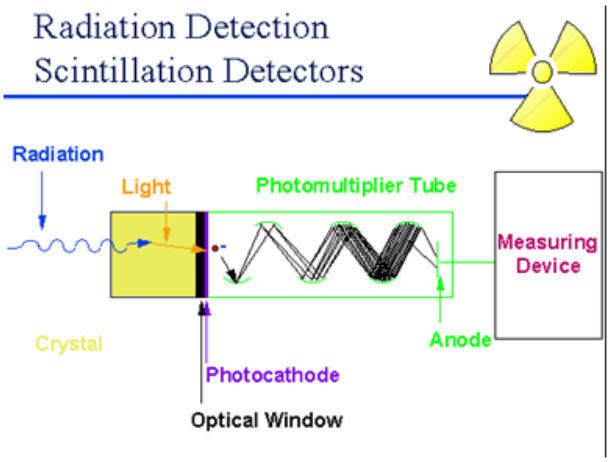
Exercises for the YAP/digitizer practical

Summary

This practical aims to teach the students how to use a scintillating crystal in combination with a digital data acquisition system (DDAQ). The students are introduced to some of the typical steps required in order to digitize a signal from a scintillating crystal and are guided to think about typical advantages and disadvantages that a DDAQ system can offer.

Introduction

In this practical the detector system consists of a YAP (Yttrium Aluminum Perovskite activated by Ce3+) crystal coupled to a photo multiplier tube. The photomultiplier tube is fed by a high voltage module.
YAP crystals are an example of a solid scintillator. When radiation impinges on a solid scintillator some electrons are promoted from the valence band to the conduction band of the crystal. These electrons are free to migrate throughout the crystal until they de-excite back to the valence band. The role of the impurity, called the activator (Cerium for YAP), is to introduce energy levels inside the band gap. When each electron de-excites back to the valence band through energy states introduced by the impurity, a visible photon is emitted. YAP has a quite high light yield, about 24000 photons/MeV of energy for each impinging gamma-ray quantum. The visible, scintillating light of the crystal is converted into a certain number of electrons by the photocathode, thanks to the photoelectric effect. This signal is amplified in the photo multiplier tube through a set of dynodes, producing the final output pulse that can be collected by a reading circuit.
A schematic of the detector system is shown below.



In this practical I have already arranged the system for you. The 2''x1'' YAP crystal lies on the photomultiplier tube. A bit of optical grease has been used for light coupling between the crystal and the photocathode. The whole system is packed in a mu-metal tube and sealed by a black lid to avoid light dispersion. That's why you can't see the crystal from the outside!

Exercise 1 - Identify the signal using an oscilloscope

There are two cables coming out of the detector (i.e. the sealed black tube). The red cable is the high voltage (HV) cable, while the black one carries the signal you are interested in.

Connect the red cable to the HV module and the black cable to the oscilloscope. Put a radioactive source close to the crystal. Power on the HV module and gradually increase the HV. Setup the oscilloscope so to identify a signal.

1) At which HV value do you start to appreciate a signal?

2) What is the time width of the signal?

3) Study how the signal amplitude changes as a function of the applied HV. Fill in the table below.

HV (V)	Signal amplitude (mV)
600	
700	
800	
900	
1000	
1100	
1200	

Plot your data using Origin (the program is found in the folder Frascati_Detector_School on the Desktop). Which law do they follow?

Exercise 2 - Entering the digital world

As I wrote in the summary section, one of the aims of this practical is to introduce you to the digital world. In this practical you will digitize the pulse you have seen on the oscilloscope using the digitizer DT5751 produced by CAEN. Read the datasheet of this digitizer contained in the folder Frascati_Detector_School/Manuals (this folder can be found on my laptop Desktop). Try to understand the meaning of each parameter. If you don't understand something, ask for explanations!

Now you are ready for the real exercise. Power on the digitizer and connect it to my laptop using the USB cable provided. Connect the black cable from the detector to the digitizer. Open the folder WaveDumpFolder in Frascati_Detector_School. Open the bin subfolder and launch the program named WaveDump. This program talks to the digitizer and let you acquire digitized data.

a) Start the acquisition. Press SHIFT+P (for plotting). You should see a pulse similar to that you were seeing on the oscilloscope. If so, that's a good starting point!

