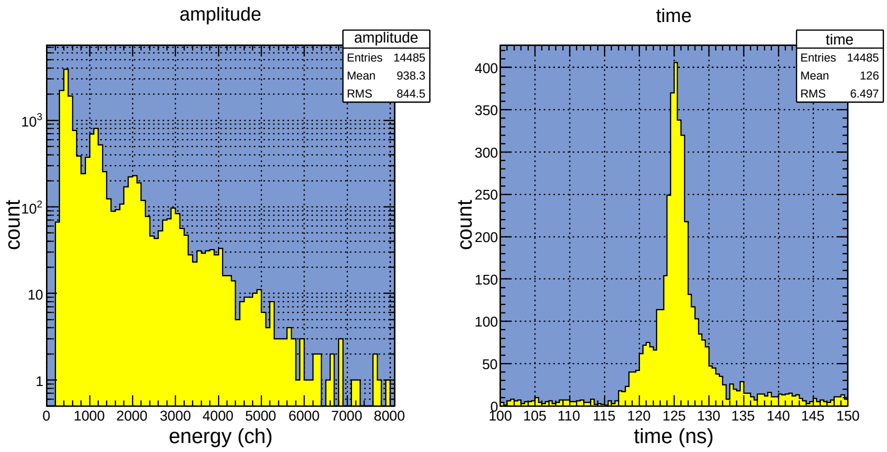


Figure 16: An example of energy and time spectra drawn as histograms.

To plot a histogram, create a file hist1D.C as follow;

```
void hist1D(){

TH1D *h1 = new TH1D("h1","h1",10,0,10);
h1->Fill(2,5);
h1->Fill(4,1);
h1->Fill(6,9);
h1->Fill(7,3);
h1->Fill(9,1);

h1->Draw();
}
```

Note that the name of the function must be the same as the file name. Then execute the macro using the follow command and you will get Figure 17.

```
root -l hist1D.C
```

4.13.2 Graph

A Graph is a graphics object made of two arrays X and Y with n points each. It is often used to represent the dependence of parameters X and Y.

To plot a graph, create a file graph.C as follow;

```
void graph(){

double x[5],y[5]; // declare 2 arrays, x and y

for(int i=0;i<5;i++){
x[i]=i;
y[i]=i*i;  // plotting y=x^2
}

TGraph *g1 = new TGraph(5,x,y);
g1->Draw("AP");
}
```

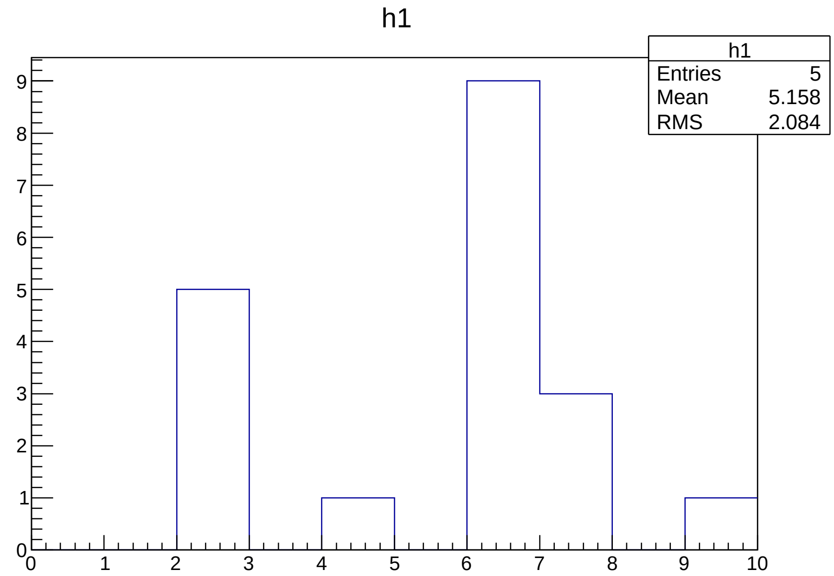


Figure 17: Result from macro hist1D.C.

Then execute the macro using the follow command and you will get Figure 18.

```
root -l graph.C
```

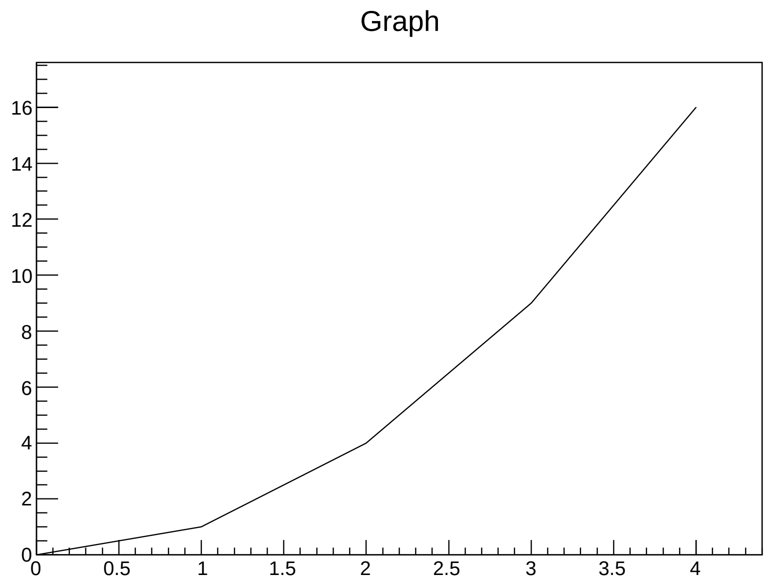


Figure 18: Result from macro graph.C.

5 Summary

After understanding and mastering everything in this guidebook, you are now able to conduct your own experiment without needing much help. Good luck and hope you enjoy working with me in this lab.

A Instructions for the experiment

During the course of the experiment, you must constantly read up the recommended literature to improve your knowledge about it. You are supposed to keep a logbook as well to keep track of every details about the conditions and what you have done. Discussions, questions and interactions with the assistants during the experiment are warmly encouraged.

References

[1] M. Kuhlen et al. Timing properties of long scintillation counters based on scintillating fibers. *Nucl. Instr. and Meth. A*, 301(223), 1991.

[2] Saint-Gobain Crystals. Scintillating optical fibers, scintillating products. 2011. URL `http://www.detectors.saint-gobain.com/uploadedFiles/SGdetectors/Documents/Brochures/Scintillating-Optical-Fibers-Brochure.pdf`.

[3] Kuraray. Plastic scintillating fibers, plastic imaging fibers. 2011. URL `http://kuraraypsf.jp/`.

[4] Hamamatsu Solid State Division. Mppc, multi-pixel photon counter. 2010. URL `http://sales.hamamatsu.com/assets/applications/SSD/high_resolution_mppc_kapd0002e08.pdf`.

[5] D. Renker. Geiger-mode avalanche photodiodes, history, properties and problems. *Nucl. Instrum. Methods Phys. Res. A*, 567:48–56, 2006.

[6] CoolTronic. Universal temperature und peltier-controller tc2812-rs232. 2008. URL `http://www.cooltronic.ch/65/CONTROLLER/PRODUKTE/TC2812-RS232.html`.

[7] Stefan Ritt. Drs4 evaluation board users manual. 2012. URL `http://drs.web.psi.ch/docs/manual_rev40.pdf`.

[8] Rene Brun and Fons Rademakers. Root - an object oriented data analysis framework. 1997. URL `http://root.cern.ch/drupal/`.

[9] Ken Cecire. Histograms: Construction, analysis and understanding. 2002. URL `http://quarknet.fnal.gov/toolkits/ati/histograms.html`.