WaveDump reads a configuration file before starting. The configuration file is named WaveDumpConfig.txt and is found in the bin subfolder. There are several parameters you can play with. Have a quick look at these parameters and try to understand their meaning. We will play with some of them only.

b) Find the RECORD_LENGTH parameter. This controls how many points are stored for each triggered event. Set it first to 406 and then to 2000. This time you need to tell the device that you want to write data on the disk. This is done by pressing SHIFT+W in WaveDump.

See the differences in the two cases using WaveDump. When the parameter is set to 2000, what happens to the "busy" LED on the module?

When this LED is on, it means that the module is busy transferring data to the computer. When this occurs, the detector is missing some events. The time required to transfer data to a device has to be taken into account in choosing a DDAQ system and can sometimes be the bottle neck. There are several protocols that can be used to transfer data: USB, optical link, PCIExpress... Now we are using USB, which is not the fastest available on the market.

c) Set RECORD_LENGHT back to 406. Now look for OUTPUT_FILE_FORMAT. You can specify whether you want ASCII or BINARY data. Try to write data using both options (one at a time, of course!). Let each run last about 5 seconds. When you write data, the file containing your output is named wave0. Check the size of wave0 in both cases. What can you say?

ASCII data are easy to read (you need only a txt editor). However, ASCII files can soon become impractically huge. Writing BINARY files is always a good choice, once you know what their format is.

Exercise 3 - Making a spectrum out of digitized data

Open the ASCII file you have just created. As the file can be large, you might need to perform a new short measurement to obtain an ASCII file of practical size. Study its format.

This file has the typical format of a raw data file obtained from a digitizer. Each pulse is made of two parts: an header, that specifies several parameters, and the actual pulse itself. The most important part of the header is the time stamp, that lets the user perform a time resolved data analysis.

It's now time to use the digitized data to make a spectrum. Set OUTPUT_FILE_FORMAT back to BINARY and collect data for about 1 minute or so. The file you have just created is named wave0.dat. Copy this file to the Programs subfolder. Now comes the hard part... you have to make a spectrum out of the data you have just collected. Don't worry, I have done most of the hard work for you. You can run my program read_data to analyze your file. The program produces several files. We are interested in particular into two of them: waveform and the file that ends with _spectrum. Let's first start with waveform.

a) Open waveform with Origin. What is its format? What does it contain? Evaluate the time width of a pulse from the digitized data. Does the width agree with the estimate you have made using the oscilloscope? Consider the number of points that make up each pulse and figure out how this number varies as a function of the digitizer sampling frequency. Consider that, as a rule of thumb, one should have about 7 points in the FWHM.

Now the final part: making a spectrum. I will first tell you what my program is doing. There are two steps that one always has to do to make a spectrum out of digitized data. First, data needs to be *reduced,* i.e. from each pulse one needs to extract two numbers: the time stamp and a number that is proportional to the information (energy) contained in that pulse. This is the hardest part of the job as the algorithm chosen can affect the energy resolution in different ways. Consider that, in the non-digital world, a traditional analog spectroscopy chain generally gives a pulse with an optimized S/N ratio that is nicely fed into a multi channel analyzer. Here we have to retrieve an optimum S/N using a suitable reconstruction algorithm, there is no analog filter helping us most of the times! Besides, when pile up is of relevance, the data reduction stage is where pile up events have to be taken into account. In my program, I am actually doing

one of the simplest things to do: the energy contained in each pulse is evaluated as the maximum data point of the pulse minus the baseline. I am also not taking care of pile up events.

The second part of the job is easier: a *spectrum* has to be built from the reduced data, in a given time window. For the sake of simplicity, my program is making a spectrum using all data collected.

b) Open the _spectrum file using Origin. How does the spectrum look like? What are its main features? Why is the peak broadened? Do you think there can also be a contribution from the reconstruction algorithm?

You have now made your first steps into the digital world. Recollect what you have (possibly) learnt in this practical and compare advantages/disadvantages of analog and digital spectroscopy systems